Challenges for Perceptual Computer Applications and How They Were Overcome

Jerry M. Mendel
University of Southern California, USA

Dongrui Wu
GE Global Research, USA

Abstract—*Perceptual Computing* is a methodology of Computing with Words (CWW) to assist humans in making subjective judgments. This article introduces the *Perceptual Computer* (Per-C), our instantiation of Perceptual Computing. Per-C consists of three components: encoder, CWW engine and decoder. Perceptions–words–activate the Per-C and are the Per-C output (along with data); so, it is possible for a human to interact with the Per-C using just a vocabulary. The encoder transforms words into fuzzy sets and leads to a *codebook*–words with their associated fuzzy set models. The outputs of the encoder activate a CWW engine whose output is one or more other fuzzy sets, which are then mapped by the decoder into a recommendation (subjective judgment) with supporting data. The recommendation may be in the form of a word, group of similar words, rank or class. When the Per-C was applied to actual applications, challenges occurred that had to be overcome. In this article we describe three applications (investment decision making, social judgment making, and distributed decision making), the challenges encountered and how they were overcome.

I. Introduction

The phrase *Computing With Words* (CWW), originated by Zadeh in 1996 [31], equates fuzzy logic to it (see Box 1). Oh, if it were only that simple! The 2010 Computational Intelligence Magazine article [9] presents seven points of view of what CWW means, and, it should be clear to anyone who reads this article that, there is no consensus on what it means. Additionally, the Foreword to the June 2010 Special Section of the *IEEE Transactions on Fuzzy Systems* on CWW [8] provides further thoughts about CWW by Zadeh, as well as some new important distinctions between two levels of CWW ("basic" and "advanced" CWW). We conclude from all of this that the entropy level of "CWW" is quite high, which presented to us a fantastic field to work in.

We think it is now fair to state that CWW is a broad overarching high-level paradigm that makes it very rich because it is open to different interpretations and different instantiations, but all such interpretations require fuzzy logic to implement them.

Digital Object Identifier 10.1109/MCI.2012.2200627
Date of publication: 12 July 2012

For more than a decade we have been interested in CWW to assist humans in making subjective judgments, and call the methodology for doing this *Perceptual Computing* [19]. Such judgments, are personal opinions that have been influenced by one's personal views, experiences and/or background and can also be interpreted as personal assessments of the levels of variables of interest, made using a mixture of qualitative and quantitative information. Using Zadeh's distinction between basic and advanced CWW (see Box 1), Perceptual Computing at present is basic CWW.

Our instantiation of Perceptual Computing is called a *Perceptual Computer* (Per-C) [10,] [12], [13]. It has the architecture that is depicted in Fig. 1, and consists of three components: encoder, CWW engine and decoder. Perceptions-words-activate the Per-C and are the Per-C output (along with data); so, it is possible for a human to interact with the Per-C using just a vocabulary. A vocabulary is application (context) dependent, and must be large enough so that it lets the end-user interact with the Per-C in a user-friendly manner. The encoder transforms words into fuzzy sets (FSs) and leads to a *codebook*–words with their associated FS models. The outputs of the encoder activate a CWW engine whose output is one or more other

1556-603X/12/$31.00©2012IEEE

FSs, which are then mapped by the decoder into a recommendation (subjective judgment) with supporting data. The recommendation may be in the form of a word, group of similar words, rank or class.

The Per-C is an interactive device that can aid people in making subjective judgments. It can propagate random and linguistic uncertainties into the subjective judgment, but in a way that can be modeled and observed by the judgment maker. The Per-C is not a single device for all problems, but is instead a device that must be designed for each specific problem by using the methodology of Perceptual Computing, a methodology that is described in the next section.

We agree with Zadeh that fuzzy logic should be used for CWW, and so it is used as the mathematical vehicle for the Per-C, but not the ordinary fuzzy logic. Because *words can mean different things to different people*, it is important to use an FS model that lets us capture word uncertainties. We use interval type-2 fuzzy sets (IT2 FSs) (Box 2) and fuzzy logic because they can do this. Detailed discussions about this have already appeared in [14] and are not repeated here.

There were many challenges in the implementation of Perceptual Computing. Some obstacles are common to all

Box 1: Computing with Words

According to Zadeh [31]–[33], "CWW is a methodology in which the objects of computation are words and propositions drawn from a natural language. [It is] inspired by the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. CWW may have an important bearing on how humans… make perception-based rational decisions in an environment of imprecision, uncertainty and partial truth." He did not mean that computers would actually compute using words—single words or phrases—rather than numbers. He meant that computers would be activated by words, which would be converted into a mathematical representation using fuzzy sets (FSs), and that these FSs would be mapped by a CWW engine into some other FS, after which the latter would be converted back into a word.

More recently, Zadeh [9] has distinguished two kinds of CWWs, *basic* (or Level 1) and *advanced* (or Level 2). According to Zadeh: "In basic CWW the carriers of information are numbers and words. In advanced CWW, the carriers of information are numbers, words and propositions."

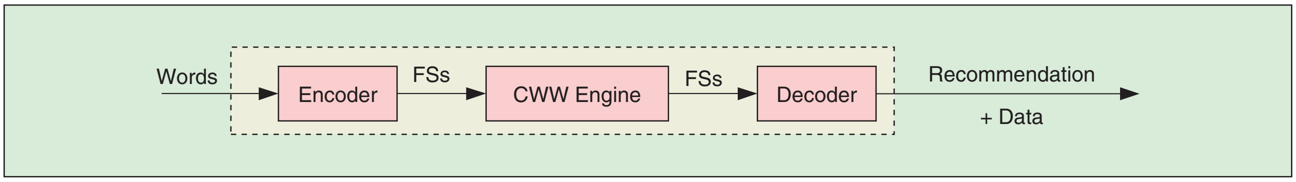

FIGURE 1 Architecture for the Perceptual Computer (Per-C).

applications, whereas others are not. This article explains what the obstacles are and how they were overcome. The ones that are application-dependent are explained in the context of three specific applications: Investment decision making, social judgment making, and distributed decision-making. These applications are explained in more detail in Section III.

Definition 1: The *centroid* of an IT2 FS $\tilde{A}$, $C_{\tilde{A}}$, is an interval of numbers $[c_l, c_r]$, where

$$c_l = \min_{\forall \mu(x_i) \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^{N} x_i \mu(x_i)}{\sum_{i=1}^{N} \mu(x_i)}$$

$$c_r = \max_{\forall \mu(x_i) \in [\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)]} \frac{\sum_{i=1}^{N} x_i \mu(x_i)}{\sum_{i=1}^{N} \mu(x_i)}.$$

c_l and c_r are computed by the KM [4] or EKM [28] Algorithms. The more uncertainty in $\tilde{A}$ (i.e., the more area in its FOU), the wider the centroid. The average centroid (center of centroid) of $\tilde{A}$ is defined as

$$c_{\tilde{A}} = (c_l + c_r)/2. \quad \square \tag{1}$$

Box 2: Interval Type-2 Fuzzy Sets

An interval type-2 fuzzy set (IT2 FS) $\tilde{A}$ is described by its *footprint of uncertainty FOU*$(\tilde{A})$ (Fig. 2), which can be thought of as the blurring of a type-1 membership function (MF). The FOU is completely described by its two bounding functions, a lower membership function (LMF) $LMF(\tilde{A}) = \underline{\mu}_{\tilde{A}}(x)$ and an upper membership function (UMF) $UMF(\tilde{A}) = \bar{\mu}_{\tilde{A}}(x)$, both of which are type-1 FSs. Consequently, it is possible to use type-1 FS mathematics to characterize and work with IT2 FSs. For lots more information about IT2 FSs, see, e.g. [2], [11], [15].

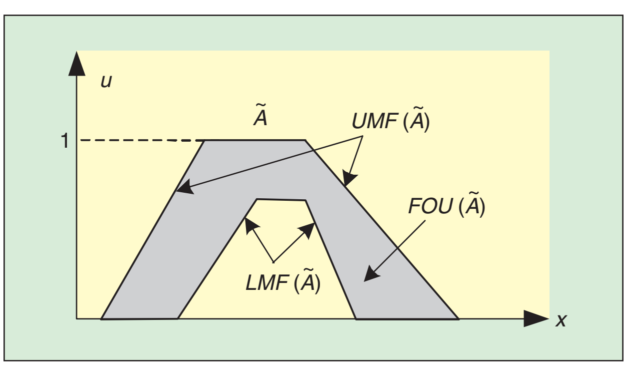

FIGURE 2 FOU for an IT2 FS $\tilde{A}$.

II. Application-Independent Challenges and How They Were Overcome

To operate the Per-C shown in Fig. 1, one needs to be able to construct the encoder, the CWW engine and the decoder, all of which pose some application-independent challenges. Next we will explain these challenges and how they were overcome.

A. Encoder

Our first ***challenge*** (all challenges are summarized in Table 6) in implementing the Per-C was how to transform words into IT2 FSs, i.e., the encoding problem. Our ***solution*** requires: (1) a continuous scale for each variable of interest, and (2) a vocabulary of words that covers the entire scale. Our methods are described for the continuous scale numbered 0–10. One begins by establishing a vocabulary of application-dependent words that is large enough so a person will feel linguistically comfortable interacting with the Per-C. This vocabulary must include subsets of words that feel, to each subject, like they will collectively cover the scale 0–10. The collection of words, $\tilde{W}_i$, in the vocabulary and their IT2 FS models, $FOU(\tilde{W}_i)$, constitutes a codebook for an application $(\mathcal{A})$, that is, Codebook $(\mathcal{A})$= $\{(\tilde{W}_i, FOU(\tilde{W}_i)), i = 1, \dots, N_{\mathcal{A}}\}$.

We then randomize the words in the vocabulary and survey a group of subjects to provide end-point data for the words on the scale. The subjects are asked the following question: *On a scale of 0-10, what are the end-points of an interval that you associate with the word* _? Once enough data intervals (e.g., 30) have been obtained, they can be processed by the Interval Approach (IA) ([7]; see also Box 3) to obtain an IT2 FS model for each word.

B. CWW Engine

Next we consider how to construct the CWW engine, which maps IT2 FSs into IT2 FSs. There are different kinds of CWW engines, e.g.,

1) The *novel weighted average* (NWA) [19]. Aggregation of numerical subcriteria (data, features, decisions, recommendations, judgments, scores, etc.) obtained by using a weighted average of those numbers is quite common and widely used. In many situations, however, providing a single number for either the subcriteria or weights is problematic (there could be uncertainties about them), and it is more meaningful to provide uniformly-weighted intervals, non-uniformly-weighted intervals (T1 FSs),

or words (IT2 FSs), or a mixture of all these, for them. The ***challenge*** was how to aggregate this disparate information. Our ***solution*** was to use the NWA, a weighted average in which at least one subcriterion or weight is not a single real number, but is instead an interval, T1 FS, or an IT2 FS. NWAs include the interval weighted average (IWA), fuzzy weighted average (FWA) [6], and linguistic weighted average (LWA) [25], [26]. More details about the NWAs, especially the LWA, are given in Box 4.

2) *Perceptual reasoning* (PR) [17], [19] [29]. One of the most popular CWW engines uses if-then rules. The use of if-then rules in a Per-C is quite different from their use in most engineering applications of rule-based systems—fuzzy logic systems (FLSs)—because in an FLS the output almost always is a number, whereas the output of the Per-C is a recommendation. For CWW, our ***challenge*** was how to make the output FOU of the if-then rule-based CWW engine resemble the three kinds of FOUs in a CWW codebook. This is so that the decoder can do its job properly (map an FOU into a word in a codebook), and agrees with the adage, "not only do words mean different things to different people, but they must also mean similar things to different people," or else people would not be able to communicate with each other. Our ***solution*** was PR, which consists of two steps: 1) A firing quantity is computed for each rule by computing a scalar Jaccard similarity measure [19], [27] between each input word and its corresponding antecedent word, and, if a rule has p antecedents, then taking the minimum of the p Jaccard similarity measures; and, 2) The IT2 FS consequents of the fired rules are combined using an NWA in which the "weights" are the firing quantities and

The Per-C is an interactive device that can aid people in making subjective judgments.

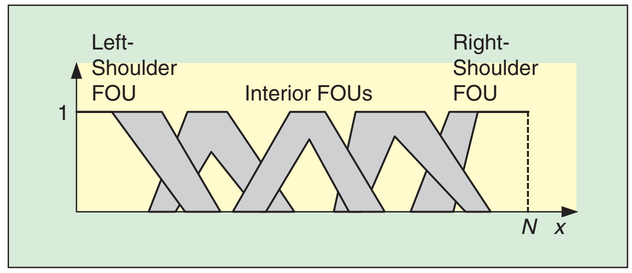

FIGURE 3 Left-shoulder, right-shoulder and interior FOUs, all of whose LMFs and UMFs are piecewise linear [7], [19].

Box 3: Interval Approach (IA)

The IA consists of two parts, a data part and an FS part. In the data part, data intervals that have been collected from a group of subjects are preprocessed, after which data statistics are computed for the surviving intervals. In the FS part, FS uncertainty measures are established for a pre-specified triangle T1 MF (always beginning with the assumption that the FOU is an interior FOU, and, if need be, later switching to a shoulder FOU). Then the parameters of the triangle T1 MF are determined using the data statistics, and the derived T1 MFs are aggregated to form an FOU for a word, and finally a mathematical model is obtained for the FOU.

Only three FOU shapes can be obtained from the IA: interior, left shoulder, and right shoulder, as shown in Fig. 3. A word that is modeled by an interior FOU has a UMF that is a trapezoid and an LMF that is a triangle, but, in general, neither the trapezoid nor the triangle are symmetrical. A word that is modeled as a left- or right-shoulder FOU has trapezoidal upper and lower MFs; however, the legs of the respective two trapezoids are not necessarily parallel. One of the strong points of the IA is that subject data establish which FOU is used to model a word, that is, the FOU is not chosen ahead of time–the data speaks!

An enhanced IA is also now available [30].

Box 4: Novel Weighted Average (NWA)

Because there can be four possible models (numbers, intervals, T1 FSs, and words modeled by IT2 FSs) for subcriteria or weights, there can be 16 different weighted averages. When at least one subcriterion or weight is modeled as an interval, and all other subcriteria or weights are modeled by no more than such a model, the resulting weighted average is called an IWA, denoted $\boldsymbol{Y_{IWA}}$. On the other hand, when at least one subcriterion or weight is modeled as a T1 FS, and all other subcriteria or weights are modeled by no more than such a model, the resulting weighted average is called an FWA, denoted Y_{FWA}. And, finally, when at least one subcriterion or weight is modeled as an IT2 FS, the resulting weighted average is called an LWA. The IWA and FWA are special cases of the LWA; hence, here our focus is only on the *latter*.

The following is a very useful *expressive* way to summarize the LWA:

$$\tilde{Y}_{LWA} = \frac{\sum_{i=1}^{N} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{N} \tilde{W}_i},$$

where $\tilde{X}_i$, the sub-criteria, and $\tilde{W}_i$, the weights, are words modeled by IT2 FSs. $\tilde{Y}_{LWA}$ is also an IT2 FS. This is called an expressive way to summarize the LWA rather than a computational way to summarize the LWA, because the LWA is not computed by multiplying, adding, and dividing IT2 FSs. It is more complicated than that. It is has been shown [19], [25], [26] that the UMF of $\tilde{Y}_{LWA}$ is an FWA [6] of the UMFs of $\tilde{X}_i$ and $\tilde{W}_i$, and the LMF of $\tilde{Y}_{LWA}$ is an FWA of the LMFs of $\tilde{X}_i$ and $\tilde{W}_i$. The LWA and FWA are computed using alpha-cuts and the details of how to do this are found in [19], [25], [26].

TABLE 1 Investment alternatives/investment criteria array.
Example of the linguistic ratings of investment alternatives for investment criteria, provided by an individual.

	INVESTMENT CRITERIA			
	(RISK OF LOSING CAPITAL)	(VULNERABILITY TO INFLATION)	(AMOUNT OF PROFIT RECEIVED)	(LIQUIDITY)
a_1 (COMMODITIES)	HIGH	MORE OR LESS HIGH	VERY HIGH	FAIR
a_2 (STOCKS)	MORE OR LESS HIGH	FAIR	MORE OR LESS HIGH	MORE OR LESS GOOD
a_3 (GOLD)	LOW	LOW	FAIR	GOOD
a_4 (REAL ESTATE)	LOW	VERY LOW	FAIR	BAD
a_5 (LONG-TERM BONDS)	VERY LOW	HIGH	MORE OR LESS LOW	VERY GOOD

the "subcriteria" are the IT2 FS consequents. The result of PR is a convex and normal FOU, which does indeed resemble the three kinds of FOUs in a CWW codebook.

C. Decoder

The ***challenge*** in decoding was in mapping the output of the CWW engine into a recommendation. Our ***solution*** consisted of three kinds of decoders according to three forms of recommendations [19]:

1) *Word*. This is the most typical case, for which a similarity measure that compares the similarity between two FOUs is used. Obviously, if two FOUs have the same shape and are located very close to each other, they should be linguistically similar; or, if they have different shapes and are located close to each other, they should not be linguistically similar; or, if they have the same or different shapes but are not located close to each other they should also not be linguistically similar. We have found that the Jaccard similarity measure [27] provides a crisp numerical similarity measure that agrees with all three of the previous statements.
2) *Rank*. In some decision-making situations, several strategies/candidates are compared at the same time to find the best one(s). Ranking methods are needed to do this. We have used a very simple ranking method that is based on the average centroid of an IT2 FS in (1).
3) *Class*. In some decision making applications, the output of the CWW engine has to be mapped into a class. Classifiers are needed to do this. The classification literature is huge. Our classifiers are based on subsethood [19], which defines the degree of containment of one set in another. The subsethood between two IT2 FSs may either be an interval of numbers or a single number. We prefer to use a single subsethood number for our classifiers.

For details of ranking, similarity and subsethood measures see Chapter 4 of [19].

III. Application-Dependent Challenges and How They Were Overcome

When the methodology of perceptual computing was applied to actual applications, challenges occurred that had to be overcome. In this section we describe some applications, the challenges encountered and how they were overcome.

A. Investment Judgment Advisor (IJA)

The following investment decision application is modified from Tong and Bonissone's example [23].

A private citizen has a moderately large amount of capital that he wishes to invest to his best advantage. He has selected five possible investment areas $\{a_1, a_2, a_3, a_4, a_5\}$ and has four investment criteria $\{c_1, c_2, c_3, c_4\}$ by which to judge them. These are:

- ❏ a_1-the commodity market, a_2-the stock market, a_3-gold, a_4-real estate, and a_5-long-term bonds
- ❏ c_1-the risk of losing the capital sum, c_2-the vulnerability of the capital sum to modification by inflation, c_3-the amount of interest [profit] received, and c_4-the cash realizability of the capital sum [liquidity].

The investor's goal is to decide which investments he should partake in because he does not want to invest in all of them. In order to arrive at his decisions, he must first rate each of the five alternative investment areas for each of the four criteria and assign weights to them. He fills in Table 1 by answering the following questions:

- ❏ *To me, the risk of losing my capital in investment alternative a_j seems to be ______?*
- ❏ *To me, the vulnerability of investment alternative a_j to inflation seems to be ______?*
- ❏ *To me, the amount of profit that I would receive from investment alternative a_j seems to be ______?*
- ❏ *To me, the liquidity of investment alternative a_j seems to be ______?*

He also fills in Table 2 by answering the following questions:

TABLE 2 Example of the linguistic weights for the investment criteria, provided by an individual.

c_1 (RISK OF LOSING CAPITAL)	c_2 (VULNERABILITY TO INFLATION)	c_3 (AMOUNT OF PROFIT RECEIVED)	c_4 (LIQUIDITY)
VERY IMPORTANT	MORE OR LESS IMPORTANT	VERY IMPORTANT	MODERATELY UNIMPORTANT

❑ *The importance that I attach to the investment criterion* c_j *is* _____*?*

His ratings and weights use words and therefore are linguistic. The problem facing the individual investor is how to aggregate the linguistic information in Tables 1 and 2 so as to arrive at *his* preferential ranking of the five investments.

1) *Encoder for the IJA:* We will use the codebook for liquidity as an example. Initially, the following 11 words were chosen to rate *liquidity*:

Very bad, more or less bad, somewhat bad, bad, somewhat fair, fair, very fair, more or less good, somewhat good, good, very good.

During the first four months of 2008 a word survey was conducted and data were collected from 40 adult (male and female) subjects. The IA was applied to the data collected to compute the FOUs; however, we observed that when an individual was given the opportunity to choose a word from the full 11-word codebook and then changed the words to the ones either to the left or to the right of them, there was almost no change in the outputs of the IJA. The individuals who tested the IJA did not like this because they were expecting to see changes when they changed the words. This made the IJA not "user-friendly." This "human factor" was surprising to us because we have always advocated providing the individual who will interact with the Per-C with a large vocabulary in order to make this interaction "user-friendly." So, the ***challenge*** was how to trim a large codebook down to size so that it is more user-friendly, i.e., how to provide an individual with vocabularies that contain *sufficiently dissimilar* words so that when a change is made from one word to another there is a noticeable change in the output of the IJA.

According to several researchers [20], [22], a codebook for making preference judgments should have 5-9 words. In order to accomplish this, the similarity matrix for the 11 words were computed using the Jaccard similarity measure, as shown in Table 3. Our ***solution*** was to start from the left column of the similarity matrix and to remove all of the words to which it is similar to degree > 0.6. Beginning with *Very Bad*, observe that it is not similar to any word with degree > 0.6; so, it is kept in the user-friendly codebook and we move to the next word *Bad*. Observe that it is similar to *More* or *Less Bad* to degree 0.78; hence, *More or Less Bad* is eliminated. There are no other words in the row for *Bad* for which the similarity is > 0.6; hence, no other words are eliminated, *Bad* is kept in the user-friendly codebook, and we move next to the word *Somewhat Bad*. Focusing on the elements on the right-hand side of the diagonal element in the row for *Somewhat Bad*, observe that *Somewhat Bad* is not similar to any other words to degree > 0.6; hence, no words are eliminated, *Somewhat Bad* is kept in the user-friendly codebook, and we move next to the word *Fair*. Proceeding in this way through the rest of the similarity matrix, the following user-friendly seven-word codebook was obtained:

Very bad, bad, somewhat bad, fair, somewhat good, good, very good.

2) *CWW Engine for the IJA:* The IJA uses an LWA to aggregate the results for each of the rows in Table 1. Observe that two of the investment criteria have a positive connotation—*amount of profit received* and *liquidity*—and two have a negative connotation—*risk of losing capital* and *vulnerability to inflation*. "Positive connotation" means that an investor generally thinks positively about *amount of profit received* and *liquidity* (i.e., the more the better) whereas "negative connotation" means that an investor generally thinks negatively about *risk of losing capital* and *vulnerability to inflation* (i.e., the less the better). The ***challenge*** here was how sub-criteria which have negative connotations and whose inputs are words are handled.

Our ***solution*** was that a small-sounding word should be replaced by a large-sounding word, and vice versa. This kind of word replacement is essentially the well-known idea of an *antonym* [5]. In this article the most basic *antonym* definition is used [5], i.e.,

$$\mu_{10-A}(x) = \mu_A(10 - x), \ \forall x, \tag{2}$$

where $10 - A$ is the antonym of the T1 FS A, and 10 is the right end of the domain of all FSs used for the application. The definition in (2) can easily be extended to IT2 FSs, i.e.,

TABLE 3 Similarity matrix for the 11-word vocabulary. The words that are similar to degree > 0.6 are underlined, starting from the left-most word VB.

WORD	VB	B	MLB	SB	F	SF	VF	SG	MLG	G	VG
VERY BAD (VB)	1	.29	.27	.17	.04	.03	.03	0	0	0	0
BAD (B)	.29	1	.78	.56	.15	.14	.14	.03	.01	.01	0
MORE OR LESS BAD (MLB)	.27	.78	1	.54	.11	.11	.11	.01	0	0	0
SOMEWHAT BAD (SB)	.17	.56	.54	1	.23	.22	.22	.06	.03	.02	0
FAIR (F)	.04	.15	.11	.23	1	.88	.87	.49	.35	.30	.1
SOMEWHAT FAIR (SF)	.03	.14	.11	.22	.88	1	.99	.58	.43	.38	0
VERY FAIR (VF)	.03	.14	.11	.22	.87	.99	1	.59	.44	.38	0
SOMEWHAT GOOD (SG)	0	.03	.01	.06	.49	.58	.59	1	.64	.53	.28
MORE OR LESS GOOD (MLG)	0	.01	0	.03	.35	.43	.44	.64	1	.81	.4
GOOD (G)	0	.01	0	.02	.30	.38	.38	.53	.81	1	.5
VERY GOOD (VG)	0	0	0	0	.15	.21	.21	.28	.49	.54	1

TABLE 4 Histogram of survey responses for single-antecedent rules between indicator x = touching level and consequent y = flirtation level. Entries denote the number of respondents out of 47 that chose the consequent, (adapted from J.M. Mendel [11] ©2001, Prentice-Hall). The top half shows the histograms before pre-processing, and the bottom half shows the histograms after pre-processing.

		FLIRTATION				
	TOUCHING	NVL	S	MOA	LA	MAA
BEFORE DATA PREPROCESSING	1. NVL	42	3	2	0	0
	2. S	33	12	0	2	0
	3. MOA	12	16	15	3	1
	4. LA	3	6	11	25	2
	5. MAA	3	6	8	22	8
AFTER DATA PREPROCESSING	1. NVL	42	0	0	0	0
	2. S	33	12	0	0	0
	3. MOA	12	16	15	3	0
	4. LA	0	6	11	25	2
	5. MAA	0	6	8	22	8

$$\mu_{10-\tilde{A}}(x) = \mu_{\tilde{A}}(10-x),\ \forall x, \tag{3}$$

where $10-\tilde{A}$ is the antonym of the IT2 FS $\tilde{A}$. Because an IT2 FS is completely characterized by its LMF and UMF, each of which is a T1 FS, $\mu_{10-\tilde{A}}$ in (3) is obtained by applying (2) to both $LMF(\tilde{A})$ and $UMF(\tilde{A})$.

2) *Decoder for the IJA:* The IJA decoder provides a linguistic ranking (first, second,..., fifth) using an average centroid based ranking method. It also provides similarities between those alternatives. However, the investor may also want to know the uncertainties and risks associated with the ranking. As such, the ***challenge*** here was how to obtain a ranking band and a risk band.

In our ***solution***, the interval centroid was used as a *ranking band* for each alternative. The amount of overlap of the ranking bands is another indicator of how similar the investment alternatives are. The antonym of the ranking band was used to provide a *risk band* (of course, other definitions are possible), i.e., high rank implies low risk, and vice-versa; hence,

$$\begin{aligned}\text{risk band}(a_i) &= 10 - \text{Centroid}(\tilde{Y}_{LWA(a_i)})\\ &= [10 - c_r(\tilde{Y}_{LWA(a_i)}), 10 - c_l(\tilde{Y}_{LWA(a_i)})].\end{aligned}$$

Frequently, an investor is asked to provide a numerical value of the risk that he/she associates with an investment alternative, so that optimal allocations can be determined to minimize risk while achieving a prescribed level of profit (return). Such numerical values of risk are usually quite uncertain and may therefore be unreliable. One of the very interesting by-products of the IJA is a numerical risk band; hence, by using the IJA it should no longer be necessary to ask an investor for a numerical value of the risk that he/she associates with an investment alternative. Additionally, optimal allocations can now be performed using risk bands instead of risk values, so that the uncertainties about the risk bands flow through the calculations of the optimal allocations.

B. Social Judgment Advisor (SJA)

According to Mendel et al. [16]:

In everyday social interaction, each of us is called upon to make judgments about the meaning of another's behavior. Such judgments are far from trivial, since they often affect the nature and direction of the subsequent social interaction and communications. But, how do we make this judgment? Although a variety of factors may enter into our decision, behavior is apt to play a critical role in assessing the level of the variable of interest.

Some examples of behavior are kindness, generosity, flirtation, jealousy, harassment, vindictiveness, morality, etc. In this subsection we focus on flirtation, and the result is called a *social judgment advisor* (SJA).

1) *Encoder:* Assuming that the only[1] indicator of importance of flirtation is touching. The following *user friendly 10-word vocabulary* could be established for both touching and flirtation: *none to very little, very little, little, small amount, some, a moderate amount, a considerable amount, a large amount, very large* and *a maximum amount.* Surveyed subjects could be asked a question such as: "On a scale of zero to ten where would you locate the endpoints of an interval for this word?" These data are then mapped by means of the Encoder and the IA into an IT2 FS model for each word (Box 3).

2) *Rulebase Construction*: For the SJA the CWW engine uses IF-THEN rules. A small set of, e.g., five, rules could be established, using a subset of five of the 10 words, e.g., *none to very little* (NVL), *some* (S), *moderate amount* (MOA), *large amount* (LA), and *maximum amount* (MAA). One such rule might be: *IF touching is* a moderate amount, *THEN the level of flirtation* is some.

Another survey could be conducted in which subjects choose one of these five flirtation terms for each rule (i.e., for the rule's consequent). Because all respondents do not agree on the choice of the consequent, this introduces uncertainties into this IF-THEN rule-based CWW engine. The top half of Table 4 provides the data collected from 47 respondents to such a survey. Observe that there are bad responses defined below and outliers in the survey histograms. So the ***challenge*** was how to remove these bad data and outliers by data pre-processing when the data are words. Our ***solution*** consisted of three steps: 1) bad data processing, 2) outlier processing, and, 3) tolerance limit processing. Rule 2 in the top half of Table 4 is used below as an example to illustrate the details of these three steps.

- *Bad Data Processing:* This removes gaps (a zero between two non-zero values) in a group of subject's responses. In Table 4, for the question *"IF there is* some *touching,*

[1]Multi-antecedent SJAs are discussed in Section III-B5 and also Chapter 8 of [19].

THEN there is _____ flirtation," three different consequents were obtained: *none to very little, some*, and *large*. A gap exists between *some* and *large amount*. Let $G_1 = \{none\ to\ very\ little, some\}$ and $G_2 = \{large\ amount\}$. Because G_1 has considerably more responses than G_2, it is passed to the next step of data pre-processing and G_2 is discarded.

- *Outlier processing:* Outlier processing uses a Box and Whisker test [24]. Outliers are points that are unusually too large or too small. A Box and Whisker test is usually stated in terms of first and third quartiles and an interquartile range. The first and third quartiles, $Q(0.25)$ and $Q(0.75)$, contain 25% and 75% of the data, respectively. The inter-quartile range, *IQR*, is the difference between the third and first quartiles; hence, *IQR* contains 50% of the data between the first and third quartiles. Any datum that is more than 1.5 *IQR* above the third quartile or more than 1.5 *IQR* below the first quartile is considered an outlier [24]; however, rule consequents are words modeled by IT2 FSs, thus the Box and Whisker test cannot be directly applied to them. So, the ***challenge*** is how to perform the Box and Whisker test on IT2 FSs. In our ***solution***, the Box and Whisker test is applied to the set of centers of centroids formed by the centers of centroids of the rule consequents. Focusing again on Rule 2, the centers of centroids of the consequent IT2 FSs *NVL*, *S*, *MOA*, *LA* and *MAA* are first computed, and are 0.48, 4.50, 4.95, 8.13 and 9.68, respectively. Then the set of centers of centroids is

$$\{\underbrace{0.48,\ldots,0.48}_{33},\underbrace{4.50,\ldots,4.50}_{12}\},\qquad (4)$$

  where each center of centroid is repeated a certain number of times according to the number of respondents after bad data processing. The Box and Whisker test is then applied to this crisp set, where $Q(0.25) = 0.48$, $Q(0.75) = 4.50$, and $1.5\,IQR = 6.03$. For Rule 2, no data are removed in this step. On the other hand, for Rule 1, the three responses to *some* and the two responses to *moderate amount* are removed.
- *Tolerance limit processing:* Let m and σ be the mean and standard deviation of the remaining histogram data after outlier processing. If a datum lies in the tolerance interval $[m - k\sigma, m + k\sigma]$, then it is accepted; otherwise, it is rejected [24]. k is determined such that one is 95% confident that the given limits contain at least 95% of the available data. For Rule 2, tolerance limit processing is performed on the *set of centers of centroids* in (4), for which $m = 1.55$, $\sigma = 1.80$ and $k = 2.41$. No word is removed for this particular example; so, two consequents, *none to very little* and *some*, are accepted for this rule.

The final pre-processed responses for the histograms in the top half of Table 4 are given in its bottom half. Observe that most responses have been preserved; however, most rule consequents are still histograms instead of a single word. The next ***challenge*** was how to use a histogram of consequent words in rulebase construction. Our ***solution*** was to preserve the distributions of the responses for each rule by using an NWA to obtain the rule consequents, as illustrated by the following:

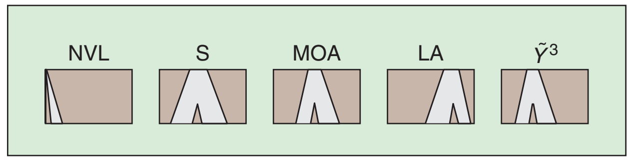

FIGURE 4 $\tilde{Y}^3$ obtained by aggregating the consequents of $R_1^3 - R_4^3$.

Example: Observe from the bottom half of Table 4 that when the antecedent is MOA there are four valid consequents, so that the following four rules will be fired:
R_1^3: *IF touching is MOA, THEN flirtation is NVL.*
R_2^3: *IF touching is MOA, THEN flirtation is S.*
R_3^3: *IF touching is MOA, THEN flirtation is MOA.*
R_4^3: *IF touching is MOA, THEN flirtation is LA.*
These four rules should not be considered of equal importance because they have been selected by different numbers of respondents. An intuitive way to handle this is to assign weights to the four rules, where the weights are proportional to the number of responses, e.g., the weight for R_1^3 is $12/46$, and the weight for R_2^3 is $16/46$. The aggregated consequent $\tilde{Y}^3$ is

$$\tilde{Y}^3 = \frac{12\mathrm{NVL} + 16\mathrm{S} + 15\mathrm{MOA} + 3\mathrm{LA}}{12 + 16 + 15 + 3}.$$

$\tilde{Y}^3$ is computed by the NWA. The result is shown in Fig. 4. Observe that the shape of $\tilde{Y}^3$ looks like the shape of MOA; however, it is shifted somewhat leftwards along the flirtation-level axis, so $\tilde{Y}^3$ is not the same as MOA.

3) *CWW Engine and Decoder:* Once the rulebase is constructed, the next step is to compute the output for a new input word. We use Perceptual Reasoning (see Section II-B).

   Consider single-antecedent rules of the form

$$R^i: \text{If } x \text{ is } \tilde{F}^i,\ \text{Then } y \text{ is } \tilde{Y}^i \quad i = 1,\ldots,N,$$

where $\tilde{F}^i$ and $\tilde{Y}^i$ are words modeled by IT2 FSs. In PR, the Jaccard similarity measure is used to compute the firing levels of the rules, f^i, $i = 1,\ldots,N$. Then, the output FOU of the SJA is computed as

$$\tilde{Y}_C = \frac{\sum_{i=1}^{N} f^i \tilde{Y}^i}{\sum_{i=1}^{N} f^i}.$$

The subscript C in $\tilde{Y}_C$ stands for *consensus* because $\tilde{Y}_C$ is obtained by aggregating the survey results from a population of people, and the resulting SJA is called a *Consensus Flirtation Advisor*. $\tilde{Y}_C$ is then mapped to the most similar word in the 10-word codebook using the Jaccard similarity measure.

4) *How to Use the Flirtation Advisor:* A flirtation adviser could be used to train a person to better understand the relationship between touching and flirtation, so that they reach correct conclusions about such a social situation. Their perception of flirtation for each of the 10 words for touching

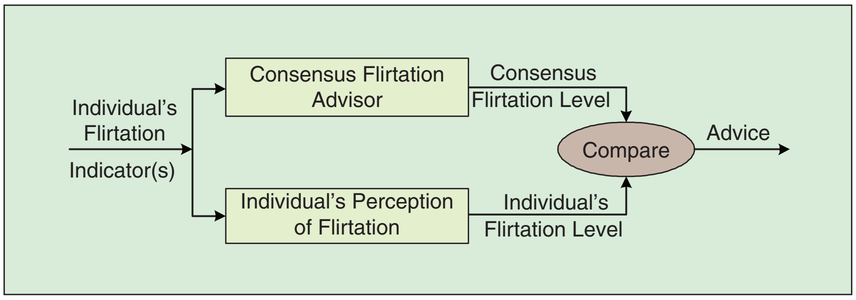

FIGURE 5 One way to use the SJA for a social judgment [19].

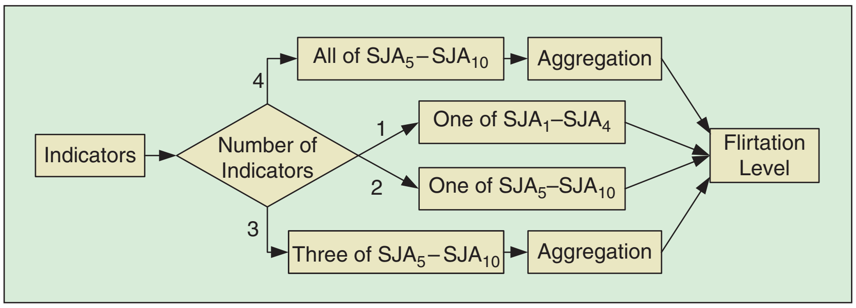

FIGURE 6 An SJA architecture for one-to-four indicators [19].

leads to their individual flirtation level (Fig. 5) for each level of touching, and their individual flirtation level is then compared with the corresponding consensus flirtation level. If there is good agreement between the consensus and individual's flirtation level, then the individual is given positive feedback about this; otherwise, he or she is given advice on how to re-interpret the level of flirtation for the specific level of touching. It is not necessary that there be exact agreement between the consensus and individual's flirtation levels for the individual to be given positive feedback, because the consensus and individual's flirtation levels may be similar enough. The Jaccard similarity measure can be used to quantify what is meant by "similar enough."

5) *On Multiple Indicators:* Generally, people have difficulties in answering questions with more than two antecedents. So, in the survey each rule consists of only one or two antecedents; however, in practice an individual may observe one indicator or more than one indicators. The ***challenge*** was how to deduce the output for multiple antecedents using rulebases consisting of only one or two antecedent rules.

For the sake of this discussion, assume there are four indicators of flirtation, *touching, eye contact, acting witty* and *primping.* Ten SJAs can be created, where SJA_1-SJA_4 are single-antecedent SJAs, and SJA_5-SJA_{10} are two-antecedent SJAs (touching & eye contact, touching & acting witty, touching & primping, eye contact & acting witty, eye contact & primping, acting witty & primping). An example rule for SJA_{10} is: IF *acting witty is* _____ and *primping is* _____, THEN *flirtation* is _____.

Our ***solution*** was:

- ❏ When only one indicator is observed, only one single-antecedent SJA from SJA_1-SJA_4 is activated.
- ❏ When only two indicators are observed, only one two-antecedent SJA from SJA_5-SJA_{10} is activated.
- ❏ When more than two indicators are observed, the output is computed by aggregating the outputs of the activated two-antecedent SJAs[2]. The final output is some kind of aggregation of the results from these SJAs. There are different aggregation operators, e.g., mean, linguistic weighted average, maximum, etc. An intuitive approach is to survey the subjects about the relative importance of the four indicators and hence to determine the linguistic relative importance of SJA_5–SJA_{10}. These relative importance words can then be used as the weights for SJA_5–SJA_{10}, and the final flirtation level can then be computed by a linguistic weighted average.

A diagram of the proposed SJA architecture for different numbers of indicators is shown in Fig. 6.

Finally, note that a missing observation is not the same as an observation of zero value; hence, even if it was possible to create four antecedent rules, none of those rules could be activated if one or more of the indicators had a missing observation. It is therefore very important to have sub-advisors that will be activated when only one or two of these indicators are occurring.

C. Procurement Judgment Advisor (PJA)

This subsection is directed at the following hierarchical multi-criteria missile evaluation problem [21]:

A contractor has to decide which of three companies is going to get the final mass production contract for a missile. The contractor uses five criteria to base the decision, namely: tactics, technology, maintenance, economy *and* advancement. *Each of these criteria has some associated technical sub-criteria (see Table 5). The contractor creates a performance evaluation table, Table 5, in order to assist in choosing the winning system. The sub-criteria evaluations range from numbers to words, and the weights for the sub-criteria and criteria are T1 fuzzy numbers, e.g., around seven, around five, etc. Somehow the contractor has to aggregate this disparate information to determine the winning company.*

The missile evaluation problem is summarized in Fig. 7, a figure that is adopted from [21] where it first appeared. It is very clear from this figure that this is a multi-criteria and two-level decision making problem. At the first level each of the three systems (A, B and C) is evaluated for its performance on five criteria: tactics, technology, maintenance, economy and

[2]Some of the four single-antecedent SJAs, SJA_1–SJA_4, are also fired; however, they are not used because they do not fit the inputs as well as two-antecedent SJAs, since the latter account for the correlation between two antecedents, whereas the former do not.

advancement. The second level in this hierarchical decision making problem involves a weighted aggregation of the five criteria for each of the three systems.

Next we introduce our Per-C approach for the PJA.

1) *Encoder*: In this application, mixed data are used—crisp numbers, T1 fuzzy numbers and words. The codebook contains the crisp numbers, the T1 fuzzy numbers with their associated T1 FS models, and the words and their IT2 FS models.

Our first ***challenge*** was how to ensure NWAs are not unduly-influenced by large numbers. The ***solution*** was to map all of the Table 5 numbers into [0, 10]. Let x_1, x_2 and x_3 denote the raw numbers for Systems A, B and C, respectively. For the 13 sub-criteria whose inputs are numbers, those raw numbers were transformed into:

$$x_i \rightarrow x_i' = \frac{10x_i}{\max(x_1, x_2, x_3)}. \quad (5)$$

Examining Table 5, observe that the words used for the remaining 10 sub-criteria are: {*low, high*} and {*poor, average, good, very good*}. The IA can be used to map their survey data into IT2 FSs.

As in the IJA, where it was first observed that some sub-criteria may have a positive connotation and others may have a negative connotation, a similar situation occurs here. Observe from Table 5 that the following six sub-criteria have a negative connotation:

- *Flight height:* The lower the flight height the better, because it is then more difficult for a missile to be detected by radar.

TABLE 5 Performance evaluation table. Criteria with their weights, sub-criteria with their weights and sub-criteria performance valuation data for the three systems [19].

ITEM	WEIGHT ($\tilde{W}_i$)	SYSTEM A ($\tilde{X}_{Ai}$)	SYSTEM B B ($\tilde{X}_{Bi}$)	SYSTEM C ($\tilde{X}_{Ci}$)
CRITERION 1: TACTICS	$\tilde{9}$			
1. EFFECTIVE RANGE (km)	$\tilde{7}$	43	36	38
2. FLIGHT HEIGHT (m)	$\tilde{1}$	25	20	23
3. FLIGHT VELOCITY (M. No)	$\tilde{9}$	0.72	0.80	0.75
4. RELIABILITY (%)	$\tilde{9}$	80	83	76
5. FIRING ACCURACY (%)	$\tilde{9}$	67	70	63
6. DESTRUCTION RATE (%)	$\tilde{7}$	84	88	86
7. KILL RADIUS (m)	$\tilde{6}$	15	12	18
CRITERION 2: TECHNOLOGY	$\tilde{3}$			
8. MISSILE SCALE (cm) (l × d-span)	$\tilde{4}$	521×35-135	381×34-105	445×35-120
9. REACTION TIME (min)	$\tilde{9}$	1.2	1.5	1.3
10. FIRE RATE (round/min)	$\tilde{9}$	0.6	0.6	0.7
11. ANTI-JAM (%)	$\tilde{8}$	68	75	70
12. COMBAT CAPABILITY	$\tilde{9}$	VERY GOOD	GOOD	GOOD
CRITERION 3: MAINTENANCE	$\tilde{1}$			
13. OPERATION CONDITION REQUIREMENT	$\tilde{5}$	HIGH	LOW	LOW
14. SAFETY	$\tilde{6}$	VERY GOOD	GOOD	GOOD
15. DEFILADE	$\tilde{2}$	GOOD	VERY GOOD	GOOD
16. SIMPLICITY	$\tilde{3}$	GOOD	GOOD	GOOD
17. ASSEMBLY	$\tilde{3}$	GOOD	GOOD	POOR
CRITERION 4: ECONOMY	$\tilde{5}$			
18. SYSTEM COST (10,000)	$\tilde{8}$	800	755	785
19. SYSTEM LIFE (YEARS)	$\tilde{8}$	7	7	5
20. MATERIAL LIMITATION	$\tilde{5}$	HIGH	LOW	LOW
CRITERION 5: ADVANCEMENT	$\tilde{7}$			
21. MODULARIZATION	$\tilde{5}$	AVERAGE	GOOD	AVERAGE
22. MOBILITY	$\tilde{7}$	POOR	VERY GOOD	GOOD
23. STANDARDIZATION	$\tilde{3}$	GOOD	GOOD	VERY GOOD

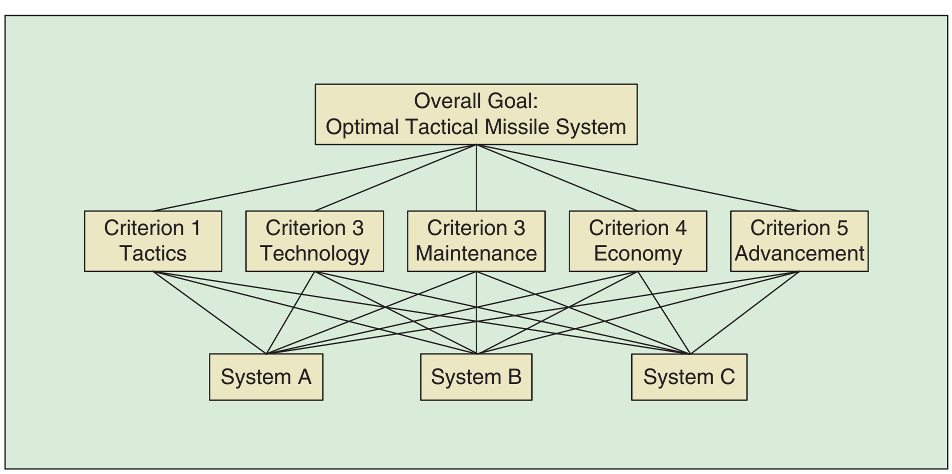

FIGURE 7 Structure of evaluating competing tactical missile systems from three companies [21].

TABLE 6 Challenges and their occurrences in the applications.

CHALLENGES	APPLICATIONS IJA	SJA	PJA
HOW TO TRANSFORM WORDS INTO IT2 FSs IN THE ENCODER? (SECTION II-A)	✓	✓	✓
HOW TO AGGREGATE DISPARATE INFORMATION (NUMBERS, INTERVALS, T1 FSs, WORDS) IN A WEIGHTED AVERAGE? (SECTION II-B)	✓		✓
HOW TO USE IF-THEN RULES IN A CWW ENGINE SO THAT THE OUTPUT FOU RESEMBLES THE CODEBOOK FOUs? (SECTION II-B)		✓	
HOW TO MAP THE OUTPUT OF THE CWW ENGINE INTO A RECOMMENDATION? (SECTION II-C)	✓	✓	✓
HOW TO TRIM A TOO LARGE CODEBOOK SO THAT IT IS MORE USER-FRIENDLY? (SECTION III-A1)	✓		
HOW TO HANDLE SUB-CRITERIA WHICH HAVE NEGATIVE CONNOTATIONS AND WHOSE INPUTS ARE WORDS? (SECTION III-A2)	✓		✓
HOW TO OBTAIN A RANKING BAND AND A RISK BAND? (SECTION III-A3)	✓		
HOW TO REMOVE BAD DATA AND OUTLIERS WHEN RESPONSES ARE WORDS AND NOT NUMBERS? (SECTION III-B2)		✓	
HOW TO USE A HISTOGRAM OF CONSEQUENT WORDS IN RULEBASE CONSTRUCTION? (SECTION III-B2)		✓	
HOW TO PERFORM THE BOX AND WHISKER TEST ON IT2 FSs? (SECTION III-B2)		✓	
HOW TO DEDUCE THE OUTPUT FOR MULTIPLE ANTECEDENTS USING RULEBASES CONSISTING OF ONLY ONE OR TWO ANTECEDENT RULES? (SECTION III-B5)		✓	
HOW TO ENSURE NWAs ARE NOT UNDULY-INFLUENCED BY LARGE NUMBERS? (SECTION III-C1)			✓
HOW TO HANDLE SUB-CRITERIA WHICH HAVE NEGATIVE CONNOTATIONS AND WHOSE INPUTS ARE NUMBERS? (SECTION III-C1)			✓

- *Missile scale*: A smaller missile is harder to detect by radar.
- *Reaction time*: A missile with shorter reaction time can respond more quickly.
- *System cost*: The cheaper the better.
- *Operation condition requirement*: A missile with lower operation condition requirement can be deployed more easily and widely.
- *Material limitation*: A missile with lower material limitation can be produced more easily, especially during wartime.

The inputs to the last two sub-criteria with negative connotations are words modeled by IT2 FSs, and hence their *antonyms* can be used in the aggregation, similar to the case in the IJA. The ***challenge*** was how to handle the first four of the six sub-criteria with negative connotations, whose inputs are numbers. In our ***solution***, a preprocessing step was used to convert a large x_i' into a small number x_i^* and a small x_i' into a large number x_i^*:

$$x_i \rightarrow x_i^* = 1/x_i \tag{6}$$

and then (5) was applied to x_i^*:

$$x_i^* \rightarrow x_i' = \frac{10x_i^*}{\max(x_1^*, x_2^*, x_3^*)}.$$

2) *CWW Engine*: Observe from Table 5 that the inputs to the sub-criteria consists of numbers, T1 FSs and words modeled by IT2 FSs, and the weights are T1 FSs. The NWAs are used to aggregate such disparate information. Each major criterion has an NWA computed for it. Consider System A as an example. Examining Table 5, observe that the NWA for *Tactics* (Y_{A1}) is an FWA (because the weights are T1 FSs and the sub-criteria evaluations are numbers), whereas the NWAs for *Technology* $(\tilde{Y}_{A2})$, *Maintenance* $(\tilde{Y}_{A3})$, *Economy* $(\tilde{Y}_{A4})$ and *Advancement* $(\tilde{Y}_{A5})$ are LWAs (because at least one sub-criterion evaluation is a word modeled by an IT2 FS), e.g.,

$$Y_{A1} = \frac{\sum_{i=1}^{7} X_{Ai} W_i}{\sum_{i=1}^{7} W_i} \tag{7}$$

$$\tilde{Y}_{A2} = \frac{\sum_{i=8}^{12} \tilde{X}_{Ai} \tilde{W}_i}{\sum_{i=8}^{12} \tilde{W}_i}. \tag{8}$$

Equations similar to (8) can be written for $\tilde{Y}_{A3}$, $\tilde{Y}_{A4}$ and $\tilde{Y}_{A5}$. These six NWAs are then aggregated by another NWA to obtain the overall performance of System A, $\tilde{Y}_A$, as follows:

$$\tilde{Y}_A = \frac{\tilde{9}\tilde{Y}_{A1} + \tilde{3}\tilde{Y}_{A2} + \tilde{1}\tilde{Y}_{A3} + \tilde{5}\tilde{Y}_{A4} + \tilde{7}\tilde{Y}_{A5}}{\tilde{9} + \tilde{3} + \tilde{1} + \tilde{5} + \tilde{7}}.$$

As a reminder to the reader, when $i = \{2, 8, 9, 18\}$, (6) must be used, and when $i = \{13, 20\}$, the antonyms of the corresponding word-IT2 FSs must be used. For all other values of i the numbers or word-IT2 FSs are used directly.

3) *Decoder*: Similar to the IJA, the centroid based ranking method is applied to the final aggregation results of the three systems to identify the winner. To assess the uncertainties associated with the ranking, ranking bands of the three systems can also be computed.

IV. Conclusions

Perceptual computing is a *methodology* of CWW for assisting people in making subjective judgments. The Perceptual Computer–Per-C–is our instantiation of perceptual computing; it consists of three components–encoder, decoder and CWW engine. Stepping back from the details for designing each of these components, the *methodology of perceptual computing* is:

1) Focus on an application ($\mathcal{A}$).
2) Establish a vocabulary (or vocabularies) for $\mathcal{A}$.
3) Collect interval end-point data from a group of subjects (representative of the subjects who will use the Per-C) for all of the words in the vocabulary.
4) Map the collected word data into word-FOUs by using the Interval Approach (Box 3). The result of doing this is the *codebook* (or codebooks) for $\mathcal{A}$, and completes the design of the encoder of the Per-C.
5) Choose an appropriate CWW engine for $\mathcal{A}$; it maps IT2 FSs into one or more IT2 FS.
6) If an existing CWW engine is available for $\mathcal{A}$, then use its available mathematics to compute its output(s) (Section II-B). Otherwise, develop such mathematics for your new kind of CWW engine. Your new CWW engine should be constrained so that its output(s) resemble the FOUs in the codebook(s) for $\mathcal{A}$.
7) Map the IT2 FS outputs from the CWW engine into a recommendation at the output of the decoder. If the recommendation is a word, rank or class, then use existing mathematics to accomplish this mapping (Section II-C). Otherwise, develop such mathematics for your new kind of decoder.

The constraint in Step 6, that the output FOU of the CWW engine should resemble the FOUs in the codebook(s) for $\mathcal{A}$, is the major difference between perceptual computing and function approximation applications of FSs and systems.

When the methodology of perceptual computing was applied to actual applications, challenges occurred that had to be overcome. In this article we have described three applications, the challenges encountered and how they were overcome. A summary of all the challenges and their occurrences in the applications is shown in Table 6. More applications of Per-C have also been reported in the literature (see [1], [3], [18] and Chapter 10 of [19]). For example, in [1] the Per-C was used to evaluate the marine invasion risk caused by recreational vessels and the LWA was used to aggregate expert opinions before they were used in PR; in [18] and Chapter 10 of [19] the Per-C was used as a journal publication judgment advisor and a subsethood measure was used to map the final aggregated FOU (representing the overall quality of a paper) into three decision categories (accept, rewrite, or reject); and, in [3] the Per-C was applied to a location choice problem in which the LWA was used to obtain a consensus weight for each sub-criterion when each judge provided his/her own weight.

Matlab functions for implementing the Per-C can be downloaded from the authors' websites at http://sipi.usc.edu/~mendel/ and http://www-scf.usc.edu/~dongruiw/files/Matlab_PerceptualComputing.rar.

V. Acknowledgment

The authors would like to thank the Wiley-IEEE Press for giving them the permission to reproduce materials from their book *Perceptual Computing: Aiding People in Making Subjective Judgments*.

References

[1] H. Acostaa, D. Wu, and B. M. Forrestc, "Fuzzy experts on recreational vessels, a risk modelling approach for marine invasions," *Ecol. Model.*, vol. 221, no. 5, pp. 850–863, 2010.
[2] S. Coupland and R. John, "Type-2 fuzzy logic: A historical view," *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 57–62, 2007.
[3] S. Han and J. M. Mendel, "A new method for managing the uncertainties in evaluating multi-person multi-criteria location choices, using a Percetual computer," *Ann. Oper. Res.*, vol. 195, no. 1, pp. 277–309, 2012.
[4] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inform. Sci.*, vol. 132, pp. 195–220, 2001.
[5] C. S. Kim, D. S. Kim, and J. S. Park, "A new fuzzy resolution principle based on the antonym," *Fuzzy Sets Syst.*, vol. 113, pp. 299–307, 2000.
[6] F. Liu and J. M. Mendel, "Aggregation using the fuzzy weighted average, as computed using the Karnik-Mendel algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 1–12, 2008.
[7] F. Liu and J. M. Mendel, "Encoding words into interval type-2 fuzzy sets using an interval approach," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1503–1521, 2008.
[8] J. M. Mendel, J. Lawry, and L. A. Zadeh, "Foreword to the special section on computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 437–440, 2010.
[9] J. M. Mendel, L. A. Zadeh, E. Trillas, R. Yager, J. Lawry, H. Hagras, and S. Guadarrama, "What computing with words means to me," *IEEE Comput. Intell. Mag.*, vol. 5, pp. 20–26, 2010.
[10] J. M. Mendel, "The perceptual computer: An architecture for computing with words," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Melbourne, Australia, Dec. 2001, pp. 35–38.
[11] J. M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Upper Saddle River, NJ: Prentice-Hall, 2001.
[12] J. M. Mendel, "An architecture for making judgments using computing with words," *Int. J. Appl. Math. Comp. Sci.*, vol. 12, no. 3, pp. 325–335, 2002.
[13] J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Inform. Sci.*, vol. 177, pp. 988–1006, 2007.
[14] J. M. Mendel, "Computing with words: Zadeh, Turing, Popper and Occam," *IEEE Comput. Intell. Mag.*, vol. 2, pp. 10–17, 2007.
[15] J. M. Mendel, "Type-2 fuzzy sets, a tribal parody," *IEEE Comput. Intell. Mag.*, vol. 5, no. 4, pp. 24–27, 2010.
[16] J. M. Mendel, S. Murphy, L. C. Miller, M. Martin, and N. N. Karnik, "The fuzzy logic advisor for social judgments," in *Computing with Words in Information/Intelligent Systems*, L. A. Zadeh and J. Kacprzyk, Eds. Physica-Verlag, 1999, pp. 459–483.
[17] J. M. Mendel and D. Wu, "Perceptual reasoning for perceptual computing," *IEEE Trans. Fuzzy Systems*, vol. 16, no. 6, pp. 1550–1564, 2008.
[18] J. M. Mendel and D. Wu, "Computing with words for hierarchical and distributed decision making," in *Computational Intelligence in Complex Decision Systems*, D. Ruan, Ed. Paris, France: Atlantis, 2010.
[19] J. M. Mendel and D. Wu, *Perceptual Computing: Aiding People in Making Subjective Judgments*. Hoboken, NJ: Wiley-IEEE Press, 2010.
[20] G. Miller, "The magical number seven plus or minus two: Some limits on the capacity for processing information," *Psychol. Rev.*, vol. 63, no. 2, pp. 81–97, 1956.
[21] D.-L. Mon, C.-H. Cheng, and J.-L. Lin, "Evaluating weapon system using fuzzy analytic hierarchy process based on entropy weight," *Fuzzy Sets Syst.*, vol. 62, pp. 127–134, 1994.
[22] T. Saaty and M. Ozdemir, "Why the magic number seven plus or minus two," *Math. Comput. Model.*, vol. 38, no. 3, pp. 233–244, 2003.
[23] R. M. Tong and P. P. Bonissone, "A linguistic approach to decision making with fuzzy sets," *IEEE Trans. Syst., Man, Cybern.*, vol. 10, pp. 716–723, 1980.
[24] R. W. Walpole, R. H. Myers, A. Myers, and K. Ye, *Probability and Statistics for Engineers and Scientists*, 8th ed. Upper Saddle River, NJ: Prentice-Hall, 2007.
[25] D. Wu and J. M. Mendel, "Aggregation using the linguistic weighted average and interval type-2 fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 6, pp. 1145–1161, 2007.
[26] D. Wu and J. M. Mendel, "Corrections to 'Aggregation using the linguistic weighted average and interval type-2 fuzzy sets'," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1664–1666, 2008.
[27] D. Wu and J. M. Mendel, "A comparative study of ranking methods, similarity measures and uncertainty measures for interval type-2 fuzzy sets," *Inform. Sci.*, vol. 179, no. 8, pp. 1169–1192, 2009.
[28] D. Wu and J. M. Mendel, "Enhanced Karnik-Mendel algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 923–934, 2009.
[29] D. Wu and J. M. Mendel, "Perceptual reasoning for perceptual computing: A similarity-based approach," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1397–1411, 2009.
[30] D. Wu, J. M. Mendel, and S. Coupland, "Enhanced interval approach for encoding words into interval type-2 fuzzy sets and its convergence analysis," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 3, pp. 499–513, 2012.
[31] L. A. Zadeh, "Fuzzy logic = Computing with words," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 103–111, 1996.
[32] L. A. Zadeh, "From computing with numbers to computing with words—From manipulation of measurements to manipulation of perceptions," *IEEE Trans. Circuits Syst. I*, vol. 46, no. 1, pp. 105–119, 1999.
[33] L. A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *AI Mag.*, pp. 73–84, 2001.