Using fuzzy sets in flexible querying: Why and how?*

Didier DUBOIS and Henri PRADE

Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier – CNRS, 118 route de Narbonne
31062 Toulouse Cedex 4 – France
Email: {dubois, prade}@irit.fr

1 - Introduction

The last few years have witnessed a tremendous increase in the use of computers in more and more domains, the need for managing new kinds of data and for providing new capabilities for storage, access and display of information. In this respect, one may imagine introducing what is often dubbed "uncertainty" into databases. This term may refer to two main streams of problems. On the one hand, one wants to store and manipulate incomplete data (i.e., the available information about attribute values may be tainted with imprecision and/or uncertainty for some items). In that case, the retrieval process will also return results involving some uncertainty (if we are uncertain about the precise value of John's age, we cannot always be sure that John does (or does not) satisfy a given requirement in the context of a query selecting people on basis of their age). On the other hand, the term "uncertainty" is sometimes (and somewhat misleadingly) used for referring to flexible queries, since one may then consider that there is some ambiguity pertaining to their meaning. In fact, flexible queries are useful for describing preferences and thus for getting an ordered set of answers accordingly.

Research in "fuzzy databases" is more than fifteen years old. For a long time the research in this area has been developed by only a dozen of small groups, most of the time outside the main stream of the regular database community. The situation is now slowly evolving with the consolidation of the most important results in recent books solely devoted to the fuzzy database topic (Bosc and Kacprzyk, 1995; Petry, 1996), and with a larger acceptance of the fuzzy set approach by database people.

If we except the recent and promising area of research in fuzzy data mining (e.g., Wu and Mahlén, 1995; Yager, 1996), the fuzzy database literature has been concentrating on three issues:

---

* This is a revised version of the main part of a paper entitled "Using fuzzy sets in database systems: Why and how?" in the Proceedings of the 1996 Workshop on Flexible Query-Answering Systems (FQAS'96) (H. Christiansen, H.L. Larsen, T. Andreasen, eds.), held in Roskilde, Denmark, May 22-24, 1996, pp. 89-103.

– flexible querying (Tahani, 1977; Kacprzyk and Ziolkowski, 1986; Bosc and Pivert, 1995);

– handling of imprecise, uncertain, or fuzzy data (Umano, 1982; Zemankova and Kandel, 1984; Prade and Testemale, 1984; Vandenberghe et al., 1989);

– defining and using fuzzy dependencies (Raju and Majumdar, 1988; Chen et al., 1994; Cubero and Vila, 1994; Bosc et al., 1996).

An introduction to these different issues may be found in a recent survey paper by Bosc and Prade (1997).

These tasks involve three basic semantics which can be naturally attached to a fuzzy set: preference, uncertainty and similarity: i) the flexibility of a query reflects the preferences of the user. Using a fuzzy set representation, the extent to which an object described in the database satisfies a request then becomes a matter of degree; ii) the information to be stored in a database may be pervaded with imprecision and uncertainty. Then ill-known attribute values can be represented by means of fuzzy sets viewed as possibility distributions; iii) close values are often perceived as similar, interchangeable (Buckles and Petry, 1982). Indeed, if for instance an attribute value v satisfies an elementary requirement, a value "close" to v should still somewhat satisfy the requirement. The idea of approximate equality, of similarity plays a key role also in the modelling of fuzzy dependencies.

In the following, we only discuss some questions pertaining to flexible querying, trying to identify what are the problems from a fuzzy set point of view. We focus on representation issues, emphasizing in each case what might be the intended meanings of the flexible queries, and how to capture them in an accurate manner. Mastering the representation tool is clearly important for being able to handle practical problems. Aspects pertaining to database implementations are not discussed here.

2 - Flexible Querying

Fuzzy set membership functions (Zadeh, 1965) are convenient tools for modelling user's preference profiles and the large panoply of fuzzy set connectives can capture the different user attitudes concerning the way the different criteria present in his/her query compensate or not; see (Bosc and Pivert, 1992) for a unified presentation in the fuzzy set framework of the existing proposals for handling flexible queries.

Thus, the interest of fuzzy queries for a user are twofold:

i) A better representation of his/her preferences. For instance, "he/she is looking for an apartment which is not too expensive and not too far from downtown". In such a case, there does not exist a definite threshold for which the price becomes suddenly too high, but rather we have to differentiate between prices which are perfectly acceptable for the user, and other prices, somewhat higher, which are still more or less acceptable (especially if the apartment is close to downtown). Obviously, the meaning of vague predicate expressions like "not too expensive" is context/user dependent, rather than universal. The large panoply of fuzzy set connectives can capture the different user's attitude concerning the way the different criteria present in his/her query compensate or not. Moreover in a given query, some part of the request may be less important to fulfil; this leads to the need for weighted connectives. Elicitation procedures for membership functions and connectives are thus very important for practical applications. A procedure is suggested for connective elicitation in Section 2.4.

ii) Fuzzy queries, by expressing user's preferences, provide the necessary information in order to rank-order the answers contained in the database according to the degree to which they satisfy the query. It contributes to avoid empty sets of answers when the queries are too restrictive, as well as large sets of answers without any ordering when queries are too permissive.

Thus, flexible queries are often motivated by the expression of preferences, and of relative levels of importance. However the use of queries involving fuzzily bounded categories may be also due to an interest for more robust evaluations. This is the case in a query like "find the average salary of the *young* people stored in the database", where the use of a predicate like "young" (whose meaning is clearly context-dependent) does not here refer to the expression of a preference, but is rather a matter of convenience since the user is not obliged to set the boundaries of the category of interest in a precise and thus rather arbitrary way; in such a case, a range of possible values for the average salary instead of a precise number will be returned to the user. This range can be viewed as bounded by the lower and the upper expected values of a fuzzy number; see (Dubois and Prade, 1990). It is a robust evaluation which provides the user with an idea of the variability of the evaluation according to the different possible meanings of young (in a given context).

Making a requirement flexible is not only naturally associated with the idea of a gradual representation reflecting the preferences. It is also connected with the intuitive idea of allowing for a possible weakening of the requirement in some way: by putting some tolerance on it, by assessing its importance (in a sense to be defined), by conditioning it, or by only demanding the satisfaction of *most* of the components of the requirement (if it is a compound one).

2.1 - Relaxing a Constraint

Modeling Tolerance

Two values u_1 and u_2 belonging to a domain U may be considered as approximately equal even if they are not identical. For instance if the pattern requires somebody who is 40 years old, an item corresponding to a person who is 39 may be considered in some cases as approximately matching the request. An approximate equality can be conveniently modelled by means of a fuzzy relation R which is reflexive (i.e., $\forall u$, $\mu_R(u,u) = 1$) and symmetrical (i.e., $\forall u_1$, $\forall u_2$, $\mu_R(u_1,u_2) = \mu_R(u_2,u_1)$). The closer u_1 and u_2 are, the closer to 1 $\mu_R(u_1,u_2)$ must be. The quantity $\mu_R(u_1,u_2)$ can be viewed as a grade of approximate equality of u_1 with u_2. R is then called a proximity or a tolerance relation. Let us assume that we work with an ordinary database where data are precise and certain. When the (elementary) requirement is represented by a subset P of U (P may be fuzzy), the tolerance R can be taken into account in the degree of matching $\mu_P(d)$ of a piece of data d w.r.t. P, by replacing P by the enlarged subset $P \circ R$, defined by

$$\forall u,\ \mu_{P \circ R}(u) = \sup_{u' \in U} \min(\mu_P(u'), \mu_R(u,u')) \geq \mu_P(u). \quad (1)$$

Roughly speaking $P \circ R$ gathers the elements in P and the elements outside of P which are somewhat close to an element in P. The use of tolerance relations in the fuzzy pattern matching was already suggested in Cayrol et al. (1982). Obviously, a tolerance relation is attached to the domain of an attribute variable, and different tolerance relations can be involved in the evaluation of the matching of an item with a compound pattern.

Importance Assignment

One possible way for taking into account the relative importance of elementary requirements in a compound pattern is still to enlarge P by using a very particular tolerance relation. Generally speaking, the elementary pattern P can be all the more enlarged into $P \circ R$ as as it is considered as less important. A first idea for modelling importance is to use a fuzzy relation R, still reflexive and symmetrical, which now models an uncertain strict equality rather than an approximate equality. Namely R is now of the form

$$\mu_R(u_1,u_2) = \begin{cases} 1 \text{ if } u_1 = u_2 \\ 1 - w \text{ if } u_1 \neq u_2. \end{cases} \quad (2)$$

When $w = 1$, this is the usual strict equality and $P \circ R = P$. When $w = 0$, $P \circ R = U$ and the requirement expressed by the pattern will be trivially satisfied. The larger w, the more important the requirement expressed by the pattern. The introduction of w amounts to modifying P into $P^* = P \circ R$ such that

$$\mu_{P^*}(u) = \max(\mu_P(u), 1 - w). \quad (3)$$

As we can see, P* considers as acceptable any value outside of the support of P with the degree $1 - w$. It means that the larger w, the smaller the degree of acceptability of a value outside the support of P.

In case of the logical conjunctive combination of several requirements P_i performed by min-combination (min is the largest associative aggregation operation which extends ordinary conjunction; it is also the only idempotent one), i.e. for a piece of information $d = (u_1, \ldots, u_n)$, we obtain the combination

$$\min_{i=1,\ldots,n} \mu_{P^*_i}(d) = \min_{i=1,\ldots,n} \max(1 - w_i, \mu_{P_i}(d)). \quad (4)$$

with $\mu_{P_i}(d) = \mu_{P_i}(u_i)$ where u_i is the precise value of the attribute pertaining to P_i, and where the following condition should be satisfied by the w_i's:

$$\max_{i=1,n} w_i = 1,$$

if there is one requirement that can eliminate an object d when violated.

Clearly when $w_i = 0$, the degree of matching $\mu_{P_i}(d)$ is ignored in the combination, then P_i has absolutely no importance; the larger w_i, the smaller the degrees of matching concerning P_i which are effectively taken into account in the aggregation. The normalization expresses that the most important requirement has the maximal weight (i.e., 1) and is compulsory.

In the above model, each weight of importance is a constant and thus does not depend upon the value taken by the concerned attribute for the considered object d. This limitation may create some unnatural behaviour of the matching procedure. For instance, the price of an object you are looking for may be of a limited importance only within a certain range of values; when this price becomes very high, this criterion alone should cause the rejection of the considered object, in spite of the rather low importance weight.

To cope with this limitation it has been proposed (Dubois et al., 1988) that the weight of importance become a function of the concerned attribute value. Formally, let s(P) be the support of the fuzzy set associated with the atom P. The weighting function w is supposed to be constant at least on s(P) and to increase for values outside of P, possibly reaching 1 outside s(P). See Fig. 1.

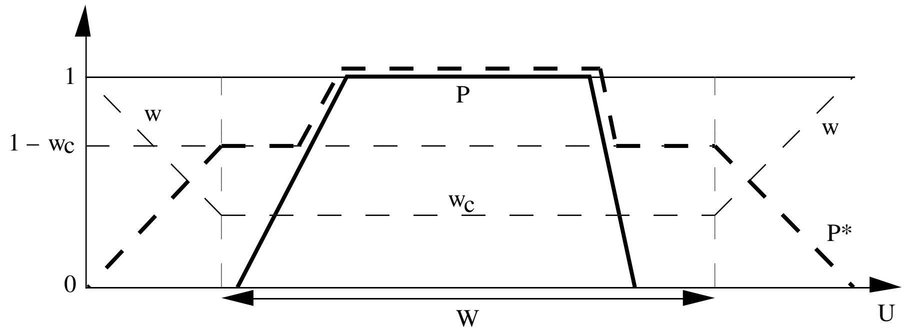

Figure 1

The weighted pattern P* is still defined by (3), i.e.

$$\mu_{P*}(u) = \max(\mu_P(u), 1 - w(u)).$$

Let $W \subseteq U$ be the subset where the weight w is constant, and equal to w_c (with $W \supseteq s(P)$). Let P' be the fuzzy set defined by

$$\mu_{P'}(u) = \begin{cases} 1 \text{ if } u \in W \\ 1 - w(u) \text{ if } u \notin W. \end{cases} \quad (5)$$

P' expresses a safeguard requirement (i.e., a minimal requirement) which should be satisfied, even if P is not. In particular, P' may be an ordinary set if $w(u) = 1, \forall u \notin W$. This can be clearly seen on the expression of P* in terms of P, P' and w_c (see Figure 1 also), namely

$$\begin{aligned}\mu_{P*}(u) &= \min(\max(\mu_P(u), 1 - w_c), \mu_{P'}(u)) \\ &= \max(\mu_P(u), \min(1 - w_c, \mu_{P'}(u)))\end{aligned} \tag{6}$$

since $\mu_{P'}(u) \geq \mu_P(u), \forall u$.

The notion of a variable weight can be interpreted in the following manner. Namely P is the fuzzy subset of values that an item component must necessarily satisfy in order to ensure that the item is compatible with the requirement (compatibility set), the complement of P' can be viewed as the fuzzy set of values which make the considered item incompatible with the requirement, regardless of other elementary requirements (rejection set). The idea of using compatibility and rejection sets, and combining the information pertaining to each ot them separately, has been also suggested by Sanchez (1991). Note that $w_c = 0$ suppresses the fuzzy set of compatible values P; only P' remain. On the contrary $w_c = 1$ inhibits the influence of P'.

When several attributes are involved, it can be established that the fuzzy sets P_i and P'_i can be separately used in the aggregation process. Then it is easy to see (since $\mu_{P'_i} > \mu_{P_i}$) that

$$\min_{i=1,n} \max(\mu_{P_i}(d), \min(1 - w_{c_i}, \mu_{P'_i}(d))) = \\ \min(\min_{i=1,n}\max(\mu_{P_i}(d), 1 - w_{c_i}), \min_{i=1,n} \mu_{P'_i}(d)). \tag{7}$$

Note that when compatibility sets are conjunctively combined, the rejection sets (the complements of the P'_i's) are disjunctively combined.

There is a different way of relaxing the requirements, namely by considering that the constraint P_i is sufficiently satisfied by d if the level of satisfaction for d reaches some threshold θ_i, i.e., $\mu_{P_i}(d) \geq \theta_i$. Then $\mu_{P_i}(d)$ will be changed into $\mu_{P*_i}(d) = 1$ and P_i will not be taken into account further in a conjunctive aggregation process (since $1 * a = a$ for any conjunction $*$). If $\mu_{P_i}(d) < \theta_i$ we may either consider that the requirement is satisfied at the level which is reached, i.e., $\mu_{P*_i}(d) = \mu_{P_i}(d)$, or in order to avoid discontinuity, make $\mu_{P*_i}(d)$ equal to the *relative* level of satisfaction $\mu_{P_i}(d) / \theta_i$ (which requires a numerical scale like [0,1] and not a simple completely ordered scale). Then the global degree of satisfaction of the requirements is

$$\min_{i=1,n} \mu_{P*_i}(d) = \min_{i=1,n} \theta_i \rightarrow \mu_{P_i}(d) \tag{8}$$

where $a \rightarrow b$ is Gödel implication in the first case ($a \rightarrow b = 1$ if $a \leq b$, $a \rightarrow b = b$ if $a > b$), and Goguen's in the second one, namely $a \rightarrow b = \min(1, b/a)$ if $a \neq 0$ and $a \rightarrow b = 1$ if $a = 0$. It contrasts with the use of Dienes implication $a \rightarrow b = \max(1-a,b)$ in (4). Here P_i is "forgotten" as soon as $\mu_{P_i}(d) \geq \theta_i$. For $\theta_i = 1$, $P*_i = P_i$. More generally, the idea of using a weighting of the form (8) dates back to Yager (1984). All the implications we considered are such that $a \rightarrow$

$b \geq b$, and thus the weighting procedure enlarges P_i. It will not be the case with Rescher-Gaines implication ($a \to b = 1$ if $a \leq b$, $a \to b = 0$ if $a > b$), with which P_i would be made crisp.

The following example can illustrate the difference of intended meanings in the two modellings of the idea of importance. Let us imagine that we look for the persons in the database who have the skills numbered I, II and III altogether. This requires a quotient operation. Quotient operations aim at finding out the sub-relation $R \div S$ of a relation R, containing sub-tuples of R which have for complements in R *all* the tuples of a relation S. In mathematical terms the quotient operation is defined by

$$R \div S = \{t, \forall\, u \in S, (t,u) \in R\} \tag{9}$$

where u is a tuple of S and t a sub-tuple of R such that (t,u) is a tuple of R. This expression can be extended to fuzzy sets (Dubois and Prade, 1996) under the form

$$\mu_{R \div S}(t) = \min_u \mu_S(u) \to \mu_R(t,u) \tag{10}$$

where $\to$ is a multiple-valued logic implication. Assume that in our example R = (<name><skill>), S = (<skill>), and the mastering of each skill is a matter of degree and that we know to which degree $\mu_R(t,u)$ each person t masters each skill u. We may look for the persons who have skills I, II, III at least to some degree $\theta_i = \mu_S(u)$ for each skill. Then we shall use (10) with Gödel (or Goguen) implication. We may also ask for the persons who master the important skills (where each skill has its own level of importance $\mu_S(u)$). Then in (10) we shall use Dienes implication $a \to b = \max(1 - a, b)$, which is such that a person who does not master at all a skill which is not fully important may still be retrieved with a non-zero evaluation. See also (Bosc et al., 1997) for considerations pertaining to implementation of fuzzy division. We may also be interested in persons mastering *most* of the skills; Yager (1991) proposes an Ordered Weighted Average operations (Yager, 1988) -based solution; see Section 2.3 for another solution.

2.2 - Conditional Requirement

A conditional requirement is a constraint which applies only if another one is satisfied. This notion will be interpreted as follows: A requirement P_j conditioned by a hard requirement P_i is imperative if P_i is satisfied and can be dropped otherwise. More generally, the level of satisfaction $\mu_{P_i}(d)$ of a fuzzy conditioning requirement P_i for an instance d is viewed as the level of priority of the conditioned requirement P_j, i.e., the greater the level of satisfaction of P_i, the greater the priority of P_j is. A conditional constraint is then naturally represented by a fuzzy set $P_i \to P_j$ such that:

$$\mu_{P_i \to P_j}(d) = \max(\mu_{P_j}(d), 1 - \mu_{P_i}(d))$$

$P_i \rightarrow P_j$ is a prioritized constraint with a variable priority.

Let us now show how to represent nested requirements with preferences, such as the ones considered by database authors (Lacroix and Lavency, 1987; Bosc and Pivert, 1993), by means of conditional prioritized requirements. Lacroix and Lavency (1987) deal with requirements of the form "P_1 should be satisfied, and among the solutions to P_1 (if any) the ones satisfying P_2 are preferred, and among those satisfying both P_1 and P_2, those satisfying P_3 are preferred, and so on", where P_1, P_2, P_3…, are hard constraints. It should be understood in the following way: satisfying P_2 if P_1 is not satisfied is of no interest; satisfying P_3 if P_2 is not satisfied is of no use even if P_1 is satisfied. Thus there is a hierarchy between the constraints. For the sake of simplicity, let us consider the case of a compound constraint P made of three nested constraints. Thus, one would like to express that P_1 should hold (with priority 1), and that if P_1 holds, P_2 holds with priority α_2, and if P_1 and P_2 hold, P_3 holds with priority α_3 (with $\alpha_3 < \alpha_2 < 1$). The constraints P_1,P_2 and P_3 are supposed to restrict the possible values of the same set of variables. Using the representation of conditional requirements presented above, this nested conditional requirement can be represented by means of the fuzzy set P*

$$\begin{aligned}\mu_{P*}(d) &= \min(\mu_{P_1}(d), \max[1 - \mu_{P_1}(d), \max(\mu_{P_2}(d), 1 - \alpha_2)],\\ &\qquad \max[1 - \min(\mu_{P_1}(d), \mu_{P_2}(d)), \max(\mu_{P_3}(d), 1 - \alpha_3)]\\ &= \min(\mu_{P_1}(d), \max(\mu_{P_2}(d), 1 - \min(\mu_{P_1}(d), \alpha_2)),\\ &\qquad \max(\mu_{P_3}(d), 1 - \min(\mu_{P_1}(d), \mu_{P_2}(d), \alpha_3)). \end{aligned} \tag{11}$$

In the above expression, it is clear that the priority level of P_2 is $\min(\mu_{P_1}(d), \alpha_2)$, i.e., is α_2 if P_1 is completely satisfied and is zero if P_1 is not at all satisfied. Similarly, the priority level of P_3 is actually $\min(\mu_{P_1}(d), \mu_{P_2}(d), \alpha_3)$. Note that it is zero if P_1 is not satisfied even if P_2 is satisfied. It is easy to check that:

$$\begin{array}{ll}
\mu_{P_1}(d) = 1 \text{ and } \mu_{P_2}(d) = 1 \text{ and } \mu_{P_3}(d) = 1 & \Rightarrow \mu_{P*}(d) = 1\\
\mu_{P_1}(d) = 1 \text{ and } \mu_{P_2}(d) = 1 \text{ and } \mu_{P_3}(d) = 0 & \Rightarrow \mu_{P*}(d) = 1 - \alpha_3\\
\mu_{P_1}(d) = 1 \text{ and } \mu_{P_2}(d) = 0 \text{ and } \mu_{P_3}(d) = 1 & \Rightarrow \mu_{P*}(d) = 1 - \alpha_2 < 1 - \alpha_3\\
\mu_{P_1}(d) = 1 \text{ and } \mu_{P_2}(d) = 0 \text{ and } \mu_{P_3}(d) = 0 & \Rightarrow \mu_{P*}(d) = 1 - \alpha_2\\
\mu_{P_1}(d) = 0 & \Rightarrow \mu_{P*}(d) = 0
\end{array}$$

Thus, as soon as P_2 is not satisfied, the satisfaction of P_3 or its violation make no difference; in both cases $\mu_{P*}(d) = 1 - \alpha_2 < 1 - \alpha_3$. P* reflects that we are completely satisfied if P_1, P_2 and P_3 are completely satisfied,we are less satisfied if P_1 and P_2 only are satisfied, and we are even less satisfied if only P_1 is satisfied.

In the preceding case an unconditioned requirement (P_1) was refined by a hierarchy of conditional prioritized requirements (P_2,P_3). A request looking for candidates such that "if they are not graduated they should have professional experience, and if they have professional experience, they should preferably have communication abilities", is an example where only conditional constraints, organized in a hierarchical way, take place. It will be represented by an expression of the form

$$\min[\max(1 - \mu_{P_1}(d), \mu_{P_2}(d)), \max(\mu_{P_3}(d), 1 - \min(\mu_{P_1}(d), \mu_{P_2}(d), \alpha)]$$

with $\mu_{P_1} = 1 - \mu_{grad.}$, $\mu_{P_2} = \mu_{prof.exp.}$ and $\mu_{P_3} = \mu_{com.ab.}$, i.e.,

$$\min[\max(\mu_{prof.exp.}(d), \mu_{grad.}(d)), \max(\mu_{com.ab.}(d), 1 - \min(1 - \mu_{grad.}(d), \mu_{prof.exp.}(d), \alpha)]$$

so that if d has professional experience and communication abilities d completely satisfies the request, as well as if d is graduated; d satisfies the request to the degree $1 - \alpha$ if d is not graduated and has professional experience only. d does not satisfy the request at all if d is neither graduated nor has professional experience (even if d has communication abilities).

2.3 - Satisfying Most of the Requirements

By "delocalizing" the weights w_i in (4), we can turn (4) into a (fuzzily) quantified conjunction, corresponding to the requirement that an item d satisfies 'at least k', or more generally 'most' requirements (rather than 'all' the requirements or more generally all the important ones). This can be done in the following way (see, e.g., Dubois et al., 1988):

i) rank-order the degree $\mu_{P_i}(d) = p_i$ *decreasingly*, where σ is a permutation of $\{1,\ldots, n\}$ in order to only consider the best satisfied constraints in the weighting process, i.e., $p_{\sigma(1)}) \geq p_{\sigma(2)} \geq \ldots \geq p_{\sigma(n)}$;

ii) let I be a fuzzy subset of the set of integers $\{0, 1, 2, \ldots, n\}$ s.t. $\mu_I(0) = 1$, $\mu_I(i) \geq \mu_I(i + 1)$. For instance, the requirement that "at least k" constraints are important will be modelled by k weights equal to 1, i.e., $w_i = \mu_I(i)$ in (4) with $\mu_I(i) = 1$ if $0 \leq i \leq k$, $\mu_I(i) = 0$ for $i \geq k + 1$;

iii) the aggregation operation is then defined by

$$\mu_{(P_1, \ldots, P_n)}(d) = \min_{i=1,n} \max(1 - \mu_I(i), p_{\sigma(i)}). \qquad (12)$$

When $\mu_I(i)=1$ for $0 \leq i \leq n$, it reduces to $\mu_{(P_1, \ldots, P_n)}(d) = p_{\sigma(n)} = \min_{i=1,n} \mu_{P_i}(d)$ as expected. When $\mu_I(1) = 1$ and $\mu_I(2) =\ldots = \mu_I(n) = 0$, it reduces to $\mu_{(P_1, \ldots, P_n)}(d) = p_{\sigma(1)} = \max_{i=1,n} \mu_{P_i}(d)$.

The expression (12) can be easily modified for accommodating relative quantifiers Q like 'most', by changing $1 - \mu_I(i + 1)$ into $\mu_Q\left(\frac{i}{n}\right)$ for $i = 0, n - 1$ and $\mu_Q(1) = 1$ where μ_Q is increasing (a required proportion of at least $\frac{k}{n}$ amounts to have k non-zero weights among n). What has been computed here is an *ordered weighted minimum* operation (OWmin), or if we prefer, the median of the set of numbers made by the $p_{\sigma(i)}$'s and the $1 - \mu_I(i)$'s. See Dubois et al. (1988). OWmin can thus be related to the idea of fuzzy cardinality. But there is no compensatory effects as opposed to Ordered Weighted Averages (Yager, 1988).

2.4 - Identifying The Correct Model for 'AND'

Queries are usually compound, and this raises the issue of finding the appropriate aggregation operation for combining the elementary degrees of matching. Even if the combination is linguistically expressed by the conjunction AND, it may correspond to very different aggregation attitudes ranging from logical to compensatory ANDs. See (Dubois and Prade, 1988) for instance.

Consider the problem of choosing a car on the basis of a catalogue of precise data concerning two objectives P_1 and P_2. Selection of an aggregation operator can be carried out by the following procedure. Three typical vehicles V_1, V_2, V_3 are presented to the decision maker for him to evaluate in terms of combined objectives P_1 and P_2 linked by means of the aggregation operator to be identified. The evaluation of each typical vehicle is an element of a 5-level scale A (completely compatible), B (pretty compatible), C (middling compatible), D (barely compatible), E (incompatible). The typical vehicles are chosen so as to enable discrimination between aggregation operators in a given list. The compatibility of each typical vehicle with each of the objectives to be combined is supposed to be known. In particular, the vehicles are supposed to be chosen so that

- V1 is incompatible (score E) with P1 but completely compatible (score A) with P2;
- V2 has medium compatibility (score C) with each of P1 and P2;
- V3 has medium compatibility (score C) with P1 and is completely compatible (score A) with P2.

The aggregation operator h between μ_{P_1} and μ_{P_2} is then approximated by a function $\hat{h}$ from $\{A, C, E\}^2$ into {A, B, C, D, E} which is increasing in the wide sense in each place, symmetric, and such that $\hat{h}$ (A,A) = A and $\hat{h}$ (E,E) = E. The decision maker thus provides the three values $\hat{h}$(E,A), $\hat{h}$(C,C), and $\hat{h}$(C,A). Each triplet of replies corresponds to a standard aggregation operator as indicated in Table 1. We are here constructing a filter for a multivalued rather than a binary logic. The function that the filter realizes is supposed to represent the behavior of the decision maker when faced with the various objects that the computer can present to him. The class of available operations is viewed as a collection of "standard functions". Table 1 is far from exhaustive and covers only some of the possible responses by the decision maker. The full list of possible responses contains 50 triples corresponding to the following constraints: (1) $\hat{h}$(C,A) ≥ max($\hat{h}$(E,A), $\hat{h}$(C,C)); (2) $\hat{h}$ is symmetric; and (3) $\hat{h}$(C,A) ≥ C (meeting objective P_2 completely cannot decrease the global level of satisfaction below the level of satisfaction of objective P_1). Note also that the function $\hat{h}$ is not completely defined if only three typical vehicles are used. Complete specification of $\hat{h}$ requires knowledge of the value of $\hat{h}$(E,C) as well as the three values provided by the decision maker. The extra information would enable a finer discrimination to be made, but would raise the number of possible responses to 93.

	Vehicle type			
	V_1	V_2	V_3	
Objectives P_1	E	C	C	Selected operators
Objectives P_2	A	C	A	with A = 1, B = .75, C = .5, D = .25, E = 0.
Examples of	E	E	C	max(0, x + y – 1)
possible responses by	E	D	C	$x \cdot y$
the decision maker	E	C	C	min(x,y)
	E	C	B	$(x \cdot y)^{1/2}$, 2xy / (x + y)
	D	C	C	med(x, y, 1/4)
	C	C	C	med(x, y, 1/2), min(x,y) / (1 – |x – y|)
	C	C	B	(x + y)/2, $\frac{x + y - xy}{1 + x + y - 2xy}$, max(x,y) / (1 + |x – y|)
	C	C	A	$\frac{xy}{1 - x - y + 2xy}$
	B	C	B	med(x, y, 3/4)
	A	C	A	max(x,y), $1 - [(1 - x)(1 - y)]^{1/2}$
	A	B	A	x + y – xy
	A	A	A	min(1, x + y)
				med = median

Table 1: Selection of aggregation operations

The four classes of operators, namely, conjunctions, disjunctions, averages, and symmetric sums (see Dubois and Prade, 1988), only cover some of the 50 possible triples. Nevertheless, many of these triples correspond to minor modifications of standard operators (e.g., (D, C, C) is very close to (E, C, C), which is represented by the min operator).

But few triples, such as (C, E, C), fall outside these four classes. However, a triple such as (C, E, C), can be reached by a "weighted" combination, in this case: min(max(e,f), max(min(e,f), 1 – (e $\Rightarrow$ C), 1 – (f $\Rightarrow$ C)))) with e = A $\rightarrow$ x, f = A $\rightarrow$ y where $\Rightarrow$ is Gödel implication and $\rightarrow$ is Rescher-Gaines implication, 1 - C = C. Indeed this triple corresponds to the following aggregation attitude: the result of the combination is E except if u = A or v = A, and in this latter case it is A if u = v = A, and it is C otherwise (assuming $\hat{h}$(E,C) = E). Indeed, e (resp. f) is equal to A = 1 if x (resp. y) reaches A, and e (resp. f) is 0 otherwise; at least one of these two conditions should hold, and either both hold or the result is C (1 – (e $\Rightarrow$ C) is equal to 0 if e = 0 and C if e = 1).

See Andrès (1989) for providing a more complete table and applying it to fuzzy query evaluation.

3 - Conclusion

This paper has emphasized the modelling capabilities of the fuzzy set framework for representing flexible queries. Different types of flexibility in queries have been considered: introducing tolerance, assigning various kinds of weights of importance, expressing conditional requirements, allowing for fuzzy quantifiers, identifying the correct type of 'AND' connectives in a compound query. A proper understanding of these capabilities is required for designing an interface which builds an accurate representation of the queries.

References

Andrès V. (1989) Filtrage sémantique dans une base de données imprécises et incertaines: Un système souple autorisant la formulation de requêtes composites pondérées. Dissertation, Université P. Sabatier, Toulouse, France.

Bosc P., Dubois D., Pivert O., Prade H. (1997) Flexible queries in relational databases —The example of the division operator—. Theoretical Computer Science, 171, 281-302.

Bosc P., Dubois D., Prade H. (1996) Fuzzy functional dependencies and redundancy elimination. In:Tech. Report IRIT/96-10-R, IRIT, Univ. P. Sabatier, Toulouse, France. To appear in J. Amer. Soc. Infor. Syst.

Bosc P., Kacprzyk J. (Eds.) (1995) Fuzziness in Database Management Systems. Physica-Verlag, Heidelberg.

Bosc P., Pivert O. (1992) Some approaches for relational databases flexible querying. J. of Intelligent Information Systems, 1, 323-354.

Bosc P., Pivert O. (1993) An approach for a hierarchical aggregation of fuzzy predicates. Proc. 2nd IEEE Int. Conf. Fuzzy Systems (FUZZ-IEEE'93), San Francisco, 1231-1236.

Bosc P., Pivert O. (1995) SQLf: A relational database language for fuzzy querying. IEEE Trans. on Fuzzy Systems, 3(1), 1-17.

Bosc P., Prade H. (1997) An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases. In: Uncertainty Management in Information Systems: From Needs to Solutions (A. Motro, Ph. Smets, eds.), Kluwer Academic Publ., Chapter 10, 285-324.

Buckles B.P., Petry F.E. (1982) A fuzzy representation of data for relational databases. Fuzzy Sets and Systems, 5, 213-226.

Cayrol M., Farreny H., Prade H. (1982) Fuzzy pattern matching. Kybernetes, 11, 103-116.

Chen G.Q., Kerre E.E., Vandenbulcke J. (1994) A computational algorithm for the FFD transitive closure and a complete axiomatization of fuzzy functional dependencies. J. of Intelligent Systems, 9(5), 421-440.

Cubero J.C., Vila M.A. (1994) A new definition of fuzzy functional dependency in fuzzy relational databases. J. of Intelligent Systems, 9(5), 441-448.

Dubois D., Prade H. (1988) Possibility Theory — An Approach to Computerized Processing of Uncertainty. Plenum Press, New York.

Dubois D., Prade H. (1990) Measuring properties of fuzzy sets: A general technique and its use in fuzzy query evaluation. Fuzzy Sets and Systems, 38, 137-152.

Dubois D., Prade H. (1996) Semantics of quotient operators in fuzzy relational databases. Fuzzy Sets and Systems, 78, 89-93.

Dubois D., Prade H., Testemale C. (1988) Weighted fuzzy pattern matching. Fuzzy Sets and Systems, 28, 313-331.

Kacprzyk J., Ziolkowski A. (1986) Data base queries with fuzzy linguistic quantifiers. IEEE Trans. on Systems, Man and Cybernetics, 16(3), 474-478.

Lacroix M., Lavency P. (1987) Preferences: Putting more knowledge into queries. Proc. of the 13rd Inter. Conf. on Very Large Data Bases, Brighton, UK, 217-225.

Petry F.E. (1996) Fuzzy Databases: Principles and Applications. Kluwer Acad. Pub., Dord.

Prade H., Testemale C. (1984) Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. Information Sciences, 34, 115-143.

Raju K.V.S.V.N., Majumdar A.K. (1988) Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. ACM Trans. on Database Systems, 13(2), 129-166.

Sanchez E. (1991) Fuzzy logic and neural networks in Artificial Intelligence and Pattern Recognition. SPIE, Vol. 1569, Stochastic and Neural Methods in Signal Processing, Image Processing and Computer Vision, 474-483.

Tahani V. (1977) A conceptual framework for fuzzy query processing — A step toward very intelligent database systems. Information Processing Management, 13, 289-303.

Umano M. (1982) FREEDOM-0: A fuzzy database system. In: Fuzzy Information and Decision Processes (M.M. Gupta, E. Sanchez, eds.), North-Holland, 339-347.

Vandenberghe R., Van Schooten A., De Caluwe R., Kerre E.E. (1989) Some practical aspects of fuzzy database techniques: An example, Information Systems, 14, 465-472.

Wu X.D., Mahlén P. (1995) Fuzzy interpretation of induction results. Proc. of the Inter. Conf. on Knowledge Discovery & Data Mining (U.M. Fayyad, R. Uthurusamy, eds.), Montréal, Canada, Aug. 20-21, 325-330.

Yager R.R. (1984) General multiple objective decision making and linguistically quantified statements. Int. J. of Man-Machine Studies, 21, 389-400.

Yager R.R. (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Trans. on Systems, Man and Cybernetics, 18, 183-190.

Yager R.R. (1991) Fuzzy quotient operators for fuzzy relational data bases. In: Fuzzy Engineering toward Human Friendly Systems, Vol. 1 (Proc. Inter. Fuzzy Engineering Symp. (IFES'91), Yokohama, Japan, Nov. 13-15, 1991) (T. Terano, M. Sugeno, M. Mukaidono, K. Shigemasu, eds.), Available from IOS Press, Amsterdam, 289-296.

Yager R.R. (1996) Database discovery using fuzzy sets. Tech. Report #MII-1601, Machine Intelligence Institute, Iona College, New Rochelle, NY.

Zadeh L.A. (1965) Fuzzy sets. Information and Control, 8, 338-353.

Zemankova M., Kandel A. (1984) Fuzzy Relational Databases — A Key to Expert Systems. Interdisciplinary Systems Research, Verlag TÜV, Rheinland.