CHAPTER 1

# Introduction

## 1.1 PERCEPTUAL COMPUTING

Lotfi Zadeh (1996, 1999, 2008), the father of fuzzy logic, coined the phrase "computing with words." Different acronyms have been used for computing with words, such as CW and CWW. In this book, the latter is chosen because its three letters coincide with the three words in "computing with words." According to Zadeh (1999):

> 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.

In a December 26, 2008, e-mail, Zadeh further stated:

> In 2008, computing with words (CW or CWW) has grown in visibility and recognition. There are two basic rationales for the use of computing with words. First, when we have to use words because we do not know the numbers. And second, when we know the numbers but the use of words is simpler and cheaper, or when we use words to summarize numbers. In large measure, the importance of computing with words derives from the fact that much of human knowledge is described in natural language. In one way or another, the fuzzy-logic-based machinery of computing with words opens the door to a wide-ranging enlargement of the role of natural languages in scientific theories, including scientific theories which relate to economics, medicine, law and decision analysis.

Of course, Zadeh 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 (Fig. 1.1).

*Perceptual Computing.* By Jerry M. Mendel and Dongrui Wu

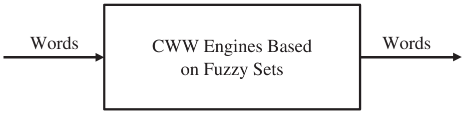


**Figure 1.1.** The CWW paradigm.

Zadeh's definition of CWW is very general and does not refer to a specific field in which CWW would be used. In this book, our focus is on CWW for making subjective judgments, which we call *perceptual computing*.[1]

A subjective judgment is a personal opinion that has been influenced by one's personal views, experience, or background. It can also be interpreted as a personal assessment of the level of a variable of interest and is made using a mixture of qualitative and quantitative information. Examples of subjective judgments are given in Section 1.2.

Zadeh (2001) also states he is interested in developing a computational theory of perceptions—the development of machinery for computing and reasoning with perceptions. Our thesis is that humans make subjective judgments by not only using perceptions but by also using data. Psychologists [e.g., Wallsten and Budescu (1995)] have evidence that although humans prefer to communicate using words, they also want to receive data to support the words. For example, if you are receiving a performance evaluation from your boss, and she tells you that your performance is below average, you will certainly want to know "Why," at which point she will provide quantitative data to you that supports her evaluation. Hence, perceptual computing, as used in this book, is associated with machinery for computing and reasoning with perceptions and data.

Our architecture for perceptual computing is depicted in Fig. 1.2. It is called a *perceptual computer* or Per-C for short [Mendel (2001, 2002, 2007)]. The 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.

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 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 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.

This book explains how to design the encoder, CWW engines, and decoders. It provides the reader with methodologies for doing all of this, so that, perhaps for the

[1]According to *Merriam Webster's On-Line Dictionary,* the word *perceptual* means "of relating to, or involving perception especially in relation to immediate sensory experience"; *perception* means "a result of perceiving"; and *perceive* means "to attain awareness or understanding of," or "to become aware of through the senses." Hopefully, this explains our choice of the word perceptual in perceptual computing.

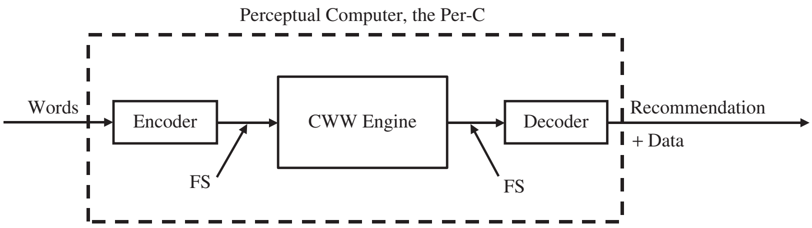


**Figure 1.2.** Specific architecture for CWW—the perceptual computer.

first time, CWW can be fully implemented, at least for making subjective judgments.

## 1.2 EXAMPLES

In this section, four examples are provided that illustrate CWW for making subjective judgments: investment decision making, social judgment making, hierarchical decision making, and hierarchical and distributed decision making. These examples are taken up later in this book, in much greater detail, in Chapters 7–10.

### 1.2.1 Investment Decision Making

Tong and Bonissone (1980) illustrated their approach to linguistic decision making using an investment decision example:

> 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,[2] $a_4$—real estate,[3] 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[4] [profit] received, and $c_4$—the cash realizeability of the capital sum [liquidity].

The individual's goal is to decide which investments he should partake in. In order to arrive at his decisions, the individual must first rate each of the five alternative

[2]Tong and Bonissone called this "gold and/or diamonds." In this book, this is simplified to "gold."

[3]The term *real estate* is somewhat ambiguous because it could mean individual properties, ranging from residential to commercial, or investment vehicles that focus exclusively on real estate, such as a real estate investment trust (REIT) or a real estate mutual fund. In this chapter, real estate is interpreted to mean the latter two.

[4]By *interest* is meant the profit percent from the capital invested; so, in this chapter the term *profit* is used.

investment areas for each of the four criteria. To do this requires that he either knows about the investments or becomes knowledgeable about them. His ratings use words and, therefore, are linguistic ratings. In order to illustrate what the linguistic ratings might look like, the ones used by Tong and Bonissone are provided in the investment alternatives/investment criteria array in Table 1.1. For example, the individual's linguistic ratings about commodities are that there is a high risk of losing his capital sum from investing in commodities, commodities have a more or less high vulnerability to inflation, the amount of profit received from commodities is very high, and commodities are fairly liquid.

What makes the individual's investment choices challenging is that his knowledge about the investments is uncertain; hence, his linguistic ratings are uncertain. Additionally, each individual does not necessarily consider each criterion to be equally important. So, he must also assign a linguistic weight to each of them. The weights chosen by Tong and Bonissone are given in Table 1.2. This individual views the risk of losing his capital as moderately important, the vulnerability to inflation as more or less important, the amount of profit received as very important, and liquidity as more or less unimportant. Although common weights are used for all five investment alternatives, they could be chosen separately for each of the alternatives.

The problem facing the individual investor is how to aggregate the linguistic information in Tables 1.1 and 1.2 so as to arrive at his preferential ranking of the five investments (Fig. 1.3). Clearly, the results will be very subjective because these tables are filled with words and not numbers. The investor may also want to play "what-if" games, meaning that he may want to see what the effects are of changing the words in one or both of the tables on the preferential rankings.

**Table 1.1.** Investment alternatives/investment criteria array. Example of the linguistic ratings of investment alternatives for investment criteria, provided by an individual[a]

| | Investment criteria | | | |
|---|---|---|---|---|
| Investment alternatives | $c_1$ (Risk of losing capital) | $c_2$ (Vulnerability to inflation) | $c_3$ (Amount of profit received) | $c_4$ (Liquidity) |
| $a_1$ (commodities) | High | More or less high | Very high | Fair |
| $a_2$ (stocks) | Fair | Fair | Fair | More or less good |
| $a_3$ (gold) | Low | From fair to more or less low | Fair | Good |
| $a_4$ (real estate) | Low | Very low | More or less high | Bad |
| $a_5$ (long-term bonds) | Very low | High | More or less low | Very good |

[a]An individual fills in this table 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___________?

**Table 1.2.** Example of the linguistic weights for the investment criteria provided by an individual[a]

| $c_1$ (Risk of losing capital) | $c_2$ (Vulnerability to inflation) | $c_3$ (Amount of profit received) | $c_4$ (Liquidity) |
|---|---|---|---|
| Moderately important | More or less important | Very important | More or less unimportant |

[a]An individual fills in this table by answering the following question: The importance that I attach to the investment criterion $c_i$ is _________?

The Per-C that is associated with this application is called an *investment judgment advisor,* and its design is studied in detail in Chapter 7. One of the interesting features of this application is that any person, such as the reader of this book, can fill in Tables 1.1 and 1.2, and immediately find out his/her preferential rankings of the five investments.

### 1.2.2 Social Judgment Making

According to Mendel et al. (1999):

> 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? By *judgment* we mean the assessment of the *level* of the variable of interest. Although a variety of factors may enter into our decision, behavior is apt to play a critical role is assessing the level of the variable of interest.

Some examples of behavior are kindness, generosity, flirtation, jealousy, harassment, vindictiveness, and morality.

Suppose the behavior of interest is flirtation, and the only indicator of importance is eye contact. The following user-friendly vocabulary could be established for both eye contact and flirtation: none to very little, very little, little, small amount, some, a moderate amount, a considerable amount, a large amount, a very

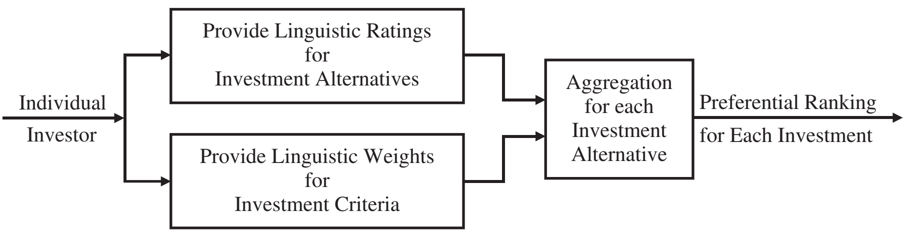


**Figure 1.3.** Investment judgment advisor.

large amount, 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 end points of an interval for this word?” These data could then be mapped by means of the encoder into a FS model for each word. The 10 words and their FS models constitute the codebook for the subjective judgment of flirtation and for eye contact.

A small set of five rules could then be established, using a subset of five of the 10 words: none to very little, some, a moderate amount, a large amount, and a maximum amount. One such rule might be:

IF eye contact 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 resulting rules from the group of subjects are then used as a *consensus flirtation advisor* (Fig. 1.4).

An individual user could interact with this flirtation adviser by inputting any one of the 10 words from the codebook for a specific level of eye contact. Rules within the consensus flirtation advisor would be fired using the mathematics of FSs (as described in Chapter 6), the result being a fired-rule FS for each fired rule. These FSs could then be aggregated into a composite FS that would be compared to the word FSs in the codebook. This comparison would be done using fuzzy set similarity computations, as described in Chapter 4, the result being the word that best describes the consensus flirtation level to the individual.

Such a flirtation adviser could be used to train a person to better understand the relationship between eye contact and flirtation, so that they reach correct conclusions about such a social situation. Their perception of flirtation for each of the 10 words for eye contact leads to their individual flirtation level (Fig. 1.4) for each level of eye contact, 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 levels, then the individual is given positive feedback about this; otherwise, he or she is given advice on how to reinterpret the level of flirtation for the specific level of eye contact. 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

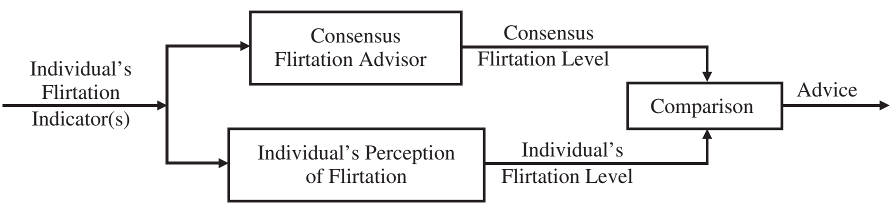


**Figure 1.4.** Flirtation advisor.

enough. Chapter 4 provides quantitative levels of similarity for words in a codebook, so that it will be possible to quantify what is meant by "similar enough."

Of course, in this very simple example of only one flirtation indicator not much confusion can occur; however, when more indicators are used (e.g., eye contact and touching, or primping and acting witty), then in an actual social situation it is possible to get "mixed signals," that is, a certain level of touching may indicate a large amount of flirtation, whereas a certain level of eye contact may indicate none to very little flirtation. So which is it? In this case, more than one rule will fire and the totality of fired rule FSs is an indicator of what is meant by "mixed signals." By aggregating the fired rule FSs and comparing the resulting FS to the word FSs in the codebook the result will again be the word that best describes the flirtation state to the individual user.

In this way, the flirtation adviser can be used to train a person to reach correct conclusions about social situations when he or she is receiving mixed signals. And, as is well known, the same levels of flirtation indicators can mean different levels of flirtation to women and men; so, a female flirtation advisor could be used to sensitize men to those differences, and vice-versa.

It is easy to extend this social judgment application, which some may feel is light-hearted, to many other social judgments and also to nonsocial judgments. Examples of the latter include global warming, environmental impact, water quality, audio quality, toxicity, and terrorism (terrorist).

The details of a social judgment advisor are described in Chapter 8.

### 1.2.3 Hierarchical Decision Making

By "hierarchical decision making" (Fig. 1.5) is meant decision making made by a single individual, group, or organization that is based on comparing the performance of competing alternatives, such as an individual's performance in an athletic, dancing, or cooking competition; a group or individual's proposal for solving a problem or building a product; or product selection (e.g., which flat-panel display should I purchase?) Each alternative is first evaluated or scored (this process may itself involve a hierarchical process involving criteria and subcriteria), after which the evaluations or scores are compared at a higher level to arrive at either a single winning competitor or a subset of winners. What can make this challenging is that the evaluations or scores of the subcriteria and criteria can use numbers, uniformly weighted intervals of numbers, nonuniformly weighted intervals of numbers, or even words. How to aggregate such disparate information (the subject of Chapter 5) is very challenging and lends itself very nicely to perceptual computing.

Two examples are:

1. Tzeng and Teng (1993) define a fuzzy multiobjective transportation selection problem as "a given finite set of $n$ potential projects $x_1, x_2, \dots, x_n$ is evaluated with respect to $m$ objectives $o_1, o_2, \dots, o_m$, and $q$ resources constraints $c_1, c_2, \dots, c_q$." The subset of projects that give the highest improvement urgency index (IUI) are the winners. Some of the $m$ objectives are expressed

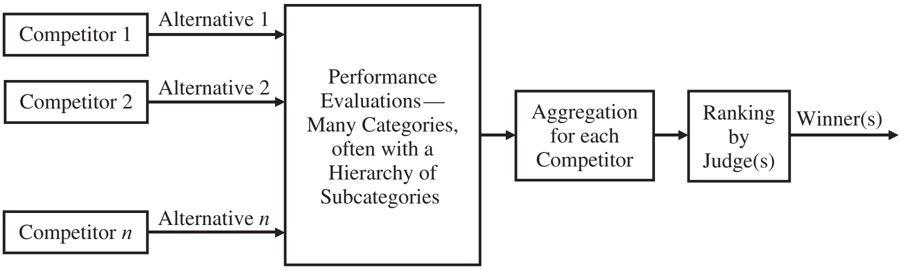


**Figure 1.5.** Hierarchical decision making.

linguistically, for example, environmental impact is {very good, good, fair, poor, very poor}. Each project's IUI is computed by using a fuzzy weighted average (see Chapter 5) and its resulting FS is then converted into a crisp number, $N_{IUI}$. Each project's profitability index (PI) is then computed as the ratio $N_{IUI}$/cost, after which all of the $n$ PIs are ranked. Projects not satisfying the constraints are removed, and the winning projects are selected from the highest PI to the lowest PI, within the limits of an available budget.

2. Mon et al. (1994) consider the following hierarchical multicriteria missile evaluation system. A contractor has to decide which of three companies is going to get the final mass production contract for the missile. The contractor uses five criteria to arrive at his final decision, namely: tactics, technology, maintenance, economy, and advancement. Each of these criteria has some associated technical subcriteria; for example, for tactics the subcriteria are effective range, flight height, flight velocity, reliability, firing accuracy, destruction rate, and kill radius, whereas for economy the subcriteria are system cost, system life, and material limitation.

   The contractor creates a performance evaluation table (Table 1.3) in order to assist in choosing the winning company. Contained within this table are three columns, one for each of the three competing companies. The rows of this table are partitioned into the five criteria, and each of the partitions has additional rows, one for each of its subcriteria. Entries into this table are evaluations of the subcriteria. Additionally, weights are assigned to all of the subcriteria, because they are not of equal importance. These weights are fuzzy numbers such as around seven and around five. The subcriteria evaluations range from numbers to words.

   Somehow, the contractor has to aggregate this disparate information, and this is even more difficult because the five criteria are themselves not of equal importance and have their own fuzzy weights assigned to them.

This application is the subject of Chapter 9, where it is shown how the Per-C can be used to assist the contractor to choose the winning company. Other hierarchical decision making applications are also reviewed in that chapter.

**Table 1.3.** Performance evaluation table. Criteria and subcriteria with their kinds of weights, and kinds of subcriteria data provided for the three companies

| Item | Weighting | Company A | Company B | Company C |
|---|---|---|---|---|
| Criterion 1: Tactics<br>Effective range (km)<br>Flight height (m)<br>Flight velocity (Mach no.)<br>Reliability (%)<br>Firing accuracy (%)<br>Destruction rate (%)<br>Kill radius (m) | Fuzzy numbers | Numerical evaluations | Numerical evaluations | Numerical evaluations |
| Criterion 2: Technology<br>Missile scale (cm)<br>(l × d-span)<br>Reaction time (min)<br>Fire rate (round/min)<br>Antijam (%)<br>Combat capability | Fuzzy numbers | Numerical and linguistic evaluations | Numerical and linguistic evaluations | Numerical and linguistic evaluations |
| Criterion 3: Maintenance<br>Operation condition requirement<br>Safety<br>Defilade<br>Simplicity<br>Assembly | Fuzzy numbers | Linguistic evaluations | Linguistic evaluations | Linguistic evaluations |
| Criterion 4: Economy<br>System cost (10,000)<br>System life (years)<br>Material limitation | Fuzzy numbers | Numerical and linguistic evaluations | Numerical and linguistic evaluations | Numerical and linguistic evaluations |
| Criterion 5: Advancement<br>Modularization<br>Mobility<br>Standardization | Fuzzy numbers | Linguistic evaluations | Linguistic evaluations | Linguistic evaluations |

### 1.2.4 Hierarchical and Distributed Decision Making

By "hierarchical and distributed decision making" (Fig. 1.6) is meant decision making that is ultimately made by a single individual, group or organization, but that is based on aggregating independently made recommendations about an object from other individuals, groups, or organizations (i.e., judges). An object could be a person being considered for a job, an article being reviewed for publication in a journal, a military objective, and so on. It is the independent nature of the recommendations that leads to this being called "distributed," and it is the aggregation of the distributed recommendations at a higher level that leads to this being called "hierarchical."

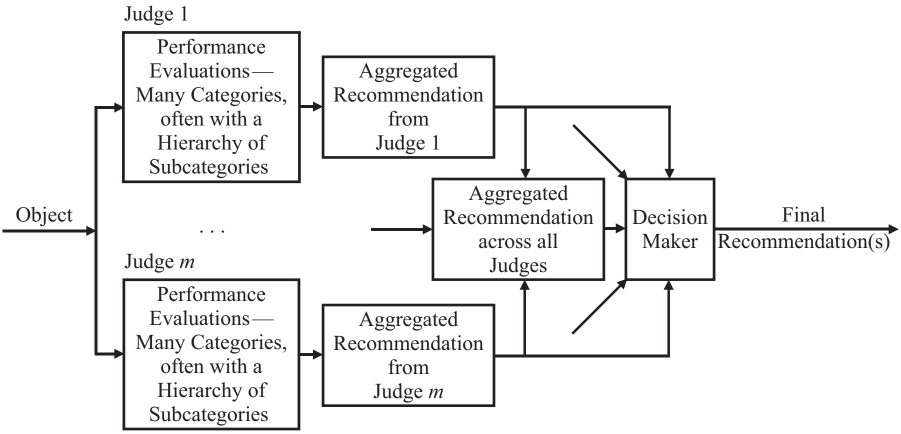


**Figure 1.6.** Hierarchical and distributed decision making.

There can be multiple levels of hierarchy in this process, because each of the independent recommendations may also involve a hierarchical decision making process, as just described in subsection 1.2.3. Additionally, the individuals, groups, or organizations making their independent recommendations may not be of equal expertise, and so a weight has to be assigned to each of them when they are aggregated. The independent recommendations can involve aggregating numbers, uniformly weighted intervals of numbers, nonuniformly weighted intervals of numbers, and even words. The final recommendation (or decision) is made by a decision maker who not only uses an aggregated recommendation that is made across all of the judges but may also use the aggregated recommendation from each of the judges.

Consider the problem of hiring a new employee.[5] For this process, there often is a selection team with a diversity of views. Typically, the selection team is comprised of the position owner, peers (technical experts), customers, and a manager. Prior to posting a job, selection criteria are created and each candidate is evaluated against those criteria. Each selection team member may be weighted differently, and more weight may be applied to the selection criteria in which they have the greatest expertise. For example, peers might care more about a candidate's technical skills and teamwork ability, so more weight could be applied to a peer's evaluation of the candidate's technical and teamwork capabilities. On the other hand, customers might want to know if the individual has the skills to help them with their business problems, and managers might be looking for candidates who can be used in other roles, so for them more weight could be applied to these selection criteria.

Today, a traditional decision sciences hierarchical matrix is used to assist in making the final hiring decision. For this matrix, everyone on the selection team must rate the candidate on a scale from, say, 1 to 10, and usually this is done in a distributed

[5]This example was provided to us by David Tuk (Chevron Corp.).

manner. Individuals are uncomfortable distinguishing between a 7 and an 8, but they might be willing to say outstanding, strong, fair, and poor. So looking at the difference between how the hiring decision is made deterministically versus what would be discovered if FL were used would be very interesting. The paper by Doctor et al. (2008) is the first attempt to develop a version of the Per-C for this problem.

Chapter 10 explains in detail how the Per-C can be applied to the so-called *journal publication judgment advisor,* in which, for the first time, only words are used at every level of the following hierarchical and distributed decision making process. *n* reviewers have to provide a subjective recommendation about a journal article that has been sent to them by the Associate Editor, who then has to aggregate the independent recommendations into a final recommendation that is sent to the Editor-in-Chief of the journal. Because it is very problematic to ask reviewers to provide numerical scores for paper-evaluation subcategories (the two major categories are technical merit and presentation), such as importance, content, depth, style, organization, clarity, references, and so on, each reviewer will only be asked to provide a linguistic score for each of these categories. They will not be asked for an overall recommendation about the paper because in the past it was quite common for reviewers who provided the same numerical scores for such categories to give very different publishing recommendations. By leaving a specific recommendation to the Associate Editor, such inconsistencies can hopefully be eliminated.

Aggregating words to reflect each reviewer's recommendation as well as the expertise of each reviewer about the paper's subject matter is done using a linguistic weighted average (explained in Chapter 5).

Although the journal publication judgment advisor uses reviewers and an associate editor, the word "reviewer" could be replaced by judge, expert, low-level manager, commander, referee, etc, and the term "associate editor" could be replaced by control center, command center, higher-level manager, etc. So, this application has potential wide applicability to many other applications.

## 1.3 HISTORICAL ORIGINS OF PERCEPTUAL COMPUTING

Although Mendel (2001, 2002) was the first to use the term perceptual computer, it is interesting to go back into the literature of fuzzy sets and systems, earlier than 2001, to trace the origins of anything that resembles it. Although perceptual computing is a subset of CWW, it has a much longer history than CWW, as is demonstrated next.

The earliest article that we found that demonstrates an approach for making subjective judgments using FSs is by Tong and Bonissone (1980). In their words:

> A technique for making linguistic decision is presented. Fuzzy sets are assumed to be an appropriate way of dealing with uncertainty, and it is therefore concluded that decisions taken on the basis of such information must themselves be fuzzy. It is inappropriate then to present the decision in numerical form; a statement in natural language is much better. The basic problem is to choose between a set of alternatives $\{a_i : i = 1,$

> . . . , $m\}$, given some fuzzy information about the "suitability" of each of them. This information is given as a set of fuzzy sets, $\{S_i : i = 1, \ldots, M\}$, where each of the $S_i$ is defined by a membership function that maps the real line onto a closed interval [0,1]. Suitability is simply interpreted as a measure of the ability of an alternative to meet our decision criteria and is essentially a fuzzification of the idea of a rating. We have to select the preferred alternative on the basis of $\{S_i : i = 1, \ldots, M\}$ and then generate a linguistic statement about our decision.

Their article includes an example of perceptual computing for making a choice about investments when each of five possible investments is evaluated using four criteria. The resulting investment evaluations use words, that is, they are linguistic. This application has been described in more detail in Section 1.2.1.

Next is the monograph by Schmucker (1984). On the one hand, it contains the essence of perceptual computing, but on the other hand, by today's standards its theoretical depth is not high. Although Schmucker does not use the term perceptual computing, he talks about natural language computations and risk analysis. Figure 1.7, which uses some parts of Fig. 5.1 of his book, is an indication that the three elements of the Per-C are in his fuzzy risk analyzer (FRA). In Schmucker's figure, the "PARSE Natural Language to Fuzzy Set" block contains a collection of words, including more or less, very, normally, fairly, and extremely; the CWW Engine is the fuzzy weighted average;[6] and, the Decoder uses best fit, successive approximation, or piecewise decomposition.

Schmucker states:

> It is the goal of the system designer of an automated risk analysis facility to (1) have a sufficiently rich set of primary terms and hedges so that the user feels almost unrestricted in his range of expression, and (2) associate with each possible natural language expression that can be generated by rules a technical precise meaning that is consistent with the imprecise nebulous English meaning.

Zadeh (1996) summarizes CWW using a figure like the one in Fig. 1.8 (it is Part b of his Fig. 3). He states [Zadeh (1999)]:

> Computing with words (CW) is inspired by the remarkable human capability to perform a wide variety of physical and mental tasks without any measurements and any computations. . . . Underlying this remarkable capability is the brain's crucial ability to manipulate perceptions. . . . Manipulation of perceptions plays a key role in human recognition, decision and execution processes. As a methodology, computing with words provides a foundation for a computational theory of perceptions—a theory which may have an important bearing on how humans make—and machines might make—perception-based rational decisions in an environment of imprecision, uncertainty and partial truth. . . . A basic difference between perceptions and measurements is that, in general measurements are crisp whereas perceptions are fuzzy. . . . The computational theory of perceptions, or CTP for short is based on the methodology of CW. In CTP, words play the role of labels of perceptions and, more generally, perceptions

[6] The fuzzy weighted average is covered in Chapter 5.

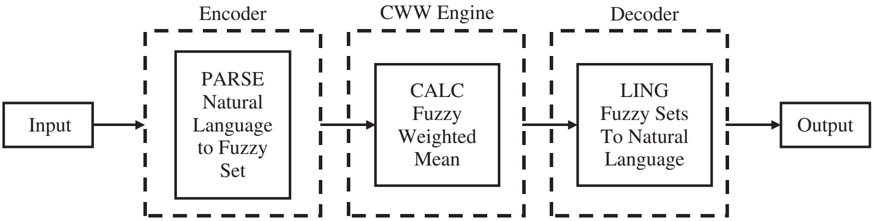


**Figure 1.7.** Schmucker's (1984) FRA. The dashed blocks and their associated labels which relate his blocks to the Per-C were put in by us.

> are expressed as propositions in a natural language. CW-based techniques are employed to translate propositions expressed in a natural language into what is called a Generalized Constraint Language (GCL). In this language, the meaning of a proposition is expressed as a generalized constraint *X isr R*, where *X* is the constrained variable, *R* is the constraining relation and *isr* is a variable copula in which *r* is a variable whose value defines the way in which *R* constrains *X*. Among the basic types of constraints are: possibilistic, veristic, probabilistic, random set, Pawlak set, fuzzy graph and usuality. . . . In CW, the initial and terminal data sets, IDS and TDS, are assumed to consist of propositions expressed in a natural language. These propositions are translated, respectively, into antecedent and consequent constraints. Consequent constraints are derived from antecedent constraints through the rules of constraint propagation. The principal constraint propagation rule is the generalized extension principle. The derived constraints are retranslated into a natural language, yielding the terminal data set (TDS).

Some of the blocks in Fig. 1.8 have been enclosed in dashed shapes so that this figure conforms to the Per-C in Fig. 1.2. The two blocks called "propositions in NL" (natural language) and "initial data set (IDS)" comprise our encoder. We have interpreted those blocks to mean: establishing a vocabulary for an application, collecting data about the words in that vocabulary, and modeling the words as fuzzy

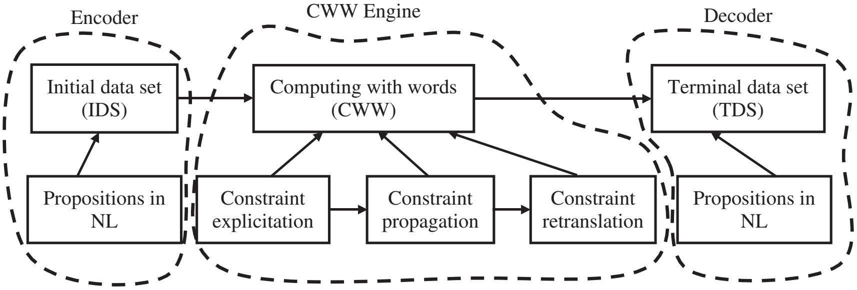


**Figure 1.8.** Zadeh's (1996) CWW. The dashed shapes and their associated labels which relate the enclosed blocks to the Per-C, were put in by us (Zadeh, 1996; © 1996, IEEE).

sets. The two blocks called "propositions in NL" and "terminal dataset (TDS)" comprise our decoder. We have interpreted those blocks to mean mapping the FS output from the CWW block into a linguistic recommendation. Finally, the four blocks called "constraint explicitation," "constraint propagation," "computing with words (CWW)," and "constraint retranslation" are our CWW engine. We have interpreted these four blocks to mean choosing and implementing a specific CWW engine.

Buckley and Feuring (1999) include Fig. 1.9 that contains within in it a Per-C. In their summary, they state:

> This chapter describes the design of a supervisory fuzzy controller for human operators of a complex plant (nuclear reactor). The human operators are allowed to verbally describe the status of various variables used to control the plant. These verbal descriptions come from a very limited vocabulary recognized by the input translator. The input translator translates these descriptions into fuzzy numbers for input to a fuzzy expert system. The fuzzy expert system processes these fuzzy numbers into fuzzy number outputs describing suggestions to the human operators. The output translator, which is a neural net, takes the fuzzy number output from the fuzzy expert system, and produces verbal suggestions, of what to do, for the human operators. The translation of fuzzy numbers into words is called inverse linguistic approximation.

In Fig. 1.9 verbal evaluations made by a human operator (who is interacting with a complex plant) who has access to a vocabulary of words for the application of supervisory control, are translated into fuzzy numbers by the input translator (hard-

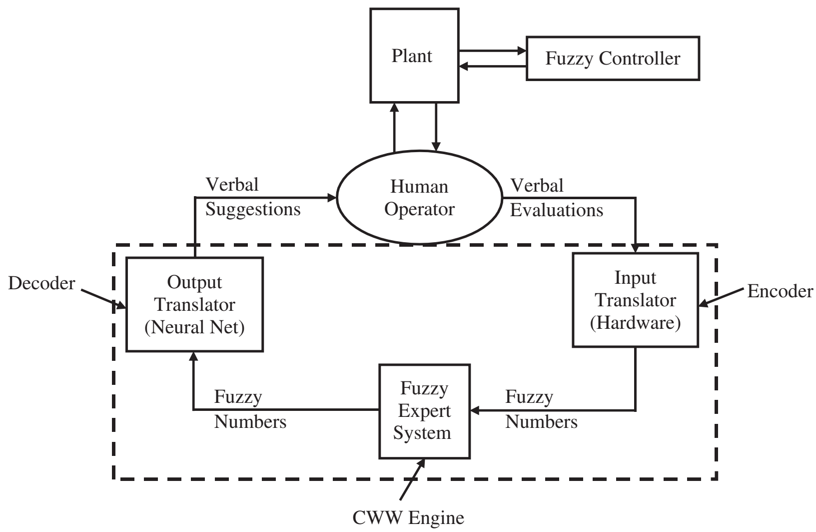


**Figure 1.9.** Buckley and Feuring's (1999) supervisory fuzzy controller. The dashed block and its associated labels relate the other blocks to the Per-C, and were put in by us (Buckley and Feuring, 1999; © 1999, Springer-Verlag).

ware). This is equivalent to our encoder. The translator's fuzzy numbers are processed by a fuzzy expert system whose outputs are other fuzzy numbers. Clearly, the fuzzy expert system is equivalent to one kind of CWW engine. Its output fuzzy numbers are translated into verbal suggestions by the output translator (neural net). This is equivalent to our Decoder.

Finally, Yager (1999, 2004) has the diagram shown in Fig. 1.10.[7] Clearly, translation, manipulation, and retranslation are synonymous with our Encoder, CWW engine, and decoder. In his 2004 article, the manipulation block is called inference/granular computing, and he states:

> We shall assume that as a result of our inference process we obtain the proposition $V$ is $A$, where $A$ is a fuzzy subset of the universe $X$. Our concern is to express this with a natural language statement. The process of retranslation is one of substituting the proposition $V$ is $F$ for $V$ is $A$, where $F$ is some element from $\Im$ and then expressing the output as $V$ is $L$, where $L$ is the linguistic term associated with $F$. The key issue in this process is the substitution of $V$ is $F$ for $V$ is $A$.

The conclusions to be drawn from this brief historical foray are:

- The elements of the perceptual computer did not originate in Mendel (2001, 2002).
- Tong and Bonissone should be credited with originating the perceptual computer, although they did not call it by that name; but, as William Shakespeare wrote: "What's in a name?"
- Additionally, the essence of perceptual computing has been reinvented a number of times and no doubt will continue to be reinvented.

## 1.4 HOW TO VALIDATE THE PERCEPTUAL COMPUTER

It is our belief[8] that for the Per-C to be successful it must provide end users with results that are equivalent to those from a human. This agrees in spirit with what the great computer scientist and philosopher Alan Turing (1950) [see, also, Hodges (1997)] proposed as a test, the Turing Test for machine intelligence. This test is as applicable to perceptual computing as it is to machine intelligence, because perceptual computing is a form of artificial intelligence.

Consider an "imitation game" played with three players, a human being, a machine and an interrogator. The interrogator stays in a room apart from the others. The object is for the interrogator to determine which of the others is the human being or the machine. If the machine cannot be distinguished from the human being

[7]Yager's interests in CWW can be found as early as 1981 [Yager (1981)]. Although the three elements of a perceptual computer are not in the paper, the methodology of the paper is that of CWW; for example, his main example only uses words.

[8]The material in this section is taken from Mendel (2007c).


**Figure 1.10.** Yager's (1999) CWW diagram (Yager, 1999; © 1999, Springer-Verlag).

under these conditions, then the machine must be credited with human intelligence [Hodges (1997)]. This is the essence of a Turing Test.

According to Saygin et al. (2000), "The Turing Test is one of the most disputed topics in artificial intelligence and cognitive sciences," because it can be interpreted in many different ways, for example, by philosophers, sociologists, psychologists, religionists, computer scientists, and so on. We are not interested in using a Turing Test to establish whether the Per-C can think so as to demonstrate that it is intelligent. We are interested in using a Turing Test, as explained in Saygin et al. (2000, p. 467), as "a test to assess a machine's ability to pass for a human being."

In order to implement the Per-C, data will be needed. This data must be collected from people who are similar to those who will ultimately be interacting with the Per-C. If such data collection is feasible, then the design of the Per-C can proceed by using some of the data for training[9] and the rest for validation (testing).[10] The validation of the designed Per-C using some of the collected data (the validation set) can be interpreted as a Turing Test.

If, on the other hand, such data collection is not feasible, then the designer of the Per-C must fabricate it or, even worse, design the Per-C using no data at all. After such a design, the Per-C will have to be validated on a group of subjects, and such a validation will again constitute a Turing Test.

Hence, one way or another, validation of a Per-C is accomplished through a Turing Test.

## 1.5 THE CHOICE OF FUZZY SET MODELS FOR THE PER-C

Because words can mean different things to different people, it is important to use an FS model for a word that lets us capture word uncertainties.[11] At present, there are two possible choices, a type-1 (T1) FS or an interval type-2 (IT2) FS[12] [Mendel (2001b, 2003, 2007b)]. These sets are fully covered in Chapter 2, a high-level syn-

[9]When the CWW Engine (Fig. 1.2) is a set of if–then rules (described in Chapter 6), then a training set can be used to optimize the parameters of antecedent and consequent membership functions, to establish the presence or absence of antecedent terms, and to even determine the number of significant rules, after which the optimized rules can be tested using a testing set.

[10]In traditional supervised system design (e.g., using the back-propagation algorithm), the validation dataset is used to determine whether the training should be terminated and the testing dataset is used to evaluate the generalization performance. In this chapter, validation and testing are used interchangeably, and both terms mean to evaluate the performance of the Per-C.

[11]The material in this section is taken from Mendel (2007c).

[12]General type-2 FSs are presently excluded, because they model higher degrees of uncertainty (see Chapter 3, Section 3.2, Premise 2), and how to do this is not generally known.

opsis of which is given at the end of the present chapter, in Section 1.8.1. In order to decide which FS to use as a word model, two different approaches can be taken:

1. Ignore the adage "words can mean different things to different people," use a T1 FS as a word model, design the Per-C, and see if it passes a Turing Test. If it does, then it is okay to use such an FS as a word model.
2. Adhere to the adage "words can mean different things to different people" and try to decide between using a T1 FS and an IT2 FS as a word model before designing the Per-C. Then design the Per-C and see if it passes a Turing Test.

Because we believe in the adage "words can mean different things to different people," the first approach is not taken in this book. Regarding the second approach, in order to choose between using a T1 FS or an IT2 FS as a word model, we shall rely on the great 20th century scientific philosopher, Sir Karl Popper, who proposed *falsificationism* [Popper (1959, 1963) and Thornton (2005)] as a way to establish if a theory is or is not scientific. Falsificationism states:

> A theory is scientific only if it is refutable by a conceivable event. Every genuine test of a scientific theory, then, is logically an attempt to refute or to falsify it, and one genuine counterinstance falsifies the whole theory. [Thornton (2005)]

According to Thornton (2005), by falsifiability Popper meant:

> If a theory is incompatible with possible empirical observations it is scientific; conversely, a theory which is compatible with all such observations, either because, as in the case of Marxism, it has been modified solely to accommodate such observations, or because, as in the case of psychoanalytic theories, it is consistent with all possible observations, is unscientific.

For a theory to be called scientific it must be testable. This means that it must be possible to make measurements that are related to the theory. A scientific theory can be correct or incorrect. An incorrect scientific theory is still a scientific theory, but is one that must be replaced by another scientific theory that is itself subject to refutation at a later date.

We suggest that using either a T1 FS or an IT2 FS as a word model can be interpreted as a scientific theory.[13] Whether or not each FS word model qualifies as a scientific theory, and then if each is a correct or incorrect scientific theory, must, therefore, be questioned.

Many methods have been reported for making measurements about words and then using those measurements to model a word as either a T1 FS or as an IT2 FS. Hence, as explained next, both kinds of FSs are "scientific" word models.

[13]Note that this is very different from T1 FSs and IT2 FSs as mathematics, which are not scientific theories, and about which we should not raise any issues.

Data collection and mapping into the parameters of a T1 membership function (MF) for a word has been reported on by a number of authors [e.g., Klir and Yuan (1995)] and has also been reported on for a T2 MF for a word [Liu and Mendel (2007, 2008), Mendel and Wu (2006, 2007a,b)]. Names for the different T1 methods include: *polling* [Hersch and Caramazza (1976), Lawry (2001)], *direct rating* [Klir and Yuan (1995), Norwich and Turksen (1982, 1984)], *reverse rating* [Norwich and Turksen (1984), Turksen (1986, 1988, 1991), Turksen and Wilson (1994)], *interval estimation* [Cornelissen (2003), Civanlar and Trussel (1986), Dubois and Prade (1986), Zwick (1987)], and *transition interval estimation* [Cornelissen (2003)]. These methods are described in Chapter 3. Names for the different T2 methods are: *person footprint of uncertainty* [Mendel (2007a)], *interval end points* [Mendel and Wu (2006, 2007a,b), Mendel (2007a)], and *interval approach* [Liu and Mendel (2007, 2008)]. These methods are also described in Chapter 3.

The term *fuzzistics* has been coined [Mendel (2003b, 2007a)] for mapping data that are collected from a group of subjects into an FS model, and represents an amalgamation of the words *fuzzy* and *statistics*. It is a term that is used in this book.

Because of the existence of both type-1 and type-2 fuzzistic's works, we conclude that using type-1 or interval type-2 fuzzy sets as models for words is scientific.

That using a T1 FS model for a word is an incorrect scientific theory follows from the following line of reasoning [Mendel (2003b)]:

- A T1 fuzzy set $A$ for a word is well-defined by its MF $\mu_A(x)$ ($x \in X$) that is totally certain once all of its parameters are specified.
- Words mean different things to different people and so are uncertain.
- Therefore, it is a contradiction to say that something certain can model something that is uncertain.

In the words of Popper, associating the original T1 FS with a word is a "conceivable event" that has provided a "counterinstance" that falsifies this approach to fuzzy sets as models for words.

Chapter 3 explains that an IT2 FS model for a word is only a first-order uncertainty model; hence, an IT2 FS is a scientifically correct first-order uncertainty model for a word and is the one used in this book.[14] As a result, the Fig. 1.2 diagram for the Per-C is modified to the diagram in Fig. 1.11, in which "FS" has been replaced by "IT2 FS."

An objection may be raised that a fixed MF also applies to an IT2 FS model; that is, once the parameters of an IT2 FS model are specified, there no longer is anything uncertain about the IT2 FS. This objection is incorrect because the IT2 FS is a first-order uncertainty model, that is, at each value of the primary variable the MF is an interval of values. For a T1 FS, the MF is a point value, and it is the interval nature of

[14] In the future, perhaps the scientifically correct IT2 FS model for a word will be falsified by a more complete T2 FS model. This will only be possible when more kinds of data than are described in Chapter 3 can be collected about words, or if the data that are presently collected are reinterpreted.

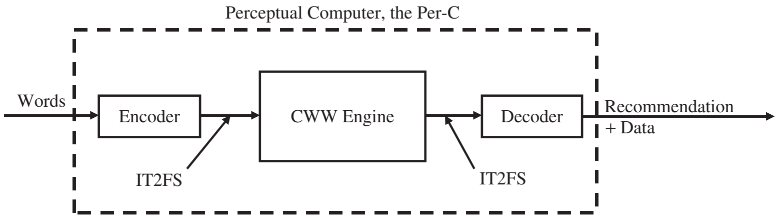


**Figure 1.11.** The perceptual computer that uses IT2 FS models for words (Mendel, 2007c; © 2007, IEEE).

the MF that provides the uncertainty in the IT2 FS model. This argument is similar to one that can be given for a probability distribution function. Once we agree that such a function does indeed model unpredictable (random) uncertainties, then fixing its parameters does not cause us to conclude that it no longer is a probability model.

One may argue that a T1 FS model for a word is a model for a *prototypical word* [Rosch (1975, 1983)]; however, if one also believes that words mean different things to different people, then this calls into question the concept of a prototypical word.

When random uncertainties are present, most of us have no problem with using probability models and analyses from the very beginning; hence, when linguistic uncertainties are present, we suggest that one must have no problem with using IT2 FS models and analyses from the very beginning. Some may ask the question, "How much linguistic uncertainty must be present before an IT2 FS should be used?" Maybe, in the very early days of probability, a similar question was asked; however, it no longer seems to be asked. When randomness is suspected, probability is used. So, when linguistic uncertainties are suspected, IT2 FSs should be used.

Finally, even a Per-C that is designed using IT2 FSs needs to be validated by a Turing Test. The difference in this second approach is that the design is begun using an FS word model that is scientifically correct. This, in itself, does not mean that the resulting Per-C will pass a Turing Test, because that test is applied to the outputs of the Per-C, and it is (Fig. 1.11) the combination of a scientifically correct FS input word model, the CWW engine, and a good decoder that leads to the output recommendation.

Consequently, in this book IT2 FSs are used to model words.

## 1.6 KEEPING THE PER-C AS SIMPLE AS POSSIBLE

Many choices have to be made when designing a Per-C.[15] For example, if the CWW engine is a set of if–then rules (see Chapter 6), then choices must be made about:

- Shapes of MFs for each IT2 FS.

[15]The material in this section is taken from Mendel (2007c).

- Mathematical operators used to model the antecedent connector words *and* and *or*. Such operators are called t-norms and t-conorms, respectively, and there are many t-norms and t-conorms to choose from [e.g., Klir and Yuan (1995)].
- Implication operators (an if–then rule is mathematically modeled using an implication operator), and there are many such operators [e.g., Klir and Yuan (1995)].
- How to aggregate fired rules, i.e., when more than one rule is fired, rule outputs must be combined (aggregated), and there are many different ways to do this [e.g., Klir and Yuan (1995)]. The result is an aggregated IT2 FS.
- How to go from the aggregated IT2 FS to a word, that is, the decoder-design in which, for example, a similarity measure is used, and there are many kinds of similarity measures [e.g., (Wu and Mendel (2008a)].

On the one hand, it is the multitude of choices that provide fuzzy logic with versatility and flexibility. On the other hand, having so many choices with none to very few guidelines on how to make them is confusing.

How does one make the choices needed to implement a Per-C?

Occam's (or Ockham's) Razor[16] is a principle attributed to the 14th century logician and Franciscan friar, William of Occam. The most useful statement of the principle is, when you have two competing theories that make exactly the same predictions, the one that is simpler is the better. This principle is sometimes misstated as "keep it as simple as possible." One can have two (or more) competing theories that lead to different predictions. Occam's Razor does not apply in that case, because the results that are obtained from the competing theories are different.

All of our fuzzy set and fuzzy logic operators originate from crisp sets and crisp logic. In the crisp domain, although there can be many different operators, they all give the same results; hence, we propose that, for the Per-C, Occam's Razor should be applied to the multitude of t-norm, t-conorms, implication operators, and so on in the *crisp domain.* It should not be applied after the operators have been fuzzified, because then it is too late as they give different results. By this argument, one would choose, for example, minimum or product t-norm and maximum t-conorm, because they are simplest t-norms and t-conorm.

Finally, note, that even a Per-C that is designed using IT2 FSs and the "simplest" operators needs to be validated by a Turing Test. If, for example, a Per-C that uses the simplest operators does not pass a Turing Test, then more complicated operators should be used.

## 1.7 COVERAGE OF THE BOOK

Many of the chapters in this book are very technical in nature, because to really understand the Per-C so that one can apply it and extend it to new situations, it is our firm belief that one must master the details. Realizing that some readers will be

[16]See Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/William_of_Ockham, "Occam's Razor" in http://en.wikipedia.org/wiki/Occam%27s_razor, or "What is Occam's Razor" in http://www.weburbia.com/physics/occam.html. Accessed on Jan. 1, 2010.

more interested in the application chapters (Chapters 7–10) rather than in the detail chapters (Chapters 2–6), a summary is given in Section 1.8 for each detail chapter that provides the applications-oriented reader with high-level understandings of the main points of the chapter. After reading these five summaries, it should be possible to read Chapters 7–10.

Chapter 2 is about IT2 FSs since, as has been argued above, they are the ones used by the Per-C. The coverage of these FSs is extensive, but all of the concepts and results of this chapter are used in later chapters of the book, so they must be mastered. Chapter 2 begins with a brief review of T1 FSs. It includes careful definitions of many new terms that are associated with IT2 FSs and are needed to communicate effectively about such sets, including the *footprint of uncertainty* (FOU) and convexity of an IT2 FS. It also includes a representation of IT2 FSs in terms of type-1 FSs that is extremely useful in that it lets all theoretical results about IT2 FSs be developed using T1 FS mathematics; derivations of set theoretic operations of union, intersection and complement for IT2 FSs; the centroid of an IT2 FS, because it provides a measure of uncertainty of such a FS and is a very widely used calculation in later chapters; properties of the centroid; iterative algorithms for computing the centroid; and cardinality and average cardinality of an IT2 FS.

Chapter 3 is about the encoder, that is, about how to model a word using an IT2 FS. It covers two methods for doing this, one called the *person footprint of uncertainty method* and the other called the *interval approach method.* The person footprint of uncertainty method can only be used by persons who are already familiar with interval type-2 fuzzy sets because they must provide a footprint of uncertainty (defined in Chapter 2) for each word; hence, it is limited to so-called *fuzzy experts.* The interval approach (IA) method is based on collecting interval end-point data from a group of subjects and does not require any a priori knowledge about FSs; hence, it can be used by anyone. Because collecting interval data about words using surveys is so important to the IA, this chapter has extensive discussions about it. The IA makes very heavy use of statistics and is a very practical method for mapping subject's data intervals into an FOU for a word. The resulting FOUs are either interior, left-shoulder, or right-shoulder FOUs, and it is the data that establishes which FOU models a word. The IA is applied to a vocabulary of 32 words and their associated IT2 FS models are obtained. The resulting codebook is frequently used throughout the rest of this book. Hedges are also discussed along with reasons for why we choose not to use them. Finally, methods for eliciting T1 MF information from either a single subject or a group of subjects are described in Appendix 3A.

Chapter 4 is about the decoder, that is, about how to go from IT2 FSs (and associated data) at the output of the CWW engine to a recommendation and associated data. The recommendation may be a word, similarity of a group of words, a rank, or a class. For example, in social judgment advising, the decoder recommendations are words; in investment judgment advising and procurement award judging, the decoder recommendations are rankings and similarities; and in journal publication advising, the decoder recommendations are classes (e.g., accept, rewrite, or reject). To map an FOU to a word, a similarity measure is used. Because the output of the CWW engine is often mapped into a codebook word by the decoder, this FOU must resemble such an FOU; therefore, a successful similarity measure for the Per-C is one that simultaneously measures similarity of both FOU shape and proximity of that FOU to a correct word. Hence, in this chapter several similarity measures for IT2 FSs are reviewed, and reasons are provided for why the Jaccard similarity measure is preferred. Additionally, two ranking methods are reviewed for IT2 FSs and a preferred ranking method is obtained, one that ranks the FOUs according to their centers of centroids. Finally, a classification method is presented that is based on the subsethood between two IT2 FSs. Some similarity measures and ranking methods for T1 FSs are described and tabulated in Appendix 4A.

Chapter 5 is about one family of CWW engines called *novel weighted averages* (NWAs) that are a new and very powerful way to aggregate disparate information ranging from numbers to uniformly weighted intervals of numbers to nonuniformly weighted intervals of numbers to words that are modeled using IT2 FSs. The novel weighted averages are grouped into three categories: *interval weighted average* (IWA), in which weights and subcriteria in the weighted average are described by uniformly weighted intervals of real numbers; *fuzzy weighted average* (FWA), in which weights and subcriteria in the weighted average are described by type-1 fuzzy sets; and *linguistic weighted average* (LWA), in which weights and subcriteria in the weighted average are described by interval type-2 fuzzy set models for words. Alpha-cuts and an alpha-cut function decomposition theorem play central roles in computing the FWA and LWA, and so they are reviewed in Chapter 5. Algorithms for computing the IWA, FWA, and LWA are derived. Finally, the *ordered weighted average* (OWA) is described and its relations to NWAs are explained.

Chapter 6 is about one of the most popular CWW engines, called *if–then rules,* and how they are processed so that their outputs can be mapped into a word recommendation by the decoder. We adopt the assumption that the result of combining fired rules must lead to an IT2 FS that resembles the three kinds of FOUs in a CWW codebook, namely, interior and left- and right-shoulder FOUs. This leads to a new way for combining fired rules that is called *perceptual reasoning* (PR), which is a special kind of LWA. The first calculation for PR is a *firing quantity* that may be either a *firing interval* or a *firing level.* The former is computed using the sup–min composition (which should be familiar to people knowledgeable about fuzzy logic systems), whereas the latter is computed using the Jaccard similarity measure (Chapter 4). We prefer a firing level because it leads to FOUs that more closely resemble those in our CWW codebook, whereas using a firing interval does not; hence, later chapters focus exclusively on PR that uses firing levels. Properties are stated and proved for PR that uses firing levels, showing that it leads to IT2 FSs that resemble the three kinds of FOUs in a CWW codebook.

Chapters 7–10 are application chapters. They contain no new theory and illustrate how the Per-C can be used to assist in making investment choices, social judgments, hierarchical decisions, and hierarchical and distributed decisions.

Chapter 7 presents the design of an *investment judgment advisor* (IJA). An investor is given a choice of five investment alternatives—the commodity market, the stock market, gold, real estate, and long-term bonds—and four investment criteria—the risk of losing the capital sum, the vulnerability of the capital sum to modification by inflation, the amount of interest (profit) received, and the cash realizeability of the capital sum (liquidity). The IJA lets an investor provide linguistic ratings for each of the investment alternative's investment criteria, and also linguistic weights for the investment criteria. It then provides the investor with preferential rankings, ranking bands, risk bands, and a similarity array for the investment alternatives so that the investor can establish the components of his/her investment portfolio. The LWA that is explained in Chapter 5 is the basic aggregation tool that is used by the IJA.

Chapter 8 presents the design of a *social judgment advisor* (SJA). The SJA is developed for flirtation judgments, based on if–then rules that are extracted from people. A six-step methodology is presented for designing a SJA. This advisor demonstrates how "I'm getting mixed signals" can be effectively handled within the framework of fuzzy logic, and can be used to sensitize individuals about their (mis-) interpretations of a flirtation situation as compared to the outputs from a consensus flirtation advisor. One of the novel aspects of a SJA is that in an actual flirtation situation, all of the indicators of flirtation will most likely not be observed; hence, a SJA must account for this by means of its architecture, which is an interconnection of subadvisors each for one or two antecedent rules.

Chapter 9 is about how a Per-C can be used to assist in hierarchical decision making. It presents the design of a *procurement judgment advisor* (PJA). 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 his/her final decision, namely: tactics, technology, maintenance, economy, and advancement. Each of these criteria has some associated technical subcriteria; for example, for tactics, the subcriteria are effective range, flight height, flight velocity, reliability, firing accuracy, destruction rate, and kill radius, whereas for economy, the subcriteria are system cost, system life, and material limitation. The contractor creates a *performance evaluation table* in order to assist in choosing the winning company. Contained within this table are three columns, one for each of the three competing companies. Entries into this table for the three companies are evaluations of the subcriteria. Additionally, weights are assigned to all of the subcriteria, because they are not of equal importance. These weights are fuzzy numbers such as around seven, around five, and so on. The subcriteria evaluations range from numbers to words. Somehow, the contractor has to aggregate this disparate information, and this is even more difficult because the five criteria are themselves not of equal importance and have their own linguistic weights assigned to them. Chapter 9 demonstrates how novel weighted averages, which are described in Chapter 5, can be used to assist the contractor in making a final decision.

Chapter 10 is about how the Per-C can be used to assist in hierarchical and distributed decision-making. It presents the design of a *journal publication judgment advisor* (JPJA). When an author submits a paper to a journal, the Editor usually assigns its review to an Associate Editor (AE), who then sends it to at least three reviewers. The reviewers send their reviews back to the AE who then makes a publication recommendation to the Editor based on these reviews. The Editor uses this publication recommendation to assist in making a final decision about the paper. In addition to the "comments for the author(s)," each reviewer usually has to complete a review form in which the reviewer has to evaluate the paper based on two major criteria, technical merit and presentation. Technical merit has three subcriteria: importance, content, and depth, and presentation has four subcriteria: style, organization, clarity, and references. Each of the subcriteria has an assessment level that is characterized by some words. A reviewer chooses one assessment level by checking off one of the words. Usually, the reviewer is also asked to give an overall evaluation of the paper and make a recommendation to the AE. The AE then makes a final decision based on the opinions of the three reviewers.

This evaluation process is often difficult and subjective. The JPJA automates the entire process and does not require that the reviewer provide an overall evaluation of the paper. Instead, this is done by the JPJA using LWAs (Chapter 5) followed by classification into one of the classes called accept, rewrite, or reject. This has the potential to relieve much of the burden of the reviewers and the AE, and, moreover, it may be more accurate and less subjective.

Chapter 11 is where we wrap things up. It summarizes the methodology of perceptual computing and provides proposed guidelines for when something should be called computing with words.

## 1.8 HIGH-LEVEL SYNOPSES OF TECHNICAL DETAILS

In this section, high-level synopses are provide for the most important technical details that are elaborated upon in Chapters 2–6, so that the applications-oriented reader, who may not be interested in those details, can go directly to the application chapters (Chapters 7–10).

### 1.8.1 Chapter 2: Interval Type-2 Fuzzy Sets

Consider [Mendel (2001b)] the transition from ordinary sets to fuzzy sets. When we cannot determine whether the membership of an element in a set is 0 or 1, we use fuzzy sets of type-1. Similarly, when the circumstances are so fuzzy that we have trouble determining the membership grade even as a crisp number in [0,1], we use fuzzy sets of type-2. A type-1 fuzzy set (T1 FS) has a grade of membership that is crisp, whereas an interval type-2 FS (IT2 FS) has grades of memberships that are fuzzy, so it could be called a "fuzzy fuzzy-set." Symbol $A$ is used for a T1 FS, whereas symbol $\tilde{A}$ is used for an IT2 FS (or, for that matter any T2 FS).

Imagine blurring the type-1 membership function depicted in Fig. 1.12 (a) by shifting the points on the triangle either to the left or to the right and not necessarily by the same amounts, as in Fig. 1.12(b). Then, at a specific value of $x$, say $x'$, there no longer is a single value for the membership function; instead, the membership function takes on values wherever the vertical line intersects the blur. When all of those values are weighted the same for all $x'$, one obtains an interval type-2 fuzzy set. Such a FS, $\tilde{A}$, is completely described by its footprint of uncertainty (FOU), $FOU(\tilde{A})$, an example of which is depicted in Fig. 1.13. The $FOU(\tilde{A})$, in turn, is completely described by its lower and upper membership functions, $LMF(\tilde{A})$ and

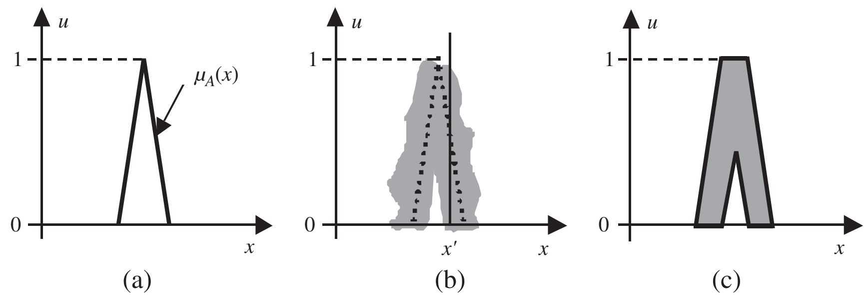


**Figure 1.12.** (a) Type-1 membership function, (b) blurred type-1 membership function, and (c) FOU for an IT2 FS (Mendel, 2001; © 2001, Prentice-Hall).

$UMF(\tilde{A})$. Although these functions are shown as triangles in Fig. 1.13, they can have many other shapes, for example, trapezoids or Gaussian.

It is very easy to compute the union, intersection, and complement of IT2 FSs just in terms of simple T1 FS operations that are performed only on LMFs or UMFs of IT2 FSs. This makes such FSs very useful for practical applications.

Examining Fig. 1.13, one senses that the uncertainty about an IT2 FS must be related to how much area is enclosed within the FOU, that is, a thinner and narrower FOU has less uncertainty about it than does a fatter and broader FOU. The centroid of $\tilde{A}$, $C_{\tilde{A}}$, provides a measure of the uncertainty about such an FS. It is an interval of numbers that has both a smallest and a largest value, that is, $C_{\tilde{A}} = [c_l(\tilde{A}), c_r(\tilde{A})]$, and $c_r(\tilde{A}) - c_l(\tilde{A})$ is small for thin FOUs and is large for fat FOUs. The trick is to compute $c_l(\tilde{A})$ and $c_r(\tilde{A})$. Unfortunately, there are no closed-form formulas for doing this; however, Karnik and Mendel (2001) have developed iterative algorithms, now known as KM algorithms, for computing $c_l(\tilde{A})$ and $c_r(\tilde{A})$. These algorithms are very heavily used in this book.

Cardinality of a crisp set is a count on the number of elements in that set. The

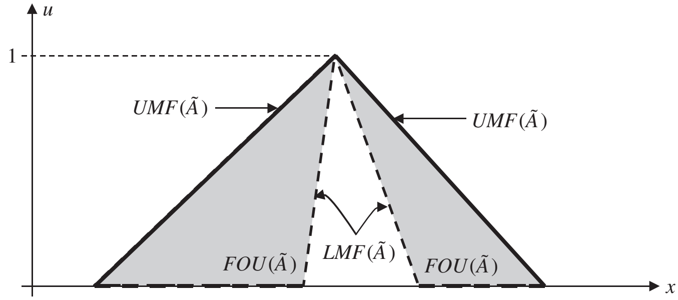


**Figure 1.13.** FOU for an IT2 FS $\tilde{A}$. The FOU is completely described by its lower and upper membership functions (Mendel, 2007c; © 2007, IEEE).

cardinality of an IT2 FS $\tilde{A}$ is an interval of numbers, the smallest number being the cardinality of $LMF(\tilde{A})$ and the largest number being the cardinality of $UMF(\tilde{A})$. The average cardinality of $\tilde{A}$ is the average of these two numbers.

### 1.8.2 Chapter 3: Encoding: From a Word to a Model—The Codebook

Words mean different things to different people, so they are uncertain; hence, as we have argued earlier in this chapter, a FS model is needed for a word that has the potential to capture its uncertainties. An IT2 FS is used as a FS model of a word because it is characterized by its FOU and, therefore, has the potential to capture word uncertainties.

In order to obtain an IT2 FS model for a word, the following are required: (1) a continuous scale must be established for each variable of interest, and (2) a vocabulary of words must be created that covers the entire scale. Our methods are described for the continuous scale numbered 0–10.

For perceptual computing, one begins by establishing a vocabulary of application-dependent words, one that is large enough so that 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 ($A$), that is, Codebook = $\{(\tilde{W}_i, FOU(\tilde{W}_i)), i = 1, \ldots, N_A\}$.

The term *fuzzistics,* which is a merging of the words *fuzzy* and *statistics,* was coined by Mendel (2003b) to summarize the problem of going from word data collected from a group of subjects, with their inherent random uncertainties that are quantified using statistics, to a word fuzzy set model that captures measures of the word data uncertainties. When the FS model is an IT2 FS, this is called *type-2 fuzzistics.*

After a scale is established and a vocabulary of words is created that is believed to cover the entire scale, interval end-point data are collected from a group of subjects. The method for doing this consists of two steps: (1) randomize the words, and (2) survey a group of subjects to provide end-point data for the words on the scale.

Words need to be randomized so that subjects will not correlate their word-interval end points from one word to the next. The randomized words are used in a survey whose wording might be:

> Below are a number of labels that describe an interval or a "range" that falls somewhere between 0 and 10. For each label, please tell us where this range would start and where it would stop. (In other words, please tell us how much of the distance from 0 to 10 this range would cover.) For example, the range "quite a bit" might start at 6 and end at 8. It is important to note that not all ranges be the same and ranges can overlap.

Experiences with carrying out such surveys show that they do not introduce methodological errors and that anyone can answer such questions.

Chapter 3 provides a very practical type-2 fuzzistics method, one that is called the *interval approach* (IA) [Liu and Mendel (2007, 2008)]. The IA consists of two parts, a *data part* and a *fuzzy set* (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 prespecified triangle T1 MF [always beginning with the assumption that the FOU is an interior FOU (as in Fig. 1.14), and, if need be, later switching to a shoulder FOU (as in Fig. 1.14)]. Then the parameters of the triangle T1 MF are determined using the data statistics, and the derived T1 MFs are aggregated using union leading to an FOU for a word, and finally to a mathematical model for the FOU.

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 only FOUs that can be obtained for a word using the IA are the ones depicted in Fig. 1.14, and so these FOUs are referred to herein as *canonical FOUs for a word.*

A word that is modeled by an interior FOU has an UMF that is a trapezoid and a 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.

That there are only three canonical FOUs for a word is very different than in function approximation applications of IT2 FSs (e.g., as in fuzzy logic control, or forecasting of time-series) where one is free to choose the shapes of the FOUs ahead of time and many different choices are possible.

### 1.8.3 Chapter 4: Decoding—From FOUs to a Recommendation

The recommendation from the decoder can have several different forms:

1. *Word.* This is the most typical case. For example, for the social judgment advisor developed in Chapter 8, the FOU at the output of the CWW engine needs to be mapped into a word (or a group of similar words) in the codebook

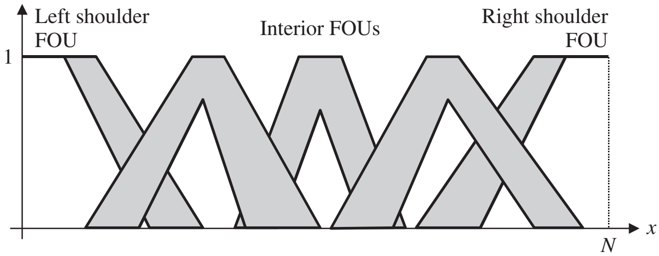


**Figure 1.14.** Left-shoulder, right-shoulder and interior FOUs, all of whose LMFs and UMFs are piecewise linear (Liu and Mendel, 2008; © 2008, IEEE).

so that it can be understood. Similarity measures that compare the similarity between two FOUs are needed to do this.

2. *Rank.* In some decision-making situations, several strategies/candidates are compared at the same time to find the best one(s). For example, in the investment judgment advisor that is developed in Chapter 7, several investment alternatives are compared to find the one(s) with the best overall match to an investor. In the procurement award judgment advisor developed in Chapter 9, three missile systems are compared to find the one with the best overall performance. Ranking methods are needed to do this.
3. *Class.* In some decision making applications, the output of the CWW engine has to be mapped into a class. For example, in the journal publication judgment advisor that is developed in Chapter 10, the outputs of the CWW engine are IT2 FSs representing the overall quality of a journal article from each reviewer and from the aggregated reviewers, and they need to be mapped into one of three decision categories: accept, revise, and reject. Classifiers are needed to do this.

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.

There are around 50 similarity measures that have been published for T1 FSs, but only six for IT2 FSs. Chapter 4 explains that of these six the Jaccard similarity measure, which utilizes both shape and proximity information about an FOU simultaneously, gives the best results, that is, the Jaccard similarity measure provides a crisp numerical similarity measure that agrees with all three of the previous statements.

Simply stated, the Jaccard similarity measure is the ratio of the average cardinality of the intersection of two IT2 FSs to the average cardinality of the union of the two IT2 FSs. The average cardinality is defined in Chapter 2 and is easy to compute.

There are more than 35 methods for ranking T1 FSs, but only two methods for ranking IT2 FSs. Chapter 4 focuses on one of those methods, one that is very simple and based on the centroid of an IT2 FS. First, the centroid (Chapter 2) is computed for each FOU, and then the center of each centroid is computed, after which the average centroids for all FOUs are sorted in increasing order to obtain the rank of the FOUs.

The classification literature is huge [e.g., Duda et al. (2001)]. Our classifiers are based on subsethood, which defines the degree of containment of one set in another. Subsethood is conceptually more appropriate for a classifier than similarity because $\tilde{A}$ and class-FOUs belong to different domains (e.g., in Chapter 10, the FOU for quality of a journal article is classified into accept, rewrite, or reject). The subsethood between two IT2 FSs, $\tilde{A}$ and $\tilde{B}$, $ss(\tilde{A}, \tilde{B})$, may either be an interval of numbers, $ss(\tilde{A}, \tilde{B}) = [ss_l(\tilde{A}, \tilde{B}), ss_r(\tilde{A}, \tilde{B})]$ or a single number. We prefer to use a single subsethood number for our classifiers.

### 1.8.4 Chapter 5: Novel Weighted Averages as a CWW Engine

This is the first of two chapters that provide CWW engines. 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 intervals, T1 FSs or IT2 FSs, or a mixture of all these, for them. A *novel weighted average* (NWA) is 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), and linguistic weighted average (LWA).

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 WA is called an IWA, denoted $Y_{IWA}$. 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 WA is called a FWA, denoted $Y_{FWA}$. And, finally, when at least one subcriterion or weight is modeled as an IT2 FS, the resulting WA is called a LWA.

The IWA and FWA are special cases of the LWA; hence, here our focus is only on the latter.[17] The following is a very useful expressive way to summarize the LWA:

$$\tilde{Y}_{LWA} = \frac{\sum_{i=1}^{n} \tilde{X}_i \tilde{W}_i}{\sum_{i=1}^{n} \tilde{W}_i} \tag{1.1}$$

where subcriteria $\tilde{X}_i$ and weights $\tilde{W}_i$ are characterized by their FOUs, and $\tilde{Y}_{LWA}$ 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. How to actually compute $\tilde{Y}_{LWA}$ is described in Chapter 5, and, somewhat surprisingly, KM algorithms are the bread-and-butter tools for the computations. The exact details of the computations are not needed here. What is needed is the recognition that given FOUs for $\tilde{X}_i$ and $\tilde{W}_i$, it is possible to compute $FOU(\tilde{Y}_{LWA})$.

### 1.8.5 Chapter 6: If–Then Rules as a CWW Engine

Chapter 6 is the second of two chapters that provide CWW engines. One of the most popular CWW engines uses if–then rules. Chapter 6 is about such rules and how they are processed within a CWW engine so that their outputs can be mapped into a word recommendation by the decoder. This use of if–then rules in a Per-C is quite different from their use in most engineering applications of rule-based sys-

[17]In Chapter 5, just the opposite is done, that is, it begins with the IWA, then the FWA, and, finally, the LWA, because the FWA is computed using IWAs, and the LWA is computed using FWAs.

tems—fuzzy logic systems (FLSs)—because in a FLS the output almost always is a number, whereas the output of the Per-C is a recommendation. This distinction has some very interesting ramifications and they are also covered in Chapter 6.

By a *rule* is meant an if–then statement, such as

$$R^l: \text{ IF } x_1 \text{ is } F_1^l \text{ and } \cdots \text{ and } x_p \text{ is } F_p^l, \text{ THEN } y \text{ is } G^l \quad l = 1, \ldots, M \tag{1.2}$$

In equation (1.2), $x_i$ are called *antecedents* and $y$ is called a *consequent.* In logic, equation (1.2) is also called an *implication,* and when $F_i^l$ is either a T1 or T2 FS it is called a *fuzzy implication*. There are many mathematical models for a fuzzy implication that have appeared under the subject heading of approximate reasoning, for example, Table 11.1 in Klir and Yuan (1995) lists 14. Each of these models has the property that it reduces to the truth table of material implication when fuzziness disappears, that is, to logical reasoning.

*Rational calculation* (Chater, et al., 2003) is the view that the mind works by carrying out probabilistic, logical, or decision-theoretic operations (e.g., by the truth table of material implication). *Rational description* is the view that behavior can be approximately described as conforming with the results that would be obtained by some rational calculation. For perceptual computing, logical reasoning will not be implemented as prescribed by the truth table of material implication; instead, rational description is subscribed to.

For CWW, our requirement is that the output of the if–then CWW engine should be an FOU that resembles the three kinds of FOUs in a CWW codebook (as explained in Section 1.8.2). 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. Because none of the widely used fuzzy reasoning models lead to FOUs that resemble the three kinds of FOUs in a CWW codebook, a new fuzzy reasoning model is proposed, called *perceptual reasoning*[18] (PR) [Mendel and Wu (2008)]. PR not only fits the concept of rational description, but also leads to FOUs that resemble the three kinds of FOUs in a CWW codebook.

PR consists of two steps:

1. A *firing quantity* is computed for each rule by computing the Jaccard similarity measure between each input word and its corresponding antecedent word, and, if a rule has $p$ antecedents, then taking the minumum of the $p$ Jaccard similarity measures.
2. The IT2 FS consequents of the fired rules are combined using a linguistic weighted average in which the "weights" are the firing quantities and the "subcriteria" are the IT2 FS consequents.

[18]"Perceptual reasoning" is a term coined in Wu and Mendel (2008) because it is used by the Per-C when the CWW engine consists of if–then rules.

The result is an FOU for PR, and, as proved in Chapter 6, this FOU does indeed resemble the three kinds of FOUs in a CWW codebook.

## REFERENCES

J. J. Buckley and T. Feuring, "Computing with words in control," in L. A. Zadeh and J. Kacprzyk (Eds.) *Computing With Words in Information/Intelligent Systems 2: Applications,* Heidelberg: Physica-Verlag, 1999, pp. 289–304.

N. Chater, M. Oaksford, R. Nakisa, and M. Redington, "Fast, frugal and rational: How rational norms explain behavior," *Organizational Behaviour and Human Decision Processes,* vol. 90, no. 1, pp. 63–86, 2003.

S.-M. Chen, "A new method for evaluating weapon systems using fuzzy set theory," *IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans,* vol. 26, pp. 493–497, July 1996.

C.-H. Cheng, "Evaluating weapon systems using ranking fuzzy numbers," *Fuzzy Sets and Systems,* vol. 109, pp. 25–35, 1999.

M. R. Civanlar and H. J. Trussel, "Constructing membership functions using statistical data," *Fuzzy Sets and Systems,* vol. 18, pp. 1–14, 1986.

A. M. G. Cornelissen, The Two Faces of Sustainability: Fuzzy Evaluation of Sustainable Development, Ph.D. Thesis, Wageningen Univ., The Netherlands, 2003.

F. Doctor, H. Hagras, D. Roberts, and V. Callaghan, "A Type-2 fuzzy based system for handling the uncertainties in group decisions for ranking job applicants within human resources systems," in *Proceedings of IEEE FUZZ Conference,* Paper # FS0125, Hong Kong, China, June 2008.

D. Dubois and H. Prade, "Fuzzy sets and statistical data," *European J. of Operational Research,* vol. 25, pp. 345–356, 1986.

R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification,* 2nd ed. New York: Wiley, 2001.

F. Herrera and E. Herrera-Viedma, "Aggregation operators for linguistic weighted information," *IEEE Trans. on Systems, Man and Cybernetics, Part-A: Systems and Humans,* vol. 27, pp. 646–656, 1997.

H. M. Hersch and A. Caramazza, "A fuzzy set approach to modifiers and vagueness in natural language," *J. Experimental Psychology,* vol. 105, no. 3, pp. 254–276, 1976.

A. Hodges, *Turing: A Natural Philosopher,* U.K.: Phoenix, 1997.

N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Information Sciences,* vol. 132, pp. 195–220, 2001.

G. J. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications,* Upper Saddle River, NJ: Prentice-Hall, 1995.

J. Lawry, "An alternative to computing with words," *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 9, Suppl., pp. 3–16, 2001.

F. Liu and J. M. Mendel, "An interval approach to fuzzistics for interval type-2 fuzzy sets," in *Proceedings of FUZZ-IEEE 2007,* pp. 1030–1035, London, U.K., July 2007.

F. Liu and J. M. Mendel, "Encoding words into interval type-2 fuzzy sets using an *interval approach,*" *IEEE Trans. on Fuzzy Systems,* vol. 16, pp 1503–1521, December 2008.

J. M. Mendel, "The perceptual computer: an architecture for computing with words," in *Proceedings of Modeling With Words Workshop* in the *Proceedings of FUZZ-IEEE 2001,* pp. 35–38, Melbourne, Australia, 2001a.

J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions,* Upper-Saddle River, NJ: Prentice-Hall, 2001b.

J. M. Mendel, "An architecture for making judgments using computing with words," *Int. J. Appl. Math. Comput. Sci.,* vol. 12, No. 3, pp. 325–335, 2002.

J. M. Mendel, "Type-2 fuzzy sets: some questions and answers," *IEEE Connections,* Newsletter of the IEEE Neural Networks Society, vol. 1, pp. 10–13, 2003a.

J. M. Mendel, "Fuzzy sets for words: a new beginning," in *Proceedings of FUZZ-IEEE 2003,* St. Louis, MO, pp. 37–42, 2003b.

J. M. Mendel, "Computing with words and its relationships with fuzzistics," *Information Sciences,* vol. 177, pp. 998–1006, 2007a.

J. M. Mendel, "Type-2 fuzzy sets and systems: an overview," *IEEE Computational Intelligence Magazine,* vol. 2, pp. 20–29, Feb. 2007b.

J. M. Mendel, "Computing with words: Zadeh, Turing, Popper and Occam," *IEEE Computational Intelligence Magazine,* vol. 2, pp. 10–17, November 2007c.

J. M. Mendel and D. Wu, "Perceptual reasoning for perceptual computing," *IEEE Trans. on Fuzzy Systems,* vol. 16, pp. 1550–1564, December 2008.

J. M. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 1, forward problems," *IEEE Trans. on Fuzzy Systems,* vol. 14, pp. 781–792, Dec. 2006.

J. M. Mendel and H. Wu, "Type-2 fuzzistics for symmetric interval type-2 fuzzy sets: Part 2, inverse problems," *IEEE Trans. on Fuzzy Systems,* vol. 15, pp. 301–308, April 2007a.

J. M. Mendel and H. Wu, "Type-2 fuzzistics for non-symmetric interval type-2 fuzzy sets: Forward problems," *IEEE Trans. on Fuzzy Systems,* vol. 15, pp. 916–930, October 2007b.

J. M. Mendel, S. Murphy, L. C. Miller, M. Martin, and N. Karnik, "The fuzzy logic advisor for social judgments: A first attempt," in L. A. Zadeh and J. Kacprzyk, (Eds.), *Computing with Words in Information/Intelligent Systems 2, Applications,* pp. 459–483, Heidelberg: Physica-Verlag, 1999.

D.-L. Mon, C.-H. Cheng, and J.-LC. Lin, "Evaluating weapon system using fuzzy analytic hierarchy process based on entropy weight," *Fuzzy Sets and Systems,* vol. 62, pp. 127–134, 1994.

A. M. Norwich and I. B. Türksen, "The construction of membership functions," in *Fuzzy Sets and Possibility Theory: Recent Developments,* (R. R. Yager, Ed.), Oxford: Pergamon Press, pp. 49–60, 1982.

A. M. Norwich and I. B. Türksen, "A model for the measurement of membership and the consequences of its empirical implementation," *Fuzzy Sets and Systems,* vol. 12, pp. 1–25, 1984.

K. Popper, *The Logic of Scientific Discovery* (translation of *Logik der Forschung*), London: Hutchinson, 1959.

K. Popper, *Conjectures and Refutations: The Growth of Scientific Knowledge,* London: Routledge, 1963.

E. Rosch, "Cognitive representations of semantic categories," *J. of Experimental Psychology: General,* vol. 104, pp. 192–233, 1975.

E. Rosch, "Prototype classification and logical classification: the two systems," In Scholnik, E. (Ed.), *New Trends in Cognitive Representation: Challenges to Piaget's Theory,* pp. 73–86, Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.

A. P. Saygin, I. Cicekli, and V. Akman, "Turing test: 50 years later," *Minds and Machines,* vol. 10, pp. 463–518, 2000.

K. S. Schmucker, *Fuzzy Sets, Natural Language Computations, and Risk Analysis,* Rockville, MD: Computer Science Press, 1984.

S. Thornton, "Karl Popper," in *The Stanford Encyclopedia of Philosophy* (Summer 2005 Edition), Edward N. Zalta (ed.), URL = http://plato.stanford.edu/archives/sum2005/entries/popper/.

R. M. Tong and P. P. Bonissone, "A linguistic approach to decision making with fuzzy sets," *IEEE Trans. on Systems, Man, Cybernetics,* vol. 10, pp. 716–723, 1980.

A. M. Turing, "Computing machinery and intelligence," *Mind,* vol. 59, pp. 433–460, 1950.

I. B. Türksen, "Measurement of membership functions," in W. Karwowski and A. Mital (Eds.), *Applications of Fuzzy Set Theory in Human Factors,* Amsterdam: Elsevier, pp. 55–67, 1986.

I. B. Türksen, "Stochastic fuzzy sets: A survey," in J. Kacprzyk and M. Fedrizzi (Eds.), *Combining Fuzzy Imprecision with Probabilistic Uncertainty in Decision Making,* New York: Springer-Verlag, pp. 168–183, 1988.

I. B. Türksen, "Measurement of membership functions and their acquisition," *Fuzzy Sets and Systems,* vol. 40, pp. 5–38, 1991 (especially Section 5).

I. B. Türksen, "Type-2 representation and reasoning for CWW," *Fuzzy Sets and Systems,* vol. 127, pp. 17–36, 2002

I. B. Türksen and I. A. Wilson, "A fuzzy set preference model for consumer choice," *Fuzzy Sets and Systems,* vol. 68, pp. 253–266, 1994.

G.-H. Tzeng and J.-Y. Teng, "Transportation investment project selection with fuzzy multi-objectives," *Transportation Planning Technology,* vol. 17, pp. 91–112, 1993.

T. S. Wallsten and D. V. Budescu, "A review of human linguistic probability processing: General principles and empirical evidence," *Knowledge Engineering Review,* vol. 10, pp. 43–62, 1995.

D. Wu and J. M. Mendel, "A vector similarity measure for linguistic approximation: Interval type-2 and type-1 fuzzy sets," *Information Sciences,* vol. 178, pp. 381–402, 2008a.

D. Wu and J. M. Mendel, "Perceptual reasoning using interval type-2 fuzzy sets: Properties," in *Proceedings of IEEE FUZZ Conference,* Paper # FS0291, Hong Kong, China, June 2008b.

R. Yager, "A new methodology for ordinal multi-objective decisions based on fuzzy sets," *Decision Sciences,* vol. 12, pp. 589–600, 1981.

R. Yager, "Aproximate reasoning as a basis for computing with words," in L. A. Zadeh and J. Kacprzyk (Eds.), *Computing With Words in Information/ Intelligent Systems 1: Foundations,* Heidelberg: Physica-Verlag, 1999, pp. 50–77.

R. Yager, "On the retranslation process in Zadeh's paradigm of computing with words," *IEEE Trans. on Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 34, pp. 1184–1195, 2004.

L. A. Zadeh, "Fuzzy logic = computing with words," *IEEE Trans. on Fuzzy Systems,* vol. 4, pp. 103–111, 1996.

L. A. Zadeh, "From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions," *IEEE Trans. on Circuits and Systems-1, Fundamental Theory and Applications,* vol. 4, pp. 105–119, 1999.

L. A. Zadeh, "A new direction in AI," *AI Magazine,* pp. 73–84, Spring 2001.

L. A. Zadeh, "Toward human level machine intelligence—Is it achievable? The need for a new paradigm shift," *IEEE Computational Intelligence Magazine,* vol. 3, pp. 11–22, August 2008.

R. Zwick, "A note on random sets and the Thurstonian scaling methods," *Fuzzy Sets and Systems,* vol. 21, pp. 351–356, 1987.