Theory and Methodology

House of quality: A fuzzy logic-based requirements analysis

Cecilia Temponi a, John Yen b,*, W. Amos Tiao b

a Southwest Texas State University, School of Business, San Marcos, TX 78666-4616, USA
b Center for Fuzzy Logic, Robotics, and Intelligent Systems, Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA

Received 12 June 1997; accepted 28 April 1998

Abstract

House Of Quality (HOQ) is one of the matrices of an iterative process called Quality Function Deployment (QFD). The foundation of the HOQ is the belief that products should be designed to reflect customers’ desires and taste. HOQ is performed by a multidisciplinary team representing marketing, design engineering, manufacturing engineering, and any other functions considered critical by the company. In general, it provides a framework in which all participants can communicate their thoughts about a product. More specifically, HOQ is often used to identify the relationships between requirements based on different viewpoints. There are two issues in analyzing these requirements using HOQ. First, requirements are often described informally using vague terms. However, lack of formal way in interpreting the semantics of these requirements makes it difficult to determine if a realization of the system meets its customer’s needs. Second, identifying relationships between requirements is often time consuming. Sometimes, it is difficult to arrive at a group consensus on a particular relationship between requirements. To address these issues, we have developed a fuzzy logic-based extension to HOQ for capturing imprecise requirements to both facilitate communication of team members and have a formal representation of requirements. Based on this representation, we developed a heuristic inference scheme to reason about the implicit relationships between requirements. We illustrate our approach using a textile mill supply business application. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Fuzzy sets; Decision support systems; House of quality

1. Introduction

World wide competition in today’s global economies has brought significant challenges to any company that wants to meet continuously changing specific requirements of actual and potential customers. Decreasing product cycle time, increasing quality and lowering costs seem to be few of the most critical issues that need to be addressed to stay competitive. A widespread practice in industry to cope with this venture has been the adoption of Quality Function Deployment (QFD) which was developed in Japan by Mitsubishi in 1972. This is a structured format used to translate
customer requirements, broadly defined, into specific product and service characteristics, and ultimately into the processes and systems that provide the valued products and services [2,9].

QFD uses four matrices, also called “houses”, to integrate informational needs. Applications begin with the House Of Quality (HOQ), which is used by a team to understand customer requirements and to translate these requirements into the voice of the engineer [8]. Posterior houses will deploy the requirements up to production requirements. QFD has been successfully applied in many Japanese organizations to improve processes and build competitive advantage [9]. In US, companies have been motivated by the success of Ford in 1983 in the use of QFD. Today, companies are using successfully QFD as a powerful tool that addresses strategic and operational decisions in businesses [6]. Thus, successful translation of customer requirements means: (1) a key criterion in total quality management, (2) enhanced sales and profits while satisfying customers and reducing the cycle time of new product development [8], and (3) emergency of new customer satisfaction practices; some of them are linking customer satisfaction to firm performance and management issues [5].

In spite of the significant number of documented successes with the use of HOQ, there are some companies that have failed in this process; some others have reported mixed experiences using HOQ [6]. Among the most significant identified problems with the use of HOQ are: (1) it is time consuming, (2) the size of the matrices are too big, (3) it is difficult to reach agreement on conflicting technical requirements, and (4) it is difficult to translate and categorize customers’ needs as well as to prioritize customer requirements [6,8,11,16,17]. Our research proposes to handle some of these issues through fuzzy logic based methodology. In particular, we proposed a systematic analysis for representing requirements and assessing conflicting requirements. Our main objective is to facilitate communication for all QFD participants with the use of the proposed analysis tool. QFD is a multiattribute measurement method that brings together major components of an organization and the complex task of capturing customers’ expectations and ultimately delivering customer satisfaction.

Capturing customers requirements is still pursued by traditional qualitative and quantitative methods. Qualitative data, which is vague and imprecise in nature, is used by the experts to assess results from quantitative data [6]. The complete process of using QFD is a complex undertaking: it is a multiexperts and multicriteria decision making process where multifunctional teams are involved.

Based on a previously established foundation for intelligent requirements engineering [18,19], this paper presents a unique decision making tool within the context of QFD and fuzzy logic. Selection of fuzzy logic as a means to represent a QFD methodology seems natural, in particular, when we review Hisdal’s proposition [10]: Fuzzy logic can handle inexact information and linguistic variables in a mathematically well-defined way which simulates the processing of information in natural-language communication. For example, expressions such as: “high competition”, “low interference”, “low impact”, or “high collaboration” are imprecise. These sentences in a natural or synthetic language are the values of linguistic variables which represent linguistic concepts such as very low, low, medium, and so on. Thus, a systematic use of words to characterize values of variables, the values of probabilities, the relations between variables, and so on, constitute a linguistic approach usually described as fuzzy logic [1,12,20]. In our approach, we use fuzzy logic to explicitly capture the customer’s requirements. By so doing, we not only facilitate the communication of different parties in a HOQ team, e.g., customers and engineers, but also have a formal and quantitative representation of requirements. Based on this representation, we identify important relationships, e.g., conflicting relationship, between two requirements. Furthermore, we develop a heuristic inference scheme to reason about the implicit relationships between technical requirements based on the identified relationships between customer’s and technical requirements.

We first describe some relevant constructs on Quality Function Deployment with emphasis on the House of Quality in Section 2. Then, in Section 3, we address foundation of HOQ and its relevance to fuzzy systems. Our fuzzy logic-based scheme for inferring requirements relationships in
the HOQ framework is presented in Section 4. In Section 5, we illustrate our approach using a textile mill supply business application.

2. Quality function deployment

2.1. QFD process

QFD employs several matrices (usually four) to clearly establish relationships between company functions and customer satisfaction. These matrices are based on the “what-how” matrix, which is called HOQ. QFD is an iterative process performed by a multifunctional team [8]. The team will use the matrices to translate customer needs to process step specifications, see Fig. 1. The matrices explicitly relate the data produced in one stage of the process to the decisions that must be made at the next process stage [6]. Product planning is the first matrix. Customers desires, in customers’ own words (whats), are determined and translated into technical description (hows) or proposed performance characteristics of the product. This first matrix is described in detail in Section 2.2 and is also the basis of our fuzzy logic-based requirements analysis methodology. The second QFD matrix relates potential product features to the delivery of performance characteristics. Process characteristics and production requirements are related to engineering and marketing characteristics with the third and fourth matrices.

2.2. HOQ’s general description and process

The different parts of the quality matrix, HOQ, are shown in Fig. 2. We describe the HOQ and its

Fig. 1. Quality function deployment process.
process by following the approaches suggested by Brown [3], and Griffin and Hauser [6].

**Step 1**: Identify the WHATs. This requires a sequence of well organized activities: the determination of customer needs and their arrangement, the assignment of priorities to customer attributes, and the evaluation of customer's perception. The wanted benefits in a product or service in the customer's own words are customer needs and usually called customer attributes (CA) or "what", area (a) in Fig. 2. The CAs are usually determined by qualitative research with one-on-one interviews and/or focus groups [6]. The CAs should be arranged in a hierarchy of levels to facilitate analysis and interpretation. This first part of step 1 relies significantly on the team members' expertise [6,11,15]. In assigning priorities to CAs, the team balances efforts to accomplish those needs that add value to the customer. The priorities are usually indicated in the area designated as (c) in Fig. 2.

The process of determining these priorities is based on team members' direct experience with a variety of marketing research techniques [9]. Customer perceptions are obtained and presented on the right-hand side of the matrix in Fig. 2, area (b). A clear understanding of how existing products/services (company's brand and competitors') are fulfilling CAs provides key information to the multidisciplinary team carrying out other steps of the HOQ and of the whole QFD process.

**Step 2**: Determination of HOWs. Technical requirements (TR) are specified as the "how" of the HOQ and also called measurable requirements. TRs are identified by a multidisciplinary team [8] and positioned on the area marked as (d) on the matrix diagram, Fig. 2.

**Step 3**: Preparation of the relationship matrix. The team judges which TRs impact which CAs and up to what degree. Team consensus is beneficial [9]. The relationships can be positive or
negative, strong or weak; symbols to represent relationships are not standardized. The relationship matrix is the area identified as (e) in Fig. 2.

Step 4: Elaboration of the correlation matrix. The physical relationships among the technical requirements are specified on an array known as “the roof matrix” and identified as (f) in Fig. 2. This HOQ’s step helps team members, specially engineers, to keep track of collateral TRs requiring improvements and/or of TRs where tradeoffs are necessary. Positive and negative correlation between pairs of TRs as well as the strength of the relationships are indicated in this matrix.

Step 5: Other measurements. The team often estimates cost, feasibility and technical difficulty for change in each of the TRs. Technical assessment or difficulty is identified by (h) in Fig. 2. The company’s targets, area (g), hold objective measures which should reflect the link among CAs, TRs, and customer assessments.

Step 6: Action plan. The weights of the TRs, identified as area (i), are placed at the base of the quality matrix. These weights are one of the main outputs of the HOQ, and are determined by

\[
\text{Weight}(TR)_i = V(TR)_1 \times \text{Imp}(CA_1) + V(TR)_2 \times \text{Imp}(CA_2) + \cdots + V(TR)_n \times \text{Imp}(CA_n),
\]

where \( V(TR)_n \) is the correlation value of \( TR_i \) with \( CA_n \), and \( \text{Imp}(CA_n) \) represents the importance or priority of \( CA_n \).

3. Fuzzy systems and HOQ

The foundation of the HOQ is the belief that products/services should be designed to reflect customers’ desires and preferences; thus, people from marketing, design engineering, R&D, manufacturing engineering, and other company functions must work closely together as a team from the time a product/service is first conceived until it is delivered to the customers to satisfy their requirements. This implies a very interrelated set of steps with a multidisciplinary team having technical and managerial expertise. This becomes critical to the delivery of desired product/service in a competitive and uncertain global market.

Hauser and Clausing [9] indicated that there is nothing mysterious or particularly difficult about HOQ. We find that the process is complex, surrounded by subjective judgments, vagueness at times, and uncertainty. Several issues that illustrate our argument are outlined. First, almost all the steps of the HOQ depends on experts’ knowledge. We note that most of the necessary activities to translate customer’s desires into CAs and the relative importance of each CA require team members’ expertise [6,9]. For instance, subjective interpretation arises in understanding what the customers really mean in their descriptions through one-on-one interviews or focus groups. There is also subjectivity and vagueness in the translation of customer perceptions of a company’s brand and the competitors’, specially if the team is identifying opportunities for improvements. These activities require team consensus and can generate nonproductive arguments during meeting [16]. Additional experts’ interpretation is required in the selection of priorities for each CA; these priorities are in general determined by classical statistical techniques; nevertheless, final decision on priorities is influenced by the knowledge of the team members and can be a balancing art of parameters that are neither measured nor known to others but experts.

Second, many steps in the HOQ process are significantly complex; thus conflict resolution of technical issues could be time consuming and generate human conflicts among team members. For example, to develop the technical requirements that affect the CAs requires a systematic, patient, and brainstorming analysis by the team; this step, itself, requires the coordination of a significant large number of parameters. To expand further, complexity and vagueness are present in the development of the relationship and the correlation matrices. Arriving at a consensus of a large number of parameters is time consuming, at times frustrating and sometimes efforts are completely dropped. In scenarios of technical conflicts, cost analysis and team members’ expertise are utilized to assess tradeoffs that have the potential to deliver customer satisfaction. We believe that
fuzzy logic can alleviate some of the problems because fuzzy logic has been well-known for its capability of representing semantics of linguistic terms [21–23]. For example, using fuzzy logic to capture the meaning of linguistic terms not only allows different parties to communicate in natural language but also facilitate expression of customer’s needs and expert’s knowledge. Furthermore, having the formal representation of customer’s requirements, we can analyze requirements and identify the relationships between requirements, e.g., conflicting and cooperative relationships. Also, it is desirable to develop an inference scheme to reason about the implicit relationships between requirements since the identification process is tedious. Once the conflicting requirements are identified, we can develop a systematic tradeoff analysis to assist customers in making the tradeoff decisions [19].

Our issues of concerns are supported by the findings of King [11], Sullivan [16], Griffin et al. [5], Vasilash [17], Hales [7], and Maduri [14], and can be summarized as: (1) Quality charts can go too big and thus become too complex for interpretation. (2) Customers demanded quality is difficult to translate into workable customer attributes and technical requirements. (3) Development of the relationship and correlation matrices is very difficult and team members’ perceptions and judgments can over or under estimate some interdependencies. (4) Correcting mistakes or changing direction once a project has been started can be very difficult due to interdependencies and the subjective judgments involved.

4. A fuzzy logic-based assistance to HOQ

4.1. Representation of requirements

The foremost effort in developing a system (product) is to know what a customer wants. Some requirements impose constraints on the development process such as cost for constructing the system and resources that could be consumed in the development process. An example of such requirements is:
- The cost for developing the rubber belt for a textile spinning frame should be low.

Some requirements impose constraints on the realization of a system and describe the desired features of a product, such as consistency and reliability. For example,
- The consistency of yarn quality should be high.

Requirements usually are expressed in natural language which is vague and ambiguous in nature. It is still desirable to express requirements using linguistic terms because it facilitates communication among different parties, e.g., customers and requirements analysts. However, a computer-based system requires an explicit formal semantics in order to analyze requirements. Therefore, it would be better if we can both express requirements using linguistic terms and have a formal representation underlying these linguistic terms. Fuzzy logic has been well-known for its capability of formally representing the semantics of linguistic terms [21–23]. Hence, we adopt fuzzy logic for representing requirements.

We view requirements as constraints on the system that we would like to build. An imprecise requirement imposes an elastic constraint on the system (i.e., a constraint that can be partially satisfied). The universe being constrained by a requirement is called its domain. A typical domain for requirements is the domain containing all possible products under consideration. Formally, the constraint imposed by a requirement \( R \) is represented as a satisfaction function, denoted as \( \text{Sat}_R \), 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:

\[
\text{Sat}_R : D \rightarrow [0, 1].
\]  

(2)

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

The canonical form in Zadeh’s test score semantics is used as a basis for expressing requirements [21]. The representation of requirements on a system product in canonical form is established below [13,18].

**Definition 1.** Let \( R \) be an requirement on system product in canonical form \( R: A_i(p) \) is \( B \), where \( p \) is
a system product, \( A_i \) is a property of the product, \( B \) is a fuzzy set. Then
\[
\text{Sat}_R(p) = \mu_B(A_i(p)).
\]

To illustrate Definition 1, let us consider the requirement \( R \): “The life expectancy of the belt should be high”. This requirement can be represented in canonical form as follows.

- \( R: \) Life_Expectancy\((p) \) should be \( \text{HIGH} \),

where \( \text{HIGH} \) is a fuzzy set. We can then characterize the satisfaction function of requirement \( R \) using the membership function of fuzzy set \( \text{HIGH} \).

One possible membership function for the fuzzy set \( \text{HIGH} \) is shown in Fig. 3. In Fig. 3, a product (belt) \( p \) that has a life expectancy above 6 weeks fully satisfies requirement \( R \), i.e., \( \text{Sat}_R(p) = 1.0 \).

If a product \( p \) has a 5-week life expectancy, it satisfies requirement \( R \) to a degree of 0.5, i.e., \( \text{Sat}_R(p) = 0.5 \).

Defining membership functions of linguistic terms requires efforts. In order to address this issue, we have developed a scheme to assist customers in identifying the structures as well as the parameters of membership functions in requirements in a related paper [19]. The approach is based on a systematic tradeoff analysis framework in decision science. We have developed techniques that assist customers in identifying linear and nonlinear structures of membership functions. Once we know the structures, several approaches have been developed to assess the parameters of membership functions. Readers are referred to [19] for more details.

### 4.2. Identification of requirements relationships

Considering different impacts of satisfying a requirement on the satisfaction degree of another requirement, we have identified four types of significant relationships between requirements: (1) mutually exclusive, (2) irrelevant, (3) conflicting, and (4) cooperative [13]. Two requirements are called mutually exclusive if they cannot be satisfied (partially or completely) at all at the same time.

That is, if one requirement is satisfied to some degree, the other requirement cannot be satisfied at all, and vice versa. Two requirements are called irrelevant if satisfaction of one requirement does not have any impact on the satisfaction of the other requirement. That is, any change of satisfaction of one requirement will not affect the satisfaction of the other requirement. Two requirements are said to be conflicting with each other if an increase in the degree one requirement is satisfied often decreases the degree the other is satisfied. 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. On the other hand, two requirements are said to be cooperative if an increase in the degree one requirement is satisfied often increases the

![Fig. 3. An example HIGH membership function.](image-url)
degree the other is satisfied. If an increase in the satisfaction degree of one requirement always increases the satisfaction degree of the other, they are said to be completely cooperative.

Two requirements may be completely conflicting or partially conflicting, as shown in Fig. 4. We should point out that the horizontal axis represents all possible products ordered in the ascending order of their satisfaction degrees in $R_2$. Although the set of all possible products is discretely finite, it is shown as a continuum in the figure for convenience. In order to characterize conflicting requirements, we formally define the conflicting degree between two requirements. The conflicting degree considers not only the number of conflicting cases but also the extent of conflict in each case. The definition of conflicting degree is as follows.

**Definition 2** (conflicting degree between requirements). Let $R_1$ and $R_2$ be two requirements of a target system in the domain $SP$ of system products. Let $U$ denote the set of product pairs, in which an increase in the satisfaction degree of one requirement decreases the satisfaction degree of the other, that is,

$$U = \{(p_i, p_j) | p_i, p_j \in SP, (Sat_{R_1}(p_i) - Sat_{R_1}(p_j)) \times ((Sat_{R_2}(p_i) - Sat_{R_2}(p_j)) < 0\}.$$

The degree $R_1$ and $R_2$ are conflicting, denoted as $\text{conf}(R_1, R_2)$, is defined as

$$\text{conf}(R_1, R_2) = \frac{\sum_{(p_i, p_j) \in U} [(Sat_{R_1}(p_i) - Sat_{R_1}(p_j)) \times (Sat_{R_2}(p_i) - Sat_{R_2}(p_j))] \times (Sat_{R_1}(p_i) - Sat_{R_1}(p_j))}{\sum_{(p_i, p_j) \in U} [(Sat_{R_1}(p_i) - Sat_{R_1}(p_j)) \times (Sat_{R_2}(p_i) - Sat_{R_2}(p_j))]}.$$

Based on the definition of conflicting degree, it is easy to see two requirements are completely conflicting whenever their conflicting degree is one. We can define crisp qualitative conflicting relationships as follows.

**Definition 3** (crisp conflicting requirements). Two requirements $R_1$ and $R_2$ are said to be conflicting if and only if

$$\text{conf}(R_1, R_2) \geq 0.5.$$

Fuzzy conflicting relationships can relax the conditions of the crisp conflicting relationship using fuzzy terms such as strong, medium, weak, etc. Hence, one can define terms such as "strong conflict", "medium conflict", and "weak conflict" using satisfaction functions. An example of fuzzy conflicting relationships is shown in Fig. 5. In this example, when two requirements have conflicting degree 0.5, we are very sure that they are weak conflicting since their satisfaction degree in membership function Weak Conflict is 1.0, and are confident in saying they are not strong conflicting since the degree of satisfaction in membership function Strong Conflict is 0. We are somewhat sure that these two requirements are medium conflicting since their degree of satisfaction in membership function Medium Conflict is 0.6.

![Fig. 4. Conflicting requirements.](image-url)
Similar to the cases of conflicting requirements, two requirements may be completely cooperative or partially cooperative, as shown in Fig. 6. In order to characterize cooperative requirements, we formally define the cooperative degree between two requirements as follows.

**Definition 4** (cooperative degree between requirements). Let $R_1$ and $R_2$ be two requirements of a target system in the domain $SP$ of system products. Let $\mathcal{V}$ denote the set of product pairs, in which an increase in the satisfaction degree of one requirement also increases the satisfaction degree of the other, that is,

$$\mathcal{V} = \{ \langle p_i, p_j \rangle | p_i, p_j \in SP, (Sat_{R_1}(p_i) - Sat_{R_1}(p_j)) \times (Sat_{R_2}(p_i) - Sat_{R_2}(p_j)) > 0 \}.$$

The degree $R_1$ and $R_2$ are cooperative, denoted as $\text{coop}(R_1, R_2)$, is defined as

$$\text{coop}(R_1, R_2) = \frac{\sum_{\langle p_i, p_j \rangle \in \mathcal{V}} |Sat_{R_1}(p_i) - Sat_{R_1}(p_j)| \times |Sat_{R_2}(p_i) - Sat_{R_2}(p_j)|}{\sum_{\langle p_i, p_j \rangle \in \mathcal{V}} |Sat_{R_1}(p_k) - Sat_{R_1}(p_k)| \times |Sat_{R_2}(p_k) - Sat_{R_2}(p_k)|}.$$  

(4)

Based on the definition of cooperative degree, it is easy to see two requirements are completely cooperative whenever their cooperative degree is one. We can define crisp qualitative cooperative relationships as follows.

Fig. 5. An example of fuzzy conflicting requirements.

Fig. 6. Cooperative requirements.
**Definition 5** (crisp cooperative requirements). Two requirements $R_1$ and $R_2$ are said to be cooperative if and only if

$$\text{coop}(R_1, R_2) \geq 0.5.$$ 

Using the similar approach as to define fuzzy conflicting relationships, one can relax the conditions of crisp cooperative relationships and define fuzzy cooperative relationships such as “strong cooperation”, “medium cooperation”, and “weak cooperation”.

### 4.3. A reasoning scheme for inferring requirements relationships

Many relationships between requirements are implicit and difficult to identify. In order to help the HOQ team members to identify various relationships between requirements, we need to develop a reasoning scheme that enables the inference of requirement relationships in HOQ. Such an inference scheme is particularly desirable for the process of HOQ because (1) the identification of the relationship and correlation relationship matrices is tedious and time consuming; (2) it is sometimes hard for a group to reach a consensus on a particular relationship between requirements.

Based on the identified relationships between requirements, we have developed a reasoning mechanism to discover the implicit relationships between requirements. We assume that the analyzed requirements share the same domain. This assumption is important in identifying relationships since different relationships may exist between two requirements due to different domains. For example, the requirements for manufacturing a rubber belt for a textile spinning frame may include:

- **R1**: the total cost for manufacturing the rubber belts should be low.
- **R2**: the quality of the rubber belts manufactured should be high.
- **R3**: the quantity of the rubber belts manufactured should be large.

Requirements $R_2$ (quality) and $R_3$ (quantity) may not have conflicting relationships since we can have different budgets for different plans. However, once we have a fixed budget, it is easy to see requirements $R_2$ and $R_3$ are conflicting. We first present three theorems for inferring completely conflicting and cooperative relationships. They can be proved easily based on the definitions of cooperative and conflicting relationships.

**Theorem 1.** Let $D$ be a domain shared by three requirements $R_1$, $R_2$, and $R_3$. If $R_1$ is completely cooperative with $R_2$ in $D$ and $R_2$ is completely cooperative with $R_3$ in $D$, then $R_1$ is completely cooperative with $R_3$ in $D$.

**Theorem 2.** Let $D$ be a domain shared by three requirements $R_1$, $R_2$, and $R_3$. If $R_1$ is completely cooperative with $R_2$ in $D$ and $R_2$ is completely conflicting with $R_3$ in $D$, then $R_1$ is completely conflicting with $R_3$ in $D$.

**Theorem 3.** Let $D$ be a domain shared by three requirements $R_1$, $R_2$, and $R_3$. If $R_1$ is completely conflicting with $R_2$ in $D$ and $R_2$ is completely conflicting with $R_3$ in $D$, then $R_1$ is completely cooperative with $R_3$ in $D$.

From Theorem 1, we have shown that completely cooperative relationship in a domain is transitive. On the other hand, Theorem 3 indicates that completely conflicting relationship in a domain is not transitive.

The above theorems deal with requirements that are either completely cooperative or completely conflicting. It is obviously desirable to reason about requirements that are partially cooperative or conflicting. Based on the definitions of conflicting degree and cooperative degree and on the development of theorems, we developed the following fuzzy if-then heuristic rules to reason about partially conflicting or partially cooperative relationships. It is important to point out that terms such as weak, medium, and strong are fuzzy relationships mentioned in Section 4.2. From two cooperative relationships, we developed fuzzy if-then rules to infer cooperative relationships.

- If $\text{coop}(R_1, R_2)$ is strong, $\text{coop}(R_2, R_3)$ is strong, and $R_1$ and $R_3$ are not irrelevant, then $\text{coop}(R_1, R_3)$ is strong.
The rationale for deriving this rule is as follows. If we increase satisfaction of \( R_1 \), the satisfaction of \( R_2 \) will be increased strongly since coop\((R_1, R_2)\) is strong. Then the satisfaction of \( R_3 \) is strongly increased since coop\((R_2, R_3)\) is strong. Therefore, we can infer coop\((R_1, R_3)\) is strong. Similar rationales could be used to derive other rules, which are listed in Table 1. The entry in the table is a term that describes the inferred value of coop\((R_1, R_3)\). The assumptions in these rules are (1) \( R_1, R_2, \) and \( R_3 \) share the same domain, and (2) \( R_1 \) and \( R_3 \) are not irrelevant.

Similarly, we developed fuzzy if–then rules to infer cooperative relationships from two conflicting relationships. These rules are shown in Table 2.

The difference between Table 1 and Table 2 is the identified relationships between requirements. In Table 1, the identified relationships are cooperative. However, in Table 2, the identified relationships are conflicting. An example of the rules in Table 2 is

- If conf\((R_1, R_2)\) is strong, conf\((R_2, R_3)\) is strong, and \( R_1 \) and \( R_3 \) are not irrelevant, then coop\((R_1, R_3)\) is strong.

The rationale for deriving this rule is as follows. If we increase the satisfaction degree of \( R_1 \), it will strongly decrease the satisfaction degree of \( R_2 \) because conf\((R_1, R_2)\) is strong. Subsequently, it will strongly increase the satisfaction degree of \( R_3 \) since conf\((R_2, R_3)\) is strong. Hence, coop\((R_1, R_3)\) is strong.

We can also infer relationships based on a conflicting relationship and a cooperative relationship. The rules for this case are shown in Table 3. We derived these rules based on the same assumptions and the similar rationales that we used in the above tables. An example of these rules is as follows.

- If coop\((R_1, R_2)\) is strong, conf\((R_2, R_3)\) is strong, and \( R_1 \) and \( R_3 \) are not irrelevant, then conf\((R_1, R_3)\) is strong.

5. A manufacturing example

An example of HOQ regarding a textile mill supply business developed by Cook et al. [4] is used in this section to illustrate the proposed inference scheme. The company manufactures rubber cord reinforced endless belts (aprons), which was used in the drafting zone on textile spinning frames. Spinning frames take fiber and spin it into thread, which is then used to make fabric. In particular, this company has decided to develop an apron for a spinning frame that runs 10 times faster than conventional spinning frames.

In developing the HOQ for this example, the process explained in Section 2 was followed. Then, experts in the multidisciplinary team identified the relationships between each pair of customer’s requirements (CAIs) and technical requirements (TRs), which are shown in the relationship matrix of Fig. 7. A blank entry in the matrix denotes “no relationship”. Cooperative and conflicting relationships are classified into three levels, weak, medium, or strong. For example, “Nontagging” customer requirement has a medium cooperative relationship with “Width Value” technical requirement and has a strong conflicting relationship with “Width Variability” technical requirement.

Based on the inference scheme developed in Section 3, we propose to partially automate the identification process of the relationship and correlation matrices. Many of the relationships between TRs can be inferred from the identified relationships between customer’s requirements (CAIs) and technical requirements (TRs), which are
shown in the roof correlation matrix of Fig. 7. For instance, technical requirements TR3 “Wall Thickness Variability” and TR7 “ID (Inner Diameter) Dimension Value” have strong conflicting and medium cooperative relationships with CA8 “Consistent Yarn Quality” requirement, respectively. Also, these relationships are not irrelevant. Considering the rules in Table 3, we can infer that the relationship between TR3 and TR7 is medium conflict. Moreover, technical requirements TR3 and TR7 have medium conflicting and weak cooperative relationships with CA9 “Proper Tracking”, respectively. Therefore, from the rules in Table 3, we can infer that the relationship between TR3 and TR7 is weakly conflicting. We need to aggregate results inferred from these two rules. The choice of aggregation is usually application-dependent [23]. In this case, more research should be done in the future to select the best aggregation operator. For illustration, the max operator is chosen. Hence, the relationship between technical requirements TR3 and TR7 are indicated as medium conflict. We should also point out if requirements have been prioritized, an alternative aggregation scheme is needed (e.g., weighted-max or weighted-sum) to take into account different priorities of requirements.

The proposed inference scheme for developing the relationships between technical requirements has three advantages. First, it can reduce the time and efforts for the team to construct the correlation matrices of HOQ. Second, it can help in identifying some requirements relationships which may be overlooked by experts. For example, the relationship between requirements TR3 Wall Thickness Variability and TR20 “Homogeneity” were not identified by experts [4]. However, based on the rules in Table 3, TR3 has strong conflicting relationship with TR20 because TR3 and TR20 have strongly conflicting and cooperative with CA8 “Consistent Yarn Quality”, respectively. Since variability of wall thickness implies differences, homogeneity would be a direct contradiction of variability and would be inversely influenced by variability. This relationship could have been overlooked by the experts due to the tedious nature of the task in manually constructing the correlation matrix. Third, the team’s efforts may be used more efficiently in discussing strategic opportunities by analyzing the output from this inference algorithm.

Some of the inference results obtained with the proposed algorithm are different from those of experts’ identification in the example. For in-

<table>
<thead>
<tr>
<th>coop($R_1$, $R_2$)</th>
<th>conf($R_1$, $R_3$)</th>
<th>coop($R_1$, $R_2$)</th>
<th>conf($R_1$, $R_3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>coop($R_1$, $R_2$)</th>
<th>conf($R_1$, $R_3$)</th>
<th>coop($R_1$, $R_2$)</th>
<th>conf($R_1$, $R_3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
<td>Strong</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
<td>Weak</td>
</tr>
</tbody>
</table>
Fig. 7. Inferred relationships between technical requirements.
stance, a conflicting relationship between TR3 Wall Thickness Variability and TR6 “ID Dimension Variability” was identified in the example by Rao [4]; on the other hand, the proposed inference algorithm showed a cooperative relationship between these two technical requirements. We believe that the discrepancies are due to different assumptions made in the identification process. In Rao’s results, the relationships between the variables about the product are identified. In the inference scheme, the relationships are identified based on the correlation of changes of satisfaction degrees of requirements. For instance, the decrease of wall thickness variability will decrease inner diameter dimension variability. Therefore, the increase of satisfaction degree of the requirement “Wall Thickness Variability Should Be Low” will increase the satisfaction degree of requirement “ID Dimension Variability Should Be Low”. Hence, our inference scheme inferred a cooperative relationship between them.

6. Conclusion

HOQ has been widely used at industry in Japan and American. The foundation of the HOQ is the belief that products should be designed to reflect customers’ desires and taste which are usually described in natural language. However, lack of precision in interpreting the semantics of these requirements makes it difficult to determine if a realization of the systems meets its customer’s needs. Moreover, it is tedious and time consuming to identify the relationships between requirements in a HOQ process. Sometimes, it is difficult to arrive at a group consensus on a particular relationship between requirements. These difficulties could be alleviated by providing appropriate tools to the teams. These tools will assist the team in identifying conflicts between requirements. To address these issues, we have described our fuzzy logic-based methodology to business decision making within the context of the HOQ. In order to facilitate communication between different team members and have a formal representation of requirements, we use fuzzy logic to capture requirements. Based on this representation, we have formally identified relationships between requirements such as conflicting and cooperative relationships. We also developed an inference scheme to reason about the implicit relationships between requirements. We demonstrate the inference scheme using a textile mill supply application. The benefit of our method is that it assists all participants to understand the meanings and implications of other team members’ requirements, assists team members to identify conflicting requirements, and facilitates a more effective communication and strategic business decision making involving all members of a HOQ team.

References