JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 15, 53-69 (1999)

A New Environment for Developing Fuzzy Expert Systems

GWO-JEN HWANG
Information Management Department
National Chi Nan University
Puli, Nantou Taiwan 545, R.O.C.
E-mail: gjhwang@ncnu.edu.tw

This work presents a grid-based fuzzy expert system environment, which employs a repertory grid, fuzzy theory and case-based reasoning. The proposed environment provides friendly user interfaces, abundant data types, intelligent knowledge acquisition facilities and an efficient inference engine. To help users make precise decisions, three factors (i.e., feasibility, reliability and certainty) are used during inference. Moreover, several tools are provided to help experts define membership functions and the knowledge of multimedia data types. Furthermore, the proposed approach is evaluated by implementing an expert system for the diagnosis of animal diseases and by performing several tests.

Keywords: expert systems, fuzzy theory, fuzzy reasoning, knowledge bases

1. INTRODUCTION

Artificial intelligence has received extensive attention in recent years. An important branch of artificial intelligence is known as expert systems, which are designed to represent and employ the knowledge of specific fields in problem solving. Expert systems have been widely applied in diverse fields such as science, engineering, diagnosis, and finance to facilitate decision making, particularly in commercial enterprises, governmental agencies and military institutions. Despite its successful application and proven effectiveness, the expert system approach is a difficult and time-consuming way to construct knowledge bases. A difficulty encountered in knowledge acquisition originates from the knowledge representation problem, e.g., how to represent and employ expertise with fuzzy concepts [10], visual concepts and audio feelings in conventional expert systems [8]. Until now, knowledge acquisition has been bottleneck in developing expert systems [6]. In light the above, this work developed a novel expert system environment by applying advanced techniques, such as multimedia interfaces, fuzzy concepts, and knowledge acquisition methodologies. With application of the proposed approach, the environment not only helps knowledge engineers elicit and represent expertise with multimedia types, but also facilitates fuzzy reasoning.

Received August 1, 1997; revised December 5, 1997; accepted June 3, 1998.
Communicated by Shing-Tsaan Huang.
*The authors would like to thank the National Science Council of the Republic of China for financially supporting this study, under contract No NSC-88-2511-S-260-001.

2. REVIEW OF PERTINENT LITERATURE

In a fuzzy expert system, the reasoning procedure involves three primary processes: fuzzification, fuzzy inference, and defuzzification. Fuzzification operations are used to combine a real time input value (e.g., temperature and speed) with stored membership function information to produce fuzzy input values. Fuzzy inference attempts to relate the fuzzified input facts to the premise patterns of fuzzy rules. Defuzzification combines all fuzzy outputs into a specific composite outcome. In addition, membership functions provide a means of translation between linguistic expressions, such as “Tom is very young”, and numerical input facts, such as “Tom is 18 years old”. Consider the fuzzy rules given as follows:

RULE 1: If a person is young and tall
Then he can play basketball
RULE 2: If a person is old
Then he can jog

Assume that the input fact is “John is 67 years old”; the membership function of “age” is depicted in Fig. 1. The vertical reference line in Fig. 1 intersects the young membership function and old membership function at 0.46 and 0.92, respectively, which implies that the expression “John is young” is true with a lower degree (0.46) and that the expression “John is old” is true with a higher degree (0.92). By means of Fuzzification operations, fuzzy input values are produced:

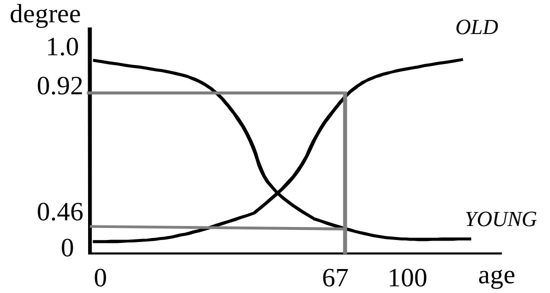


Fig. 1. The membership function of age.

Fact 1: John is young with degree 0.46
Fact 2: John is old with degree 0.92.

Each rule antecedent consists of a set of fuzzy patterns, which might be matched by the fuzzified input facts. In the fuzzy inference process, the system takes the minimum of the rule antecedent terms as the matching degree of the rule. Assume that we have “John is tall with degree 0.8”; the match degree of RULE 1 is minimum (0.46,0.8) = 0.46 and the fuzzy outputs will be

Output 1: play basketball with degree 0.46
Output 2: jogging with degree 0.92.

In some fuzzy systems, the fuzzy output with the maximum truth value is taken as the system output, which has been called maximum defuzzification. If this approach is used, "John can jog" is the final outcome.

In constructing an expert system, knowledge acquisition is known to be a critical bottleneck. To resolve this problem, techniques or methodologies have been widely developed to assist the acquisition of domain expertise. Many knowledge acquisition systems or methodologies have originated from the repertory grid test proposed by Kelly in 1955 [2-7, 9]. A repertory grid is a "two way classification of data in which events are interlaced with abstraction in such a manner to express a portion of a person's system of cross-references between his personal observations or experience of the world and his personal constructs or classifications of that experience [11]". Problem solutions and objects (or elements) are elicited and placed across the top of a grid as column labels, and solution traits, constructs, are listed down the side of the grid as bipolar scales. The conventional approach used to generate constructs is to elicit them from triads; i.e., the objects are presented in triads, and the expert is asked to give a construct to distinguish one object from the other two. Consider a situation in which we want to classify five animals, say, a duck, lion, tiger, eagle, and dolphin. Considering the eagle, lion, and tiger, in what way are two of them alike, and in what way do they differ from one another? Many factors can be considered to answer this question. One of the most obvious answers is "can fly", which can be a construct in the animal grid. A rating scale defines the relationship between an element and a construct. There are usually 5 or 7 scale degrees, and a scale value is given to illustrate the extent to which an object has the construct entity.

Assume that an animal-classification expert system is to be constructed. Initially, an expert fills in the grid to show his perception of a problem and then analyzes the grid to obtain associations and general concepts. After several iterations of repeatedly performing the procedure, a rigid grid is derived as shown in Table 1.

Table 1. An example of a repertory grid.

5	Tiger	Lion	Eagle	Dolphin	Duck	1
can fly	1	1	5	1	2	can not fly
swim	2	2	1	5	3	can not swim
eat meal	5	5	4	1	2	does not eat meal
large size	4	4	2	3	1	small in size
live alone	5	1	1	2	3	lives in a group

A previous investigation [7] proposed an extended model based on repertory grids, EMCUD, to elicit embedded meanings from domain experts. The model consists of a repertory grid (with ratings 5-1 or multiple data types as shown in Table 2) and an Attribute Ordering Table (AOT as shown in Table 3) to represent the degree of significance for each attribute (construct) to each element.

Table 2. An illustrative example to illustrate a repertory grid of multiple data types.

	Obj1	Obj2	Obj3	Obj4	Obj5
A1	{9,10,12}	20	13 < A1 < 16	17	3
A2	YES	NO	YES	YES	NO
A3	X	X	4.3	2.1	6.0

Table 3. A corresponding AOT for the repertory grid given in Table 2.

	Obj1	Obj2	Obj3	Obj4	Obj5
A1	D	D	2	1	D
A2	1	1	1	D	D
A3	X	X	D	1	D

The value of each AOT [attribute, object] entry may be labeled as 'X', 'D' or an integer number, where 'X' implies that the attribute is not related to the object. In addition, 'D' means that the attribute dominates the object; i.e., if the attribute is not equal to the entry value, the object cannot be implied. For an entry which is not labeled 'X' or 'D', an integer is used to represent the relative degree of significance for the attribute to the object. The integer implies that the attribute does not dominate the object but has some degree of significance in relation to other attributes. A larger integer implies that the attribute is more important to the object.

Notably, the integers in the AOT are not limited to 1 and 2. They are used to represent the relative degree of significance for each attribute to each object; i.e., 1 < 2 < 3 < ... < N. The third column in Table 3 expresses the following meanings:

1) A3 dominates Obj3: If A3 is not equal to 4.3, Obj3 cannot be implied.
2) A1 does not dominate Obj3: If A1 ≤ 13 or A1 > 16, Obj3 can be implied.
3) A2 does not dominate Obj3: If A2 NO, Obj3 can still be implied.
4) A1 is more important to Obj3 than is A2 since AOT[Obj3, A1] > AOT[Obj3, A2]: The possibility of implying Obj3 by negating (13 < A1 „T 16) is less than that by negating (A2 = YES) since the negation of the former degrades the degree of certainty more than does negation of the latter.

In the rule-generation phase of knowledge acquisition, an original rule is constructed by combining the values of each column in the repertory with "and" operations. For instance, the following rule is generated from the third column of the repertory grid given in Table 2:

RULE3: $(13 < A1 \leq 16) \wedge (A2 = YES) \wedge (A3 = 4.3) \longrightarrow$ Obj3.

To capture embedded meanings from the original rules, the Attribute-Ordering Table is used. For each rule, each predicate's attribute is verified. If the associative AOT entry is labeled 'D', the predicate remains the same. Those predicates whose related AOT entries are integer numbers are selected based on any possible combination. Each combination of negated predicates leads to the generation of a new rule. The generated rules are called embedded rules. For instance, the embedded rules generated from RULE3 may contain the negated forms of $13 < A1 \leq 16$ or $A2 = YES$ or both. Therefore, the following additional rules are generated:

Rule 3-1: $NOT(13 < A1 \leq 16) \wedge (A2 = YES) \wedge (A3 = 4.3) \rightarrow Obj3$
Rule 3-2: $(13 < A1 \leq 16) \wedge NOT(A2 = YES) \wedge (A3 = 4.3) \rightarrow Obj3$
Rule 3-3: $NOT(13 < A1 \leq 16) \wedge NOT\ (A2 = YES) \wedge (A3 = 4.3) \rightarrow Obj3$.

3. DEVELOPMENT OF A NEW FUZZY EXPERT SYSTEM ENVIRONMENT

Although the repertory grid and EMCUD appear to be effective approaches to eliciting domain expertise, those approaches have certain limitations:

1) Their knowledge description abilities are limited to the binary logic and data types. For instance, "the car is very expensive" cannot be expressed via these approaches.
2) The number of rules can exponentially increase if every combination of the embedded meaning is considered.
3) The certainty factor is the only measure that can be used in these approaches. In some cases, more measures are needed to make further decisions.

Moreover, most expert system environments do not offer fuzzy data types or fuzzy reasoning. Therefore, implementing fuzzy expert systems is difficult and time-consuming. To resolve these problems, this work presents a novel grid-based fuzzy expert system environment, GBFES, which employs the concepts of a repertory grid, fuzzy theory and case-based reasoning. Although its exterior resembles the repertory grid, it directly maintains expertise in each cell rather than keeping the rating of each [element, trait] pair. The system proposed herein can be treated as a case-based approach since the expertise does not require transformation into rules before the reasoning procedure is invoked.

3.1 Data Types

In addition to conventional data types, such as integers, character strings, and Boolean values, GBFES offers fuzzy and graphic data types. Eight built-in fuzzy variables (i.e., "high/low", "long/short", "heavy/light", "big/small", "expensive/cheap", "good/bad", "plenty/scarce" and "positive/negative") and three modifiers (i.e., "very", "ordinary" and "moderately") are provided to describe seven conditions of each fuzzy term (i.e., "very high", "ordinary high", "moderately high", "average", "moderately low", "ordinary low" and "very low"). Moreover, some exceptional cases, i.e., "unknown", "can't decide" and "don't care", are also available.

The graphic data type allows experts to describe their expertise more clearly and easily. For instance, consider an attribute called "the shape of mobile" to classify cars. Clearly describing the "shape" of each type of car is difficult. The meaning of each shape can be definitely presented if graphics are used to represent the "shapes".

3.2 Knowledge Representation

The knowledge base of GBFES consists of an attribute grid and an object grid. The attribute grid (Table 4) holds the attributes elicited from the experts, which can be used to distinguish between objects or determine an object. Every attribute contains five fields: attribute, data type, adjective, dominate threshold, and question. These five fields are briefly described below:

Table 4. An illustrative example of an attribute grid.

Attribute	Data type	Adjective	Dominate Threshold	Question
LEGS	a number		0.1	How many legs?
FLY	yes/no		0.1	Can it fly?
SWIM	fuzzy term	good/bad	0.1	Can it swim?
SPEED	fuzzy term	high/low	0.1	
EAT MEAL	yes/no		0.1	
WEIGHT	fuzzy term	heavy/light	0.1	
Hot Blood	yes/no		0.1	
Reside Place	a text		0.1	
Color	color		0.1	
Fur	fuzzy term	plenty/famine	0.1	

1) Attribute: name of the attribute.
2) Data type: a number, a Boolean value (yes/no), a text, a color or a fuzzy term.
3) Adjective: fuzzy term descriptions, such as "high/low" or "good/bad".
4) Dominate threshold: the threshold value that is used to determine whether or not the object can be a possible solution.
5) Question: the query statement to be invoked when the expert system must acquire the value of some attribute from the user.

The object grid (Table 5) contains the expertise of the application domain. Notably, the knowledge is elicited and stored as a set of cases. The object grid can be created when the contents of the attribute grid are complete.

For every attribute in an attribute-grid, an object grid possesses four corresponding fields: attribute name, weight, dominance, and certainty factor:

1) The attribute name column: Each attribute's name is placed at the top of the column as the field name. Each value in the column represents the relationship between attribute and the corresponding object, which is the pattern used to match the input values obtained from users to observe whether the object is a possible solution.
2) The weight column: Weight is a measure used to compare the degree of significance for each attribute of each object. If attribute A1 is more important than A2 in

Table 5. An illustrative example of an object grid.

	Attribute 1				Attribute 2			
Object	LEGS	W1	D1	C1	FLY	W2	D2	C2
TIGER	4	5	-1	1	0	5	-1	1
LION	4	5	-1	1	0	5	-1	1
Elephant	4	5	-1	1	0	5	-1	1
ZEBRA	4	5	-1	1	0	5	-1	1
CHEETAH	4	5	-1	1	0	5	-1	1
Pronghorn	4	5	-1	1	0	5	-1	1
SHEEP	4	5	-1	1	0	5	-1	1
GOAT	4	5	-1	1	0	5	-1	1
HORSE	4	5	-1	1	0	5	-1	1

determining Obj1, W1 (weight of A1) must be larger than W2 (weight of A2). The weight values range from 0 to 10, where value 0 implies that the attribute has no influence in determining its corresponding object. If the value of an attribute name column is filled with “Unknown”, “Can’t decide”, or “Don’t care”, the corresponding weight value is automatically set to 0 by the system.

3) The dominance indicator: To specify whether or not the “dominate” feature is employed, this indicator is used. Consider a circumstance in which attribute A dominates object B. B cannot be the solution (or conclusion) if the value of A does not match the corresponding pattern in the object grid [7]. For instance, in Table 5, the value of D1 (the “dominate” indicator) of object TIGER is -1 (“yes”), implying that “attribute LEGS dominates object TIGER”. Assume that the system asks the user, “How many legs does the animal have?” and “2” is given as the answer. Since the corresponding pattern in the object grid is “4” (“2” does not match “4”) and the attribute LEGS dominates object TIGER, TIGER is ruled out from the set of candidate solutions. In this case, the system establishes the inference value of LEGS on TIGER as 0, which is smaller than the dominant threshold 0.1. If the “dominate indicator” is “No”, the inference process is continued although part of the input value does not match the corresponding pattern in the object grid.
4) The certainty factor column: Certainty factors reflect the confidence level of elicited expertise. When an expert encounters difficulties in confirming the accuracy of the domain knowledge, the certainty factor is lower.

3.3 Defining Membership Functions

Defining membership functions is an obstacle to constructing fuzzy expert systems. To resolve this problem, a Bezier Curve-based algorithm is employed:

$$X(t) = \sum_{k=0}^{n} X_k C_k^n t^k (1-t)^{n-k} \quad \text{and} \quad Y(t) = \sum_{k=0}^{n} Y_k C_k^n t^k (1-t)^{n-k},$$

where 0 <= **t** <= 1 and **n** denotes the number of control points.

A Bezier curve can be defined by several control points. An important property of a Bezier curve is that it lies within the convex hull (polygon boundary) of the control points. The curve passes through the first and the remaining control points and smoothly follows the other control points without any erratic oscillation. GBFES allows the expert to construct a membership function by giving control points. In addition, the membership function can be easily adjusted by moving the control points.

3.4 Inference Engine of GBFES

The inference process of GBFES is based on the following three measures:

1) Feasibility

In GBFES, the inference result can be several objects, and each object comes with a "likelihood value" ranging from 0 to 1, which represents the feasibility. When a user attempts to select from a set of possible solutions (output objects), the object with the highest feasibility is likely to be the most adequate solution since its attribute values are the closest to those of corresponding patterns. Assume that wij is the weight (degree of importance) relating attribute Aj to object i (Obj i); the feasibility value is calculated using the following formula:

$$\sum_{j=1}^{n} a_j w_{ij} / W \quad \text{where } W = \Sigma w_{ij} \text{ and } \mathbf{n} \text{ denotes the number of attributes.}$$

The value of aj can be either a crisp data type or fuzzy data type. For a crisp data type, if the input value of Aj matches the corresponding pattern in the object grid, we have aj = 1; otherwise, aj = 0. For a scalar-fuzzy-term type, aj is defined as the membership of the input value related to the membership function of Aj. For a non-scalar-fuzzy term, aj = 1 - (the difference between the memberships of Aj and the corresponding object pattern). Each non-scalar-fuzzy term is defined by a membership table with 7 grades as shown in the following table:

0.0	0.165	0.33	0.495	0.66	0.825	0.99
very bad	ordinary bad	somewhat bad	average	somewhat good	ordinary good	very good

Normally, the feasibility of Obji increases as the user gives more inputs. However, if a user inputs "unknown" or "can't decide" for Aj, the system sets wij to zero, implying that Aj has no influence in determining the object.

2) Certainty

Generally, two primary sources of uncertainty may be encountered in an expert system. Restated, there is uncertainty with respect to the validity of domain expertise and uncertainty with respect to the validity of a user's input.

The certainty degree of the first source (confidence factor) reflects the psychological recognition of the expert based on his expertise. A biology expert may either believe it is definitely true or may simply believe that it is true in most cases. For instance, consider the following rules:

RULE1: IF A person is tall
THEN He can usually play basketball.
RULE2: IF X is the father of Y and Y is the father of Z
THEN X is definitely the grandfather of Z.

According to our results, RULE1 and RULE2 have different degrees of certainty.

The second source of uncertainty is associated with the user's input to the expert system's queries. For instance, a user may input a fact similar to "the car may be red"

Consider the following rules:

RULE1: IF A1 and A2 and A3 and A4
THEN Obj1 is true with CF = CR1.
RULE2: IF A5 and A6
THEN Obj1 is true with CF = CR2.

Assume that Ai is known to be true with certainty factor C_{Ai}; we have

$$CF(Obj1)= MAX(MIN(CA1, CA2, CA3, CA4) \times CR1, MIN(CA5, CA6) \times CR2).$$

GBFES, although not a rule-based system, supports a certainty measure that resembles the concept of certainty factors. A certainty field is given for every attribute in our approach. When the expert describes the relationship between attribute Aj and object Obj i, he or she must indicate the corresponding certainty value Cij as well. The range of certainty value in GBFES lies in [0, 1]. For the user, a certainty value must be given to indicate the assurance of each input data.

The method used to calculate the certainty factor in a rule-based system can not be directly applied to GBFES because the relationships among the attributes are neither conjunctive nor disjunctive. Therefore, we have the certainty factor of object Oi being calculated using the following formula:

$$\sum_{j=1}^{n} \frac{w_{ij} c_{ij} c'_j}{W_i},$$

where $W_i = \sum_{j=1}^{n} W_{ij}$, n = the number of attributes, cij = the certainty factor of cj related to Obji and c'j = the certainty factor of the user's reply to Qj (question j).

The certainty factor can help the user make decisions, particularly when two objects have the same (or very close) feasibility. Consider the object grid given in Table 6. Assuming that the dominate threshold is 0.1, we have W1 = 11, W2 = 10 and W3 = 8. An example illustrating the inference process is given as follows:

Step1: If A1 = "2" with CF = 1.0, we have

Obj1: is ruled out since A1= "2" ≠ (Obj1, A1) = "4" and D1= "yes",
Obj2: feasibility = 1 * 2/10 = 0.2, CF = 1* 1.0 *2/10 = 0.2,
Obj3: feasibility = 1 * 3/8 = 0.375, CF = 1* 1.0 * 3/8 = 0.375.

Step2: If A2 = "100" with CF = 0.9 and the degrees of membership of "high" and "very high" are 0.8 and 0.6 respectively, we have:

Table 6. A simple GBFES's object grid with certainty factors.

	A1	W1	D1	C1	A2	W2	D2	C2	A3	W3	D3	C3	A4	W4	D4	C4
Obj1	4	4	y	1.0	high	2	n	0.8	4	4	n	1.0	Good	1	y	0.7
Obj2	2	2	y	1.0	very high	4	n	0.9	4	2	n	1.0	very good	2	n	0.8
Obj3	2	3	y	1.0	high	1	n	0.6	2	1	n	1.0	Good	3	n	0.7

Obj2: feasibility = $1 \times 2/10 + 0.6 \times 4/10 = 0.44$, CF = $1.0 \times 1.0 \times 2/10$
$+ 0.9 \times 0.9 \times 4/10 = 0.524$,
Obj3: feasibility =$1 \times 3/8 + 0.8 \times 1/8 = 0.475$, CF = $1.0 \times 1.0 \times 3/8$
$+ 0.9 \times 0.6 \times 1/8 = 0.4425$.

Step3: If A3 = "unknown":

Obj2: feasibility = $0.44 \times 10/(10\text{-}2) = 0.55$, CF = $0.524 \times 10/(10\text{-}2) = 0.655$,
Obj3: feasibility = $0.475 \times 8/(8\text{-}1) = 0.5428$, CF = $0.4425 \times 8/(8\text{-}1) = 0.5057$.

Step4: If A4 = "very good" with CF = 0.7, we have:

Obj2: feasibility = $1 \times 2/(2 + 4 + 2) + 0.6 \times 4/(2 + 4 + 2) + 1 \times 2/(2 + 4 + 2) = 0.8$ and
CF = $1.0 \times 1.0 \times 2/(2 + 4 + 2) + 0.9 \times 0.9 \times 4/(2 + 4 + 2)$
$+ 0.7 \times 0.8 \times 2/(2 + 4 + 2) = 0.795$.
Obj3: feasibility = $1 \times 3/(3 + 1 + 3) + 0.8 \times 1/(3 + 1 + 3) + 0.8 \times 3/(3 + 1 + 3) = 0.8856$.
CF = $1.0 \times 1.0 \times 3/(3 + 1 + 3) + 0.9 \times 0.6 \times 1/(3 + 1 + 3)$
$+ 0.7 \times 0.7 \times 3/(3 + 1 + 3) = 0.7157$

3) Reliability

Reliability can be considered as the "steadiness" of the feasibility or certainty factor. Assume that the values of "feasibility" and "certainty" of Obj1 and Obj2 are the same (or that Obj1 has a higher feasibility while Obj2 has higher certainty). Reaching a final decision would be extremely difficult. Consider the attributes of Obj1 shown in Table 7. A2 and A3 are labeled "Don't care" and "Unknown" with weight 0. This fact implies that the total weight heavily depends on A1 and A4. Therefore, assuming that Obj2 is more reliable than Obj1 is straightforward since there are more attributes to support Obj2.

Table 7. An object grid to illustrate "reliability".

	A1	W1	D1	C1	A2	W2	D2	C2	A3	W3	D3	C3	A4	W4	D4	C4
Obj1	2	3	y	1.0	**Don't care**	**0**	n	0.9	**Unknown**	0	n	1.0	very good	5	n	0.8
Obj2	2	3	y	1.0	high	1	n	0.6	very low	1	n	1.0	good	3	n	0.7

The concept of entropy illustrates the concept of "reliability" or "steadiness":

$$R_i = -\sum_{j=1}^{n} \frac{W_{ij}}{W_i} \log \frac{W_{ij}}{W_i} \quad \text{where the base of log is 2.}$$

Regarding the example shown in Table 7, the reliability of Obj1 = –(3/8 log 3/8 + 5/8 log 5/8) = 0.9544, and the reliability of Obj2= –(3/8 log 3/8 + 1/8 log 1/8+ 1/8 log 1/8 +3/8 log 3/8) = 1.8108; hence, Obj2 is more reliable than Obj1.

4. IMPLEMENTATION

GBFES was coded using Visual Basic for Windows. Figs. 2 and 3 depict the user interface of GBFES used to maintain the attribute grid.

After the attribute grid becomes available, the expert or the knowledge engineer can

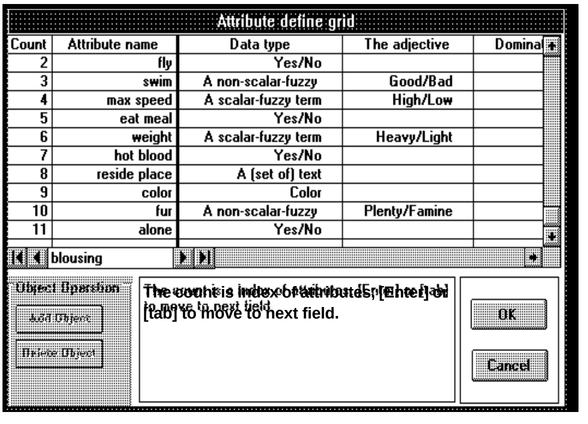


Fig. 2. An illustrative example of an attribute grid.

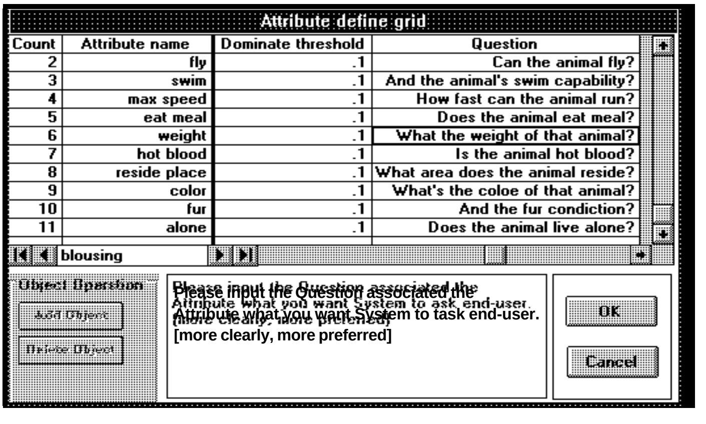


Fig. 3. Question statements and dominant thresholds of the attribute grid.

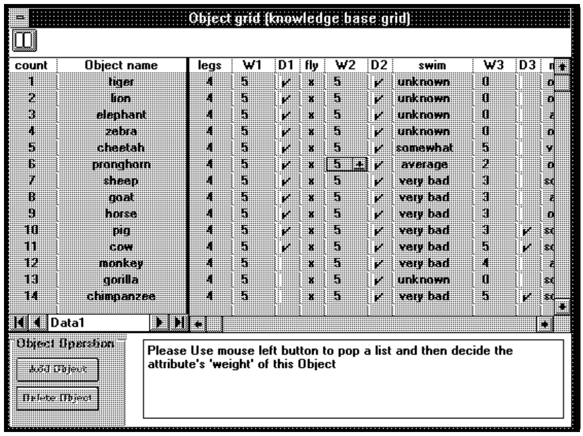


Fig. 4. An illustrative example of an object grid.

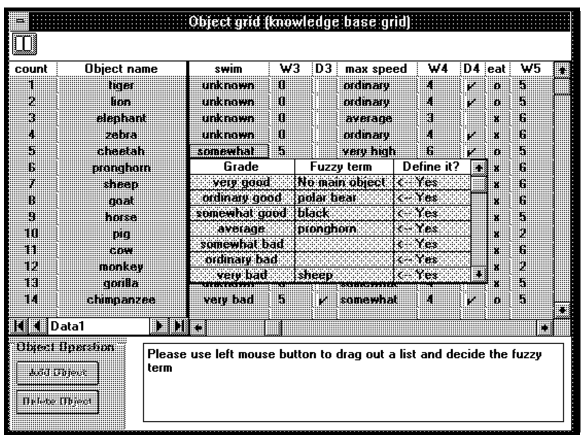


Fig. 5. An illustrative example of defining a non-scalar-fuzzy term.

insert the expertise into the object grid (Fig. 4) by following the directions shown in the right bottom rectangle. For an attribute with a non-scalar-fuzzy data type, a small window pops up to assist the expert (Fig. 5).

If the expert is confronted with a scalar-fuzzy data type, another small window (Fig. 6) pops up to assist him or her. To achieve more flexibility, the system offers a Bezier curve interface which the expert can use to define the membership function (Fig. 7).

When the attribute grid and the object grid are complete, an analysis module can be invoked on the knowledge base to check for typographical errors and to perform similarity analysis among the objects. If two objects are found to be "very similar" (i.e., nearly indistinguishable), the expert is asked to recheck the values in the object grid to provide new attributes so as to make them distinguishable.

For the animal classification problem, an example of inference processes is given as follows:

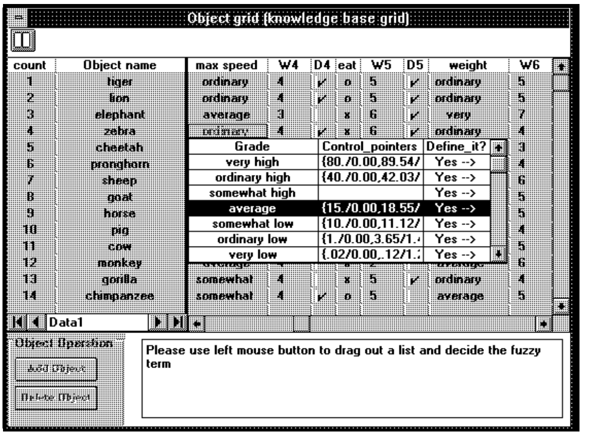


Fig. 6. An illustrative example of defining a scalar-fuzzy term.

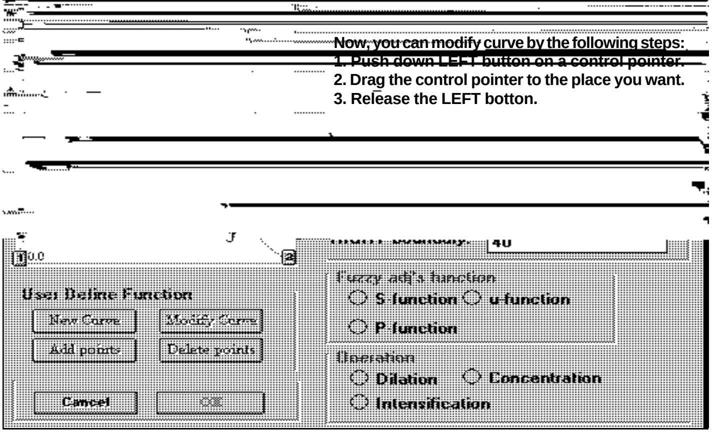


Fig. 7. An illustrative example of defining a membership function.

1) Initially, the user inputs 4 for the number of legs of the animal. Several possible solutions are listed with a certainty value (Fig. 8). Since only weak evidence is known (i.e., the number of legs is 4), all of the certainty values are low.
2) The user inputs new evidence, "the animal can fly" (Fig. 9), and the order of candidate solutions is changed according to the modified certainty values.
3) After several iterations, the final results are obtained as shown in Fig. 10. The best choice is Lion, but Tiger is still a promising candidate.

5. RESULTS AND DISCUSSIONS

Thirty cases of pig diseases were tested to evaluate the performance of GBFES. Each case was described by fourteen attributes: *weak, emaciation, growth, weight, hypothermia, gaunt, prostration, wasting, tachycardia, bradycardia, murmurs, anorexia, constipation and dyschezia*. Table 8 presents an example of a test case.

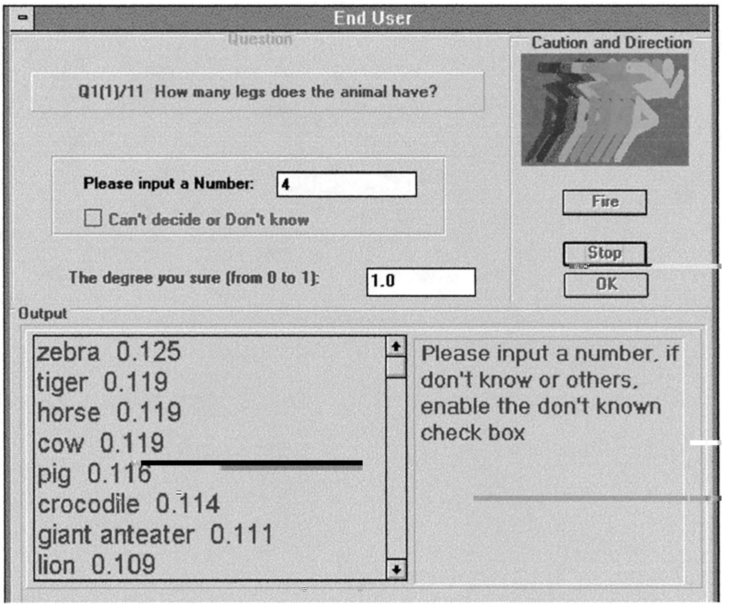


Fig. 8. An illustrative example of the inference process.

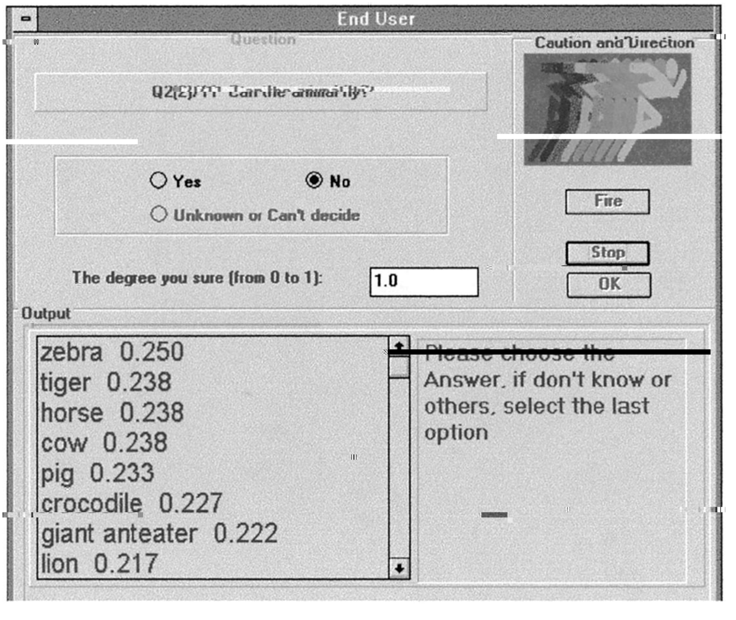


Fig. 9. An illustrative example of the inference process (continued).

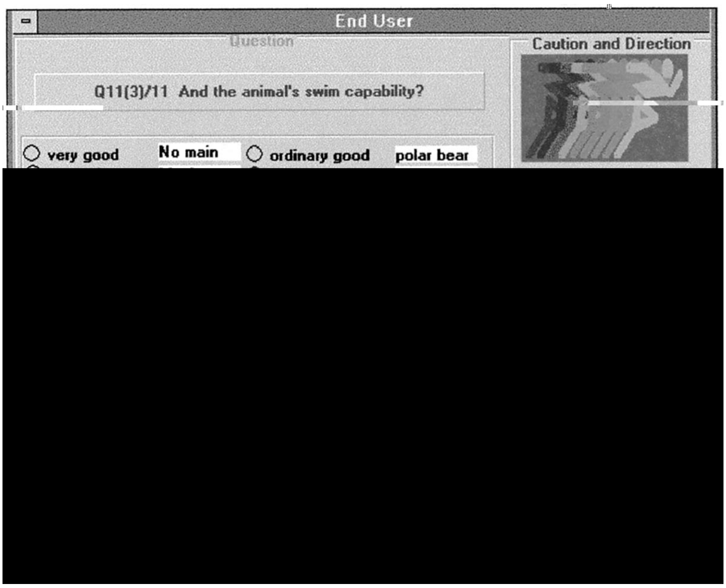


Fig. 10. The cumulative results of the inference process.

Table 8. A disease record of a pig.

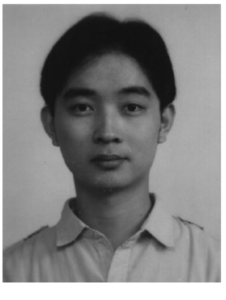
Path. No. F93-123 Clinic No. 587-65
Species: pig Breed: L Age: Nursery Sex: F
Origin : N.F. Clinician,Authanasia
Date and Time of Death or Biopsy: 2/17/93
Date and Time of Necropsy: 2/17/93 Necropsied by : S.F.Young

GE10 Very weak	WE	81/06/04
GE11 Emaciation	EM	81/06/04
GE12 Slow growth	SG	81/06/04
GE13 Retarded growth	RG	81/06/04
GE14 Weight loss	WL	81/06/04
GE15 Hypothermia	HT	81/06/04
GE16 Gaunt	GA	81/06/04
GE17 Prostration	PR	81/06/04
GE18 Wasting	WA	81/06/04
CA01 Tachycardia	TC	81/06/04
CA02 Bradycardia	BC	81/06/04
CA03 Murmurs	MU	81/06/04
AL01 Anorexia	AR	81/06/04
AL02 Constipation	CS	81/06/04
AL03 Dyschezia	DC	81/06/04

Table 9 summarizes the experimental results. Each number in the first row denotes the series number of test cases. The data in the second row and the third row of Table 9 are the answers obtained from a diagnostician and GBFES, respectively. The number for each answer indicates one of the following diseases: *1 - Blue Ear Disease*; *2- Actinobacillus Pneumonia*; *3- Salmonellosis*; *4- Streptococcal*;

REFERENCES

1. J. Adams, "Probabilistic reasoning and certainty factors," in B. Buchanan and E. Shortliffe, (Eds.), *Rule-Based Expert Systems: The MYCIN Experiments of the Standford Heuristic Programming Paper*, Reading, MA: Addison-Wesley, 1985.
2. J. Boose, "A knowledge acquisition program for expert systems based on personal construct psychology," *International Journal of Man-Machine Studies*, No. 23, 1985, pp. 495-525.
3. J. Boose and J. M. Bradshaw, "NeoETS: Capturing expert system knowledge in hierarchical rating grids," *IEEE Expert System in Government Symposium*, 1986, pp. 34-45.
4. J. Boose and J. M. Bradshaw, "Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for knowledge-based systems," *International Journal of Man-Machine Studies*, No. 26, 1987, pp. 3-28.
5. K. M. Ford, F. E. Petry, J. R. Adams-Webber and P. J. Chang, "An approach to knowledge acquisition based on the personal construct systems," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 3, No. 1, 1991, pp. 78-88.
6. B. R. Gaines, "An overview of knowledge acquisition and transfer," *International Journal of Man-Machine Studies*, No. 26, 1987, pp. 453-472.
7. G. J. Hwang and S. S. Tseng, "EMZuD: A knowledge acquisition method which captures embedded meanings under uncertainty," *International Journal of Man-Machine Studies*, No. 33, 1990, pp. 431-451.
8. G. J. Hwang, "Knowledge acquisition for fuzzy expert systems", *International Journal of Intelligent Systems*, Vol. 10, No. 6, 1995, pp. 541-560.
9. G. A. Kelly, *The Psychology of Personal Constructs*, New York: Norton, 1955.
10. G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice-Hall, 1988.
11. M. L. G. Shaw and B. R. Gaines, "KITTEN: Knowledge initiation and transfer tools for experts and novices," *International Journal of Man-Machine Studies*, No. 27, 1987, pp. 251-280.



Gwo-Jen Hwang (黃國禎) was born on April 16, 1963, in Taiwan, Republic of China. In 1991, he received his Ph.D. degree from the Department of Computer Science and Information Engineering of National Chiao Tung University, Taiwan. He is now an associate professor at National Chi-Nan University. He is also the Director of the Computer Center of that university. His research interests include expert systems, fuzzy reasoning, multimedia systems and computer-assisted instructions.