Storing Fuzzy Knowledge and Fuzzy Metaknowledge in Relational Systems

José Galindo
Dpto. Lenguajes y Ciencias de la Computación, Univ. de Málaga, Spain

Angélica Urrutia
Dpto. de Computación e Informática, Univ.Católica del Maule, Chile

Mario Piattini
Escuela Superior de Informática, Univ. de Castilla la Mancha, Spain

Abstract: - In this article we present how to implement fuzzy databases based on the relational model. This approach includes many fuzzy attribute types, which can express the most of fuzzy knowledge types. These fuzzy attribute types include imprecise attributes, fuzzy attributes associated to one or more attributes or with an independent meaning. In order to represent such fuzzy information we must study two aspects of fuzzy information: how to represent fuzzy data and how to represent fuzzy metaknowledge data. This second information is very important and it must be considered in any fuzzy database. This article study the fuzzy metaknowledge data for any fuzzy attribute and how to represent both in a relational database.

Keywords: Fuzzy relational databases, Fuzzy attributes, Fuzzy degrees, Fuzzy metaknowledge, Representation of Fuzzy Knowledge.

1. Introduction

The relational model was developed by E.F. Codd of IBM and published in 1970. This model is the most used at present. In a theoric level, there exists many Fuzzy Relational Database models that, based on the relational model, they extend it in order to allow storing and/or treating vague and uncertain information [9].

On the other hand, the FuzzyEER model [5][6][10][11] is an extension of the EER model to create a models with fuzzy semantics and notations. This extension is a good eclectic synthesis among the different models and it provides new and useful definitions: fuzzy attributes, fuzzy entities, fuzzy relationships, fuzzy specializations…

The next step is to translate the FuzzyEER definitions to a DBMS. This will allow us to represent the database fuzzy knowledge. Actually the 90% of dabatases are relational. Our target is to present this extension as simple and useful as possible. Then, we have choose the relational model and we have extended the FIRST definitions [3][7][8], which has been used in some applications [1][2].

The next section define the fuzzy attributes included in FuzzyEER model. After, we define how to represent fuzzy data and how to represent fuzzy metaknowledge data. Finally, concluding remarks and future developments are discussed.

2. Fuzzy Attributes

In order to model fuzzy attributes we distinguish between two classes of fuzzy attributes: Fuzzy attributes whose fuzzy values are fuzzy sets and fuzzy attributes whose values are fuzzy degrees.

2.1. Fuzzy Sets as Fuzzy Values

These fuzzy attributes may be classified in four types, based on the definitions of [3][7][8]. This classification is performed taking into account the type of referential or underlying domain. In all of them the values Unknown, Undefined, and Null are included:

- **Type 1**: These are attributes with "precise data", classic or crisp (traditional, with no imprecision). However, they can have linguistic labels defined over them and we can use them in fuzzy queries. This type of attribute is represented in the same way as precise data, but can be transformed or manipulated using fuzzy conditions. This type is useful for extending traditional databases allowing fuzzy queries to be made about classic data. For example, enquiries of the kind "Give me employees that earn a lot more than the minimum salary".
- **Type 2**: These are attributes that gather "*imprecise data over an ordered referential*". These attributes admit both crisp and fuzzy data, in the form of possibility distributions over an underlying ordered dominion (fuzzy sets). It is an extension of the Type 1 that does, now, allow the storage of imprecise information, such as "he is approximately 2 metres tall". For the sake of simplicity the most complex of these fuzzy sets are supposed to be a trapezoidal function (Fig. 1).
- **Type 3**: They are attributes over "*data of discreet non-ordered dominion with analogy*". In these attributes some labels are defined ("blond", "ginger", "brown",) that are scalars with a similarity (or proximity) relationship defined over them, so that this relationship indicates to what extent each pair of labels resemble each other. They also allow possibility distributions (or fuzzy sets) over this dominion, like for example, the value (1/dark, 0.4/brown) which expresses that a certain person is more likely to be dark than brown-haired. Note that underlying domain of these fuzzy sets are the set of labels and this set is non-ordered.
- **Type 4**: These are attributes proposed in this paper and they are defined in the same way as Type 3 attributes, without it being necessary for a similarity relationship to exist between the labels. In this case, we suppose that we do not need the similarity relationship (or it does not exist).

2.2. Fuzzy Degrees as Fuzzy Values

The domain of these degrees can be found in the interval [0,1], although other values are also permitted, such as a possibility distribution (usually over this unit interval). In order to keep it simple, we will only use degrees in the interval [0,1], because the other option offers no great advantages.

The meaning of these degrees is varied and depends on their use. The processing of the data will be different depending on the meaning. The most important possible meanings of the degrees used by some authors are [3][4]: Fulfillment degree, Uncertainty degree, Possibility degree and Importance degree. Of course, we can define and use other meanings.

The ways of using these fuzzy degrees are classified in two families: Associated and non-associated degrees.

Associated degrees are associated to a specific value to which they incorporate imprecision. These degrees may be associated to different concepts [4]:

- **Degree in each value of an attribute (**we will call it as **Type 5):** Some attributes may have a fuzzy degree associated to them. This implies that each value of this attribute (in every tuple or instance) has an associated degree, that measures the level of fuzziness of that value. In order to interpret it, we need to know the meaning of the degree and the meaning of the associated attribute.
- **Degree in a set of values of different attributes (Type 6):** Here, the degree is associated to some attributes. Whilst this is an unusual case, it can sometimes be very useful. It joins the fuzziness of some attributes in only one degree.
- **Degree in the whole instance of the relation (Type 7):** This degree is associated to the whole tuple of the relation and not exclusively to the value of a specific attribute of the tuple (or instance). Usually, it can represent something like the "membership degree" of this tuple (or instance) to the table (or entity) of the database.

Non-associated degrees (Type 8): There are cases in which the imprecise information which we wish to express can be represented by using only the degree, without associating this degree to another specific value or values. For example, the dangerousness of a medicine may be expressed by a fuzzy degree.

In this paper we do not aim to demonstrate the usefulness of these degrees and their different meanings. Several authors who have used these degrees have already done so.

3. Representation of Fuzzy Attributes

This representation is different according to the fuzzy attribute Type. Fuzzy attributes Type 1 are represented as usual attributes, because they do not allow fuzzy values. Fuzzy attributes Type 2 need five classic attributes: One stores the kind of value (Table 1) and the others four store the crisp values representing the fuzzy value. Note, in Fig. 1 and Table 1, that trapezoidal fuzzy values need the others four values. An approximate value (approximately *d*, *d*±margin) is represented with a triangular function centered in *d* (degree 1) and with degree 0 in *d*–margin and *d*+margin, where value margin depend on the context (Fig. 1 with b=c and b–a=d–c=margin).

Fuzzy attributes Type 3 need a variable number of attributes: One stores the kind of value (Table 2). Note, in Table 2, that number 3 need only two values, but number 4 need 2*n* values, where *n* is the maximum length for possibility distributions for each fuzzy attribute. Value *n* must be defined for each fuzzy attributes Type 3, and it is stored in the FMB (see following section).

Fuzzy attributes Type 4 are represented just like Type 3. The different between then is shown in the next section. Fuzzy degrees (Types 5, 6, 7 and 8) are represented using a classic numeric attribute, because its domain is the interval [0,1].

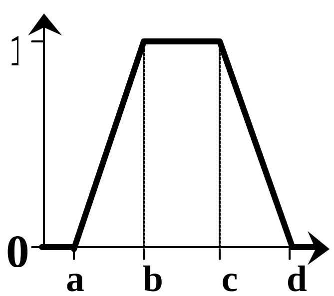

Figure 1: Trapezoidal function.

Number	Kind of values
0, 1, 2	**UNKNOWN, UNDEFINED, NULL**
3	**CRISP:** *d*
4	**LABEL:** label_identifier
5	**INTERVAL:** [*n*,*m*]
6	**APPROXIMATE VALUE:** *d*
7	**TRAPEZOIDAL:** [a,b,c,d]

Table 1: Kind of values of fuzzy attributes Type 2.

Number	Kind of values
0, 1, 2	**UNKNOWN, UNDEFINED, NULL**
3	**SIMPLE:** Degree/Label
4	**POSSIBILITY DISTRIBUTION:** $\text{Degree}_1/\text{label}_1 + \ldots + \text{Degree}_n/\text{Label}_n$

Table 2: Kind of values of fuzzy attributes Type 3 and 4.

4. Representation of Fuzzy Metaknowledge Data: The FMB

Fuzzy metaknowledge data are the necessary knowledge about the fuzzy database (fuzzy attributes specially). This information is stored in relational format in the so-called FMB (Fuzzy Metaknowledge Base). First, we define the information stored in the FMB, and then we explain the structure of it (i.e., the relations in the FMB).

4.1. Information in the FMB

The FMB include the following information:

1. Attributes with fuzzy capabilities: fuzzy attributesand fuzzy degress (Types 1 to 8).

2. The metaknowledge of each attribute is different according to its type:

   - Types 1 and 2: These fuzzy attributes store in the FMB the definition (fuzzy set) of each

linguistic label, the "margin" for approximate values, and the minimum distance to consider two values as very separated (so-called "much"). This last value is used in comparisons like "much greater than".
- Types 3 and 4: Value *n* (explained above), name of linguistic labels and, only for Type 3, the similarity relationship between whatever two labels.
- Types 5 and 6: Meaning of the degree and attribute (Type 5) or attributes (Type 6) to which the degree is associated.
- Types 7 and 8: Meaning of the degree.

3. Other objects: These objects include fuzzy qualifiers (associated to an attribute and used to set the threshold in queries) and fuzzy quantifiers (associated to a relation or to an attribute). Fuzzy quantifiers are used in queries (for example "Give me employees who belong to *most* of projects"), and in fuzzy constraints (for example "An employee must work in *many* projects").

If two fuzzy attributes (Types 1, 2, 3 or 4) need the same definitions we can register these two attributes as compatibles. This simplify data in the FMB.

4.2. Relations in the FMB

Fig. 2 shows the FMB relations (or tables), its attributes, its primary keys (underlined) and its foreign keys (with arrows). We use OBJ# as the table identifier, and COL# as the column or attribute identifier (just like Oracle). We cannot explain all attributes of all FMB relations for lack of space. Then we only try to give an idea about the usefulness of each relation:

- **FUZZY_COL_LIST:** It describes fuzzy attributes identified by (OBJ#,COL#). F_TYPE set the fuzzy type (from 1 to 8). LEN is the value *n*. CODE_SIG indicates the degree meaning when F_TYPE$\in$ [5,8].

- **FUZZY_DEGREE_SIG:** It stores all the degree meanings of our database.

- **FUZZY_OBJECT_LIST:** This relation contains declarations of fuzzy objetcs related with fuzzy attributes. These fuzzy objects are: linguistic labels, qualifiers and fuzzy quantifiers.

- **FUZZY_LABEL_DEF:** It defines the linguistic labels using trapezoidal functions (Fig. 1).

- **FUZZY_APPROX_MUCH:** Values "margin" and "much" for Types 1 and 2.

- **FUZZY_NEARNESS_DEF:** Similarity relationships for Type 3.

- **FUZZY_COMPATIBLE_COL:** Compatible fuzzy attributes, i.e., attributes which use the same linguistic labels.

- **FUZZY_QUALIFIERS_DEF:** It defines fuzzy qualifiers.

- **FUZZY_DEGREE_COLS:** This relation sets the attributes (or columns) associated to fuzzy degrees (only for Type 5 and 6). Note that a Type 5 degree has only one associated attribute, a Type 6 degree has some attributes and an attribute may have many degrees associated to it (but these degrees must be Type 5 or 6). Of course Type 7 and 8 degrees do not use this table.

- **FUZZY_ER_LIST:** Using FuzzyEER words, this relation stores fuzzy entities and fuzzy relationships. DEGREE_TYPE take 'M' for fuzzy entities, 'C' for fuzzy entities with degrees computed automaticaly, 'E' and 'I' for fuzzy weak entities (dependency on existence or dependency on identification) and, finally, 'R' for fuzzy relationships represented by the table OBJ#.

- **FUZZY_TABLE_QUANTIFIERS:** Definition of quantifiers associated to a relation or table (not to a column). These quantifiers are used in fuzzy constraints and they may be absolute or relative.

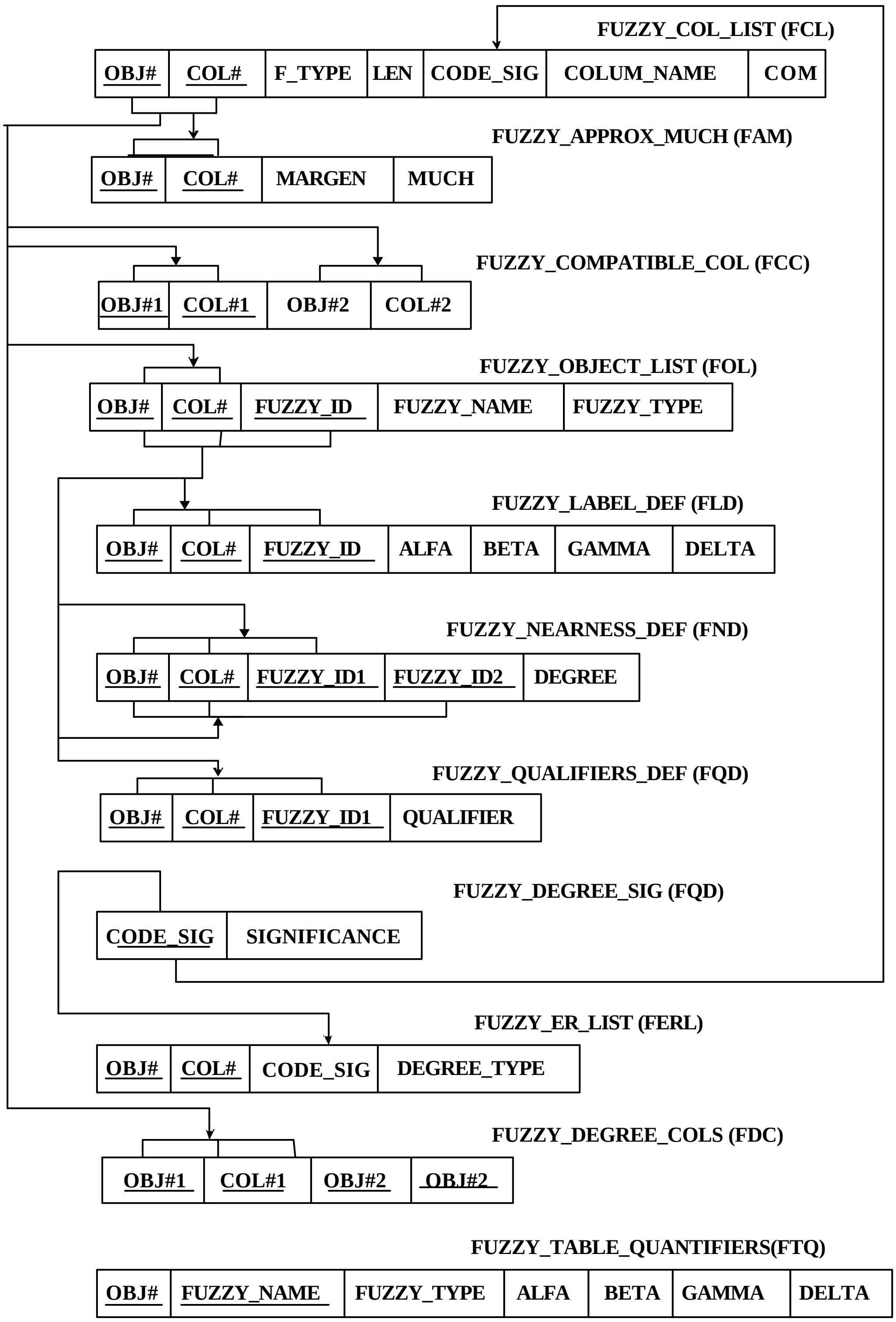

Figure 2: FMB tables in FIRST-2.

5. Conclusions and Future Lines

This article presents how to store the fuzzy knowledge of fuzzy databases in a classic relational database. This allow us to implement fuzzy databases modeled with the FuzzyEER model [5][6][10][11].

It should be stressed the fuzzy attributes types, which can express the most of fuzzy knowledge types. This research studies how to represent fuzzy data, the necessary metaknowledge about these fuzzy data and how to represent this fuzzy metaknowledge data. This second information is very important and must be considered in any fuzzy database.

Actually, we have developed fuzzy databases with some of these characteristics [3]. Besides, FSQL (Fuzzy SQL) language may be used in those databases [2]. The future is extend FSQL language, in order to treat with all fuzzy attribute types presented here.

References:

[1] Blanco I., Cubero J.C., Pons O., Vila M.A. (2000): "An Implementation for Fuzzy Deductive Relational Databases". In "Recent Issues on Fuzzy Databases", Ed. G. Bordogna and G. Pasi. Physica-Verlag (Studies in Fuzziness and Soft Computing), pp. 183-207.

[2] Galindo J., Medina M., Pons O., Cubero J. C., "A Server for Fuzzy SQL Queries". In "Flexible Query Answering Systems", Eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164-174. Ed. Springer, 1998.

[3] Galindo J., "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales". Tesis Ph. Doctoral Universidad de Granada, España, 1999 (www.lcc.uma.es).

[4] Galindo J., Medina J.M., Cubero J.C., García M.T., "Relaxing the Universal Quantifier of the Division in Fuzzy Relational Databases". International Journal of Intelligent Systems, 16(6), pp. 713-742, 2001.

[5] Galindo J., Urrutia A., Carrasco R., Piattini M., "Fuzzy Constraints using the Enhanced Entity-Relationship Model". Proceedings published by IEEE-CS Press of the XXI Int. Conf. of the Chilean Computer Science Society (SCCC 2001), pp. 86-94. Punta Arenas (Chile), 2001.

[6] Galindo J., Urrutia A., Carrasco R.A., Piattini M., "Relaxing Constraints in Enhanced Entity-Relationship Models using Fuzzy Quantifiers". To appear in IEEE Transactions on Fuzzy Systems, 2004/2005.

[7] Medina J.M., "Bases de datos Relacionales Difusas: Modelo Teórico y Aspectos de su Implementación", Ph. Tesis Doctoral, Universidad de Granada, España, 1994 (decsai.ugr.es).

[8] Medina J.M., Pons O., Vila A., "FIRST. A Fuzzy Interface for Relational SysTems". VI International Fuzzy Systems Association World Congress (IFSA 1995). Sao Paulo (Brasil), 1995.

[9] Petry F.E., "Fuzzy Databases: Principles and Applications". *International Series in Intelligent Technologies*. Ed. H.J. Zimmermann. Kluwer Academic Publ. (KAP), 1996.

[10] Urrutia A., Galindo J., Piattini M., "Modeling Data Using Fuzzy Attributes". Proceedings published by IEEE Computer Society Press of the XXII International Conference of the Chilean Computer Science Society (SCCC 2002), pp. 117-123. Copiapo (Chile), 2002. ISBN: O-7695-1867-2.

[11] Urrutia A., "Definición de un Modelo Conceptual para Bases de Datos Difusas". Ph. Doctoral Thesis, University of Castilla la Mancha (Spain), 2003.