Relaxing Constraints in Enhanced Entity-Relationship Models Using Fuzzy Quantifiers

José Galindo, Angélica Urrutia, Ramón A. Carrasco, and Mario Piattini

***Abstract*—While various articles about fuzzy entity relationship (ER) and enhanced entity relationship (EER) models have recently been published, not all examine how the constraints expressed in the model may be relaxed. In this paper, our aim is to relax the constraints which can be expressed in a conceptual model using the modeling tool, so that these constraints can be made more flexible. We will also study new constraints that are not considered in classic EER models. We use the fuzzy quantifiers which have been widely studied in the context of fuzzy sets and fuzzy query systems for databases. In addition, we shall examine the representation of these constraints in an EER model and their practical repercussions. The following constraints are studied: the fuzzy participation constraint, the fuzzy cardinality constraint, the fuzzy completeness constraint to represent classes and subclasses, the fuzzy cardinality constraint on overlapping specializations, fuzzy disjoint and fuzzy overlapping constraints on specializations, fuzzy attribute-defined specializations, fuzzy constraints in union types or categories and fuzzy constraints in shared subclasses. We shall also demonstrate how fuzzy (min, max) notation can substitute the fuzzy cardinality constraint but not the fuzzy participation constraint. All these fuzzy constraints have a new meaning, they offer greater expressiveness in conceptual design, and are included in the so-called fuzzy EER model.**

***Index Terms*—Conceptual database design, extended (or enhanced) entity-relationship model (EER), fuzzy conceptual modeling, fuzzy constraints, fuzzy databases, fuzzy quantifiers.**

I. Introduction

CONCEPTUAL modeling or design is a fundamental phase in the design of any database [14]. In this phase of conceptual design, the aim is to obtain the so-called conceptual schema, which is a concise description of the data required by users including detailed descriptions of the types of entities involved, the interrelationships existing between them and also some important constraints in these relationships. The conceptual schema is represented using a high-level data model which allows all this information to be expressed without including implementation details, and in such a way that it is easy to understand even for nontechnical users. In fact, one of the great advantages of using conceptual schemas is that they facilitate communication with this type of user.

The enhanced entity relationship (EER) model [9], [14], [24] is an extension of the entity relationship (ER) model [6]. This study is based on the EER model published in [9] and [14], which is one of the most modern, versatile, and complete versions.

The management of uncertainty in database systems is a very important problem [36] as the information is often vague. Fuzzy databases [17], [34], [38] have also been widely studied, with little attention being paid to the problem of conceptual modeling [5]. This does not mean that there are no publications, however, but that they are sparse and with no standard. Therefore, there have also been advances in modeling uncertainty in database systems [2], [8], [27], [35], [49] including object-oriented database models [1], [3], [22], [23], [49].

At the same time, the extension of the ER model for the treatment of fuzzy data (with vagueness) has been studied in various publications [4], [5], [7], [8], [28], [32], [40], [45], [47], [53], but none of these refer to the possibility of expressing constraints flexibly by using the tools offered by fuzzy sets theory. An overview of some of these fuzzy data models is published in [27].

Zvieli and Chen [53] allow fuzzy attributes in entities and relationships and they introduced three levels of fuzziness in the ER model.

1) At the **first level**, entity sets, relationships and attribute sets may be fuzzy, i.e., they have a membership degree to the model. For example, the fuzzy entity Radio may have a 0.7 importance degree as an integrating part of a car.
2) The **second level** is related to the fuzzy occurrences of entities and relationships. For example, an entity Young_Employees must be fuzzy, because its instances, its employees, belong to the entity with different membership degrees.
3) The **third level** concerns the fuzzy values of attributes of special entities and relationships. For example, the attribute quality of a basketball player may be fuzzy.

Although the first level may be useful, we must eventually decide whether such an entity, relationship or attribute will appear in the implementation or not. The second level is also useful, but it is important to consider different degree meanings (membership degree, importance degree, fulfillment degree, etc.). A list of authors using different meanings is included in [19]. The third level is useful, but it is similar to writing the data type of some attributes, because fuzzy values belong to fuzzy data types.

Chaudhry *et al.* [4], [5] propose a method for designing fuzzy relational databases (FRDBs) following the extension of the ER model of [53] taking special interest in converting crisp databases into fuzzy ones. The way to do this is to define n linguistic labels as n fuzzy sets over the universe of an attribute. Each tuple in the

Manuscript received January 2, 2002; revised November 11, 2002, October 2, 2003, and March 3, 2004. This work was supported by the RITOS2 network and the MCYT project TIC2002-00480.

J. Galindo is with the Dipartimento de Lenguajes y Ciencias de la Computación, Universidad de Málaga, Málaga 29071, Spain (e-mail: jgg@lcc.uma.es).

A. Urrutia is with the Departamento de Computación e Informática, Universidad Católica del Maule, Talca, Chile (e-mail: aurrutia@spock.ucm.cl).

R. A. Carrasco is with the Department of Knowledge Management, Caja-Granada, Granada 18006, Spain (e-mail: carrasco@caja-granada.es).

M. Piattini is with the Escuela Superior de Informática, Universidad de Castilla la Mancha, Ciudad Real 13071, Spain (e-mail: mpiattin@inf-cr.uclm.es).

Digital Object Identifier 10.1109/TFUZZ.2004.836088

crisp entity is then transformed into n fuzzy tuples in a new entity (or n values in the same tuple). Each fuzzy tuple (or value) does not store the crisp value but a linguistic label and a grade of membership giving the degree to which the corresponding crisp entity belongs to the new entity. Finally, the crisp entity and the new fuzzy entity are mapped to separate tables. Their ER model includes fuzzy relationships as relationships with at least one attribute, namely, the membership grade. They propose FERM, a design methodology for mapping a fuzzy ER data model to a crisp relational database in four steps (constructing a fuzzy ER data model, transforming it into relational tables, normalization and ensuring a correct interpretation of the fuzzy relational operators). They also present the application of FERM in order to build a prototype of a fuzzy database for a discrete control system for a semiconductor manufacturing process.

In [7], [8], and [28], Chen and Kerre introduce the fuzzy extension of several major EER concepts (superclass, subclass, generalization, specialization, category and shared subclass) without including graphical representations. The basic idea is that if E_1 is a superclass of E_2 and $e \in E_2$, then $\mu_{E_1}(e) \leq \mu_{E_2}(e)$, where μ_{E_1} and μ_{E_2} are the membership functions to E_1 and E_2, respectively. They discuss three kinds of constraints with respect to fuzzy relationships but they do not study fuzzy constraints. 1) The inheritance constraint means that a subclass instance inherits all relationships instances in which it has participated as a superclass entity. 2) The total participation constraint for entity E is defined when for any instance in $E, \exists \alpha_i$ such that $\alpha_i > 0$, where α_i is one membership degree in the fuzzy relationship. 3) The cardinality constraints 1:1, 1:N and N:M are also studied with fuzzy relationships.

In [40], Ruspini proposes an extension of the ER model with fuzzy values in the attributes, and a truth value can be associated with each relationship instance. In addition, some special relationships such as same-object, subset-of, member-of, ... are also introduced. Vandenberghe [45] applied Zadeh's extension principle to calculate the truth value of propositions. For each proposition, a possibility distribution is defined on the doubleton true, false of the classical truth values. In this way, the concepts such as entity, relationship and attribute as well as subclass, superclass, category, generalization and specialization, ... have been extended. The proposal of Vert *et al.* [47] is based on the notation used by Oracle and uses the fuzzy sets theory to treat data sets as a collection of fuzzy objects, applying the result to the area of geospatial information systems (GISs).

Finally, Ma *et al.* [32] work with the three levels of Zvieli and Chen [53] and they introduce a fuzzy extended entity-relationship (FEER) model to cope with imperfect as well as complex objects in the real world at a conceptual level. However, their definitions (of generalization, specialization, category and aggregation) impose very restrictive conditions. They also provided an approach to mapping a FEER model to a fuzzy object-oriented database schema.

Another line of work in fuzzy conceptual data modeling (without using the ER model) is reported in [15], using a graph-oriented schema for modeling a fuzzy database. Fuzziness is handled by defining various links between records of the value database (actual data values) and the explanatory database (semantic interpretation of fuzzy attributes, symmetries, ...). Extensions carried out to allow modeling imprecision in semantic data models are also described in [39], focusing on exploring the potential of semantic data models to allow fuzziness to be represented.

The ExIFO conceptual model presented in [50] allows imprecision and uncertainty in database models, based on the IFO conceptual model. They use fuzzy-valued attributes, incomplete-valued attributes and null-valued attributes. In the first case, the true data may belong to a specific set or subset of values, for example the domain of this attribute may be a set of colors {red, orange, yellow, blue} or a subset {orange, yellow} where there is a similarity relation between the colors. In the second case, the true data value is not known, for example, the domain of this attribute may be a set of years between 1990 and 1992. In the third case, the true data value is available, but is not expressed precisely, for example the domain of this attribute may be the existence or not of a telephone number. For each of these attribute types, there is a formal definition and a graphical representation. In this study, the authors introduce a high-level primitives to model fuzzy entity type whose semantics are related to each other with logic operators OR, XOR, or AND. An example involving an employee-vehicle scheme is used to illustrate the aggregation and composition of fuzzy entity types. The main contribution of this approach is the use of an extended NF^2 relation (nonfirst normal form) to transform a conceptual design into a logical design. Consequently, the strategy is to analyze the attributes that compose the conceptual model in order to establish an NF^2 model.

In another line, a set of constructs for capturing certain types of semantic integrity constraints is presented in [10], based on the specific types of logic propositions which exist on a collection of relationships between a given entity set and the entity to which it is associated through these relationships. For example, let E be an entity with two relationships (P and Q). Using classical logic, we can then apply the following constraint based on the implication function: If an instance $e \in E$ uses relation P, then e uses relation Q. It can be observed that these constraints only need classical logic and that some of their cases are also solved by the EER model.

Not all of these articles study how to relax the constraints expressed in the ER/EER model so that they can be made more flexible, because the constraints of the traditional model are either too restrictive or too permissive. This article presents an extension and an improvement of [20], [21]. Furthermore, we propose the graphic representation of these constraints in an ER/EER model and we study their practical repercussions.

First, we will present a very brief summary of the ER/EER model, focusing primarily on its constraints. We will then summarize the basic concepts of fuzzy logic, paying particular attention to fuzzy quantifiers. Next, we will study each constraint separately and look at how it can be treated in a fuzzy way. Finally, we outline some conclusions and suggest some future lines of research.

II. ER/EER Model: Constraints

The ER model graphically represents data as entities, relationships and attributes. Entities are objects which exist in the real world, and are represented in the model by rectangles. Relationships are concepts which relate different entities to each other. Relationships are represented using diamond shapes. Both entities and relationships can have different attributes which identify or characterize them.

The EER model allows us to extend the description of the entities with new types (superclasses, subclasses and categories). A subclass is a specialization of a superclass, so that each member of a subclass must be a member of the superclass. A superclass is a generalization of one or several subclasses. A specialization is represented with a circle to which the superclass and all its subclasses are connected. Subclasses are marked with the inclusion symbol ($\subset$) in the connecting line. A shared subclass is a subclass with various superclasses, so that every member (or instance) of the subclass must belong to all the superclasses. Naturally, a subclass inherits every attribute of all its superclasses. On the other hand, a category (union type) is similar to a shared subclass in which every member of the subclass must belong to only one of the superclasses, inheriting only the attributes of that superclass.

In this type of representation using the ER/EER model, constraints play a fundamental role: They express how the entities are related. Basically, we have the following types of constraints which can be represented in a schema using the ER/EER model [9], [14].

1) **The Participation constraint:** The participation of an entity in a relationship can be *total* or *partial*. If each instance must compulsorily relate to the other instance or instances of the relationship, then participation is said to be total. If this relationship is not mandatory for all instances belonging to this type, then participation is said to be partial. In ER diagrams, total participation is displayed as a double line connecting the participating entity type to the relationship, whereas partial participation is represented by a single line. Fig. 6 shows both constraint types. Section V studies fuzzy participation constraints.
2) **The Cardinality constraint:** This expresses whether the relationship between entities is "*one to one*" (1:1), "*one to many*" (1:N), or "*many to many*" (N:M). Section VI explains how to relax this constraint.
3) **The Completeness constraint on specializations:** It may be *total* or *partial*. A total specialization constraint specifies that every instance in the superclass entity must be a member of one (or some) of the subclasses entity in the specialization. This is shown in EER diagrams by using a double line to connect the superclass to the circle referred to as the specialization circle, to which all the subclasses are joined using a single line with the inclusion symbol. A single line is used to display a partial specialization, which allows an instance not to belong to any of the subclasses. The inverse is not possible since by definition each member of a subclass must be a member of the superclass. Section VIII includes an explanation of this constraint in a fuzzy model.
4) **Disjoint or overlapping constraints on specializations:** A disjoint specialization occurs when subclasses are disjoint, i.e., every member of the superclass must belong to a maximum of one of the subclasses. Disjoint specializations are shown in EER diagrams by using a circle with the letter "d." An overlapping specialization permits the subclasses to contain common elements, i.e., each member of the superclass may belong to various subclasses. Overlapping specializations are shown in EER diagrams by using a circle with the letter "o." Section X studies fuzzy disjoint and overlapping constraints on specializations. Additionally, Section IX examines the cardinality constraint on overlapping specializations, a constraint which is not studied in classic EER models.
5) **Completeness constraint in union types:** A category [13], [14] can be *total* or *partial*. A category is total if every superclass instance must be a member of the category. This is shown in EER diagrams by using a double line to connect the category with a circle with the union symbol ($\cup$). This is a strange case since this union type can be represented using a total disjoint specialization (the superclass is the category and the subclasses are all superclasses of the union type). A category is partial if every superclass instance may or may not be a member of the category. This is shown in EER diagrams by using a single line to connect the category to the circle with the union symbol. The classic model does not study the participation constraint of each superclass in the category. Section XII discusses these two constraints in a fuzzy model.

In addition, the **(min,max) notation** allows for the expression of the participation and cardinality of an entity in a relationship. In this notation, min and max indicate, respectively, the minimum and maximum number of entity instances which take part in the relationship. The (min,max) notation is better as it allows for the use of numbers other than 1 and N. It can clearly be seen that the (min,max) notation includes participation and cardinality in classic ER models. Section VII studies fuzzy (min,max) notation on relationships.

If a relationship of the ER model has a degree greater than 2, the constraints are also applicable to each entity participating in such a relationship. In this case, each entity treats the rest of the entities which participate in the relationship as if they were a single entity.

III. Fuzzy Sets: Fuzzy Quantifiers

In 1965, Zadeh defined the concept of fuzzy sets [52] based on the idea that there are sets in which it is not totally clear whether an element belongs to the set or not, or the intensity of membership is gradual. Sometimes an element belongs to the set to a certain degree which is called the membership degree. For example, the set of tall people is a fuzzy set because there is no height limit establishing the minimum height for a person to be considered tall.

A fuzzy set A is characterized by a membership function μ_A mapping the elements of a domain, space, or universe of discourse U to the unit interval $[0, 1]$

$$\mu_A : U \longrightarrow [0, 1]. \quad (1)$$

A fuzzy set A of U may therefore be represented as a set of ordered pairs of a generic element $u \in U$ and its membership degree as

$$A = \{\mu_A(u)/u \mid u \in U\}. \quad (2)$$

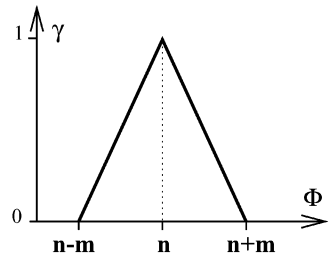

Fig. 1. Function "approximately n" (n±m, where m is a margin).

Clearly, a fuzzy set is a generalization of the concept of a set whose membership function takes on only two values $\{0, 1\}$. The value $\mu_A(u)$ describes a degree of membership of u in A. The closer $\mu_A(u)$ is to the value 1, the greater the membership of the object u is to the fuzzy set A. The values of membership vary between 0 (does not belong at all) and 1 (total belonging).

A fuzzy number is a fuzzy set, where U is a numerical domain (normally the real numbers $\mathbb{R}$). Fig. 1 shows a possible definition of the membership function of the fuzzy number "approximately n." The margin value m indicates the limits of the fuzzy set. It can be clearly seen that the nearer a number is to the value n, the greater its membership to "approximately n."

From this simple concept a complete mathematical and computing theory has been developed which facilitates the solution of certain problems [37]. Fuzzy logic has been applied to a multitude of objectives such as: control systems, modeling, simulation, pattern recognition, information or knowledge systems (databases, knowledge management systems, case-based reasoning systems, expert systems, ...), computer vision, artificial intelligence, artificial life, etc.

A. Fuzzy Quantifiers

Fuzzy or linguistic quantifiers [17], [19], [29], [30], [48], [51] allow us to express fuzzy quantities or proportions in order to provide an approximate idea of the number of elements of a subset fulfilling a certain condition or of the proportion of this number in relation to the total number of possible elements.

Fuzzy quantifiers can be *absolute* or *relative*.

- **Absolute quantifiers** express quantities over the total number of elements of a particular set, stating whether this number is, for example, "much more than 10," "close to 100," "a great number of," ... Generalizing this concept, we can consider fuzzy numbers as absolute fuzzy quantifiers, in order to use expressions like "approximately between 5 and 10," "approximately 8," ... Note that the expressed value may be positive or negative.

  In this case, we can see that the truth of the quantifier depends on a single quantity. For this reason, the definition of absolute fuzzy quantifiers is, as we will see, very similar to that of fuzzy numbers.
- **Relative quantifiers** express measurements over the total number of elements which fulfil a certain condition depending on the total number of possible elements (the proportion of elements). Consequently, the truth of the quantifier depends on two quantities. This type of quantifier is used in expressions such as "the majority" or "most," "the minority," "little of," "about half of," ...

  In this case, in order to evaluate the truth of the quantifier we need to find the total number of elements which fulfil the condition and consider this value with respect to the total number of elements which could fulfil it (including those which fulfil it and those which do not fulfil it).

Some quantifiers such as "many" and "few" can be used in either sense, depending on the context [29].

In [51], absolute fuzzy quantifiers are defined as fuzzy sets in the interval $\mathbb{R}^+$ (positive real numbers) and relative quantifiers as fuzzy sets in the interval [0, 1]. We have extended the definition of absolute fuzzy quantifiers to $\mathbb{R}$. That is to say that a quantifier Q is represented as a function Q whose domain depends on whether it is absolute or relative

$$Q_{\text{abs}} : \mathbb{R} \longrightarrow [0,1]$$
$$Q_{\text{rel}} : [0,1] \longrightarrow [0,1] \tag{3}$$

where the domain of Q_{rel} is [0, 1] because the division $a/b \in [0,1]$, where a is the number of elements fulfilling a certain condition and b is the total number of existing elements.

In order to know the fulfillment degree of the quantifier over the elements which fulfil a certain condition, we apply the function Q of the quantifier to the value of quantification Φ

$$\Phi = \begin{cases} a, & \text{if } Q \text{ is absolute} \\ a/b, & \text{if } Q \text{ is relative.} \end{cases} \tag{4}$$

Thus, the fulfillment degree is $Q(\Phi)$. If the function of the quantifier (absolute or relative) $Q(\Phi)$, has the value 1, this indicates that this quantifier is completely satisfied. The value 0 indicates, on the other hand, that the quantifier is not fulfilled at all. Any intermediate value indicates an intermediate fulfillment degree for the quantifier.

Example 1: "Approximately 8" is an absolute fuzzy quantifier, defined as shown in Fig. 1, with n$=8$, and the margin m$=3$, for example. "Almost all" is a relative fuzzy quantifier, defined as shown in Fig. 2. □

IV. THRESHOLDS AND FUZZY QUANTIFIERS FOR RELAXING CONSTRAINTS

Applied in the context of databases, the usefulness of fuzzy quantifiers is shown by the flexibility they offer when carrying out queries which involve these quantifiers, as for example in the division operation of relational algebra in fuzzy or classical databases [19]. Applied in the context of conceptual data models, fuzzy quantifiers allow expressions about the number of instances which satisfy a given condition, or the proportion with respect to the total. We will study this in subsequent sections. Of course, the quantifier Q must be previously defined in the model's data dictionary (metadata).

In this context, we need a threshold $\gamma \in [0,1]$ indicating the minimum fulfillment degree that must be satisfied. This threshold will be written in square brackets: $Q[\gamma]$. For example, we may use "almost_all [0.2]" indicating that this fuzzy quantifier must be satisfied at a minimum degree of 0.2. Consequently, the underlining constraint requires that

$$Q(\Phi) \geq \gamma. \tag{5}$$

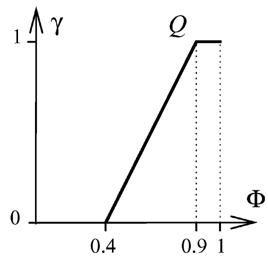

Fig. 2. Relative fuzzy quantifier "almost all": $Q(\Phi) = 0 \leftrightarrow \Phi < 0.4, Q(\Phi) = 1 \leftrightarrow \Phi > 0.9$ and $Q(\Phi) = 2(\Phi - 0.4) \leftrightarrow \Phi \in [0.4, 0.9]$.

Every time the database is modified, the DBMS computes Φ and checks whether (5) is satisfied. The meaning of Φ will be defined in subsequent sections because it depends on where the fuzzy quantifier is used. In order to simplify the expression, we can set a default value for γ at 0.5, for example. We consider 0.5 to be a good default value since it is in the middle of the interval [0, 1], but it may be changed.

If Q is an increasing function, then we can simplify (5) because

$$\Phi \geq Q^{-1}(\gamma). \quad (6)$$

Similarly, if Q is a decreasing function, then

$$\Phi \leq Q^{-1}(\gamma). \quad (7)$$

The last two equations may be useful because Q and γ are constants, whereas Φ is a varying value. Value Φ may change with every DML sentence `INSERT`, `DELETE`, or `UPDATE`. In this way, we can store $Q^{-1}(\gamma)$, avoiding the use of Q with those DML sentences.

In addition, another optional value δ can be established, which is greater than the threshold γ, in the following way: $Q[\gamma, \delta]$ such that $\gamma < \delta$. The value δ is more restrictive than γ and it establishes that, when the constraint is unfulfilled with this higher value, the DBMS will inform the user, but it will permit the modification of the database which is under way. Thus, if the quantifier is unfulfilled with a value between γ and δ, then the DBMS must warn the user (or only the database administrator). Both values would be close, in order to avoid too many warnings from the DBMS. Therefore, the warning message is generated when (5) is satisfied and the following Equation is *not* satisfied:

$$Q(\Phi) \geq \delta. \quad (8)$$

In other words, the warning area is defined with

$$\delta > Q(\Phi) \geq \gamma. \quad (9)$$

Finally, if (5) is false, then it defines the not allowed area and an error message must be generated.

Example 2: Fig. 3 depicts a fuzzy quantifier with the thresholds γ and δ. We want to evaluate to what extent value Φ satisfies the quantifier. This evaluation is carried out by $Q(\Phi)$. It should be noted that these thresholds divide the domain of Φ into three areas: the allowed area, the not allowed area and the warning area. The warning area is included in the allowed area. Note that the not allowed area is defined when (5) is false. □

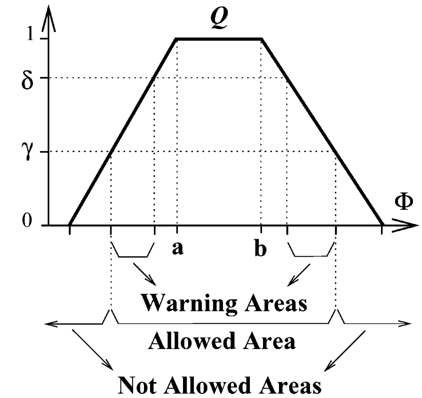

Fig. 3. Thresholds γ and δ in a fuzzy quantifier "approximately between a and b," and its generated areas.

Therefore, a fuzzy quantifier can be written in three ways.

1) Quantifier without a threshold γ: Default threshold is $\gamma = 0.5$. For example, approx_2.
2) Quantifier with a threshold γ: For example, approx_8[0.25].
3) Quantifier with two thresholds γ and δ, with $\gamma < \delta$: For example, approx_3[0.25,0.75].

V. Fuzzy Participation Constraint on Relationships

In classic ER models, the participation of an entity in a relationship can be *total* or *partial* (Section II). Furthermore, in the fuzzy model that we propose here, the participation of an entity in a relationship can be fuzzy using a relative fuzzy quantifier (principally).

Definition 1: Let E_1 and E_2 be two entities and R a relationship between them. A **fuzzy participation constraint** of E_1 in R is represented using a zigzag line (or broken line) joining E_1 and R, indicating on this line which quantifier Q has been used, followed optionally by one or two thresholds, $[\gamma]$ or $[\gamma, \delta]$, with the meaning and default value having the one explained previously. We propose another representation using a single line crossed with an arc labeled with Q.

This constraint asserts (5), with Φ defined by (4), where a is now the number of E_1 instances related to E_2, and b is the total number of instances in E_1.

If Q is used with two thresholds, then it defines a warning area (see Section IV). A warning message must, therefore, be generated when the condition is satisfied with γ and it is *not* satisfied with δ. It should be noted that the warning area is included in the allowed area. Therefore, the warning area is defined when (5) is satisfied and (8) is *not* satisfied. In other words, the warning area is defined with (9). Finally, if (5) is false then it defines the not allowed area. □

This fuzzy constraint implies that every DML sentence may generate an error or a warning when the fuzzy quantifier is not satisfied. This message forces the user to maintain a "good" database or warns the user when the database is not "good enough."

Example 3: Suppose we have an Employee entity and a Project entity linked by the relationship Works_for. The participation of Employee in this relationship can be represented by

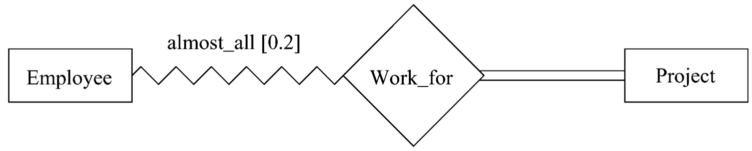

Fig. 4. Example 3: Fuzzy participation constraint in an ER model, using the fuzzy quantifier almost_all.

a relative fuzzy quantifier like "almost all" (Fig. 2), indicating that "almost all employees work for some project." Fig. 4 represents these two entities, the relationship between them and the fuzzy constraint.

The threshold $\gamma = 0.2$ in "almost_all [0.2]" indicates the minimum degree with which this quantifier must be fulfilled in the database. Thus, if we divide the number of workers who work for a project (value a) by the total number of employees in the database (value b), the result Φ should be, in accordance with (6), a value greater than or equal to $Q^{-1}(0.2) = 0.5$, because this is the value (on the X axis) for which the quantifier "almost all" attains a degree $0.2 : Q(0.5) = 0.2$. From value 0.5 of Φ this quantifier obtains a degree greater than or equal to 0.2, which was the constraint imposed by the threshold γ in the initial quantifier. The constraint then establishes that $\Phi \geq 0.5$ must be satisfied.

In this example, the value 0.5 obtained by the expression $Q^{-1}(0.2)$ indicates the constraint that, in our database, a minimum of 50% of the employees must work for some project. □

In general, if Q is a relative fuzzy quantifier with an increasing function, then (6) states that the constraint must be satisfied in the database in a minimum *percentage* of $100\,Q^{-1}(\gamma)$. In this case, it is also possible to express this percentage instead with the quantifier Q and the threshold γ. Although this may appear easier, it must be noted that the intuitive and natural expressiveness of the quantifier is lost, that this is not valid with absolute fuzzy quantifiers and the method must be adapted for decreasing fuzzy quantifiers. Fuzzy quantifiers are therefore easy and general as well as very expressive and intuitive.

On the practical level, this will be implemented as a *trigger* that, in each operation of the type `UPDATE`, `DELETE`, or `INSERT`, checks the value Φ and if (5) is false, then the DBMS must produce an error message indicating the nonfulfillment of this fuzzy constraint and the operation is aborted. On the other hand, if (9) is true, then a warning message must be generated but the operation is allowed. Finally, in any other case the operation is normally allowed.

It should be noted that fuzzy quantifiers expressed in this type of constraint can also be absolute, although, due to the significance of a participation constraint, this will generally be relative since the number of entity instances would vary too much. In the case of an absolute fuzzy quantifier, for example "many" or "approximately between 100 and 200," this quantifier will restrict the quantity of entity instances related to the other entity. In our example, the fuzzy quantifier would restrict the number of employees who are assigned to work on any project.

In some models it might even be useful to establish several fuzzy quantifiers as a constraint on fuzzy participation. In this case, all fuzzy quantifiers in the same constraint must be coherent, as two quantifiers can be contradictory.

A fuzzy participation constraint is not as restrictive as a total participation constraint, nor as permissive as the partial participation constraint, so that the fuzzy participation constraints extend the ER model, allowing a new expressiveness which would have been impossible in the traditional model.

VI. Fuzzy Cardinality Constraint on Relationships

Fuzzy participation constraints establish a condition globally on the entity instances. On the other hand, fuzzy cardinality constraints establish fuzzy conditions on each instance in a particular and individual way. In classical modeling, the cardinality constraint states whether the relationship between entities is "*one to one*" (1:1), "*one to many*" (1:N), or "*many to many*" (N:M). The model we propose allows us to express cardinality as a fuzzy value using an absolute fuzzy quantifier (principally).

Definition 2: Let E_1 and E_2 be two entities and R be a relationship between them. We suppose that e_i with $i = 1, 2, \ldots, b_1$ are the instances of E_1, and w_j with $j = 1, 2, \ldots, b_2$ are the instances of E_2. A **fuzzy cardinality constraint** is defined with two quantifiers, separated by the notation ":" Q_1:Q_2, just below the diamond which represents the relationship between both entities. The quantifier on the left of the separator ":" will correspond to the entity on the left (or above) and the quantifier on the right will correspond to the other entity (E_1 and E_2 respectively). This constraint establishes two conditions.

1) Condition of Q_1:

$$Q_1(\Phi_{1i}) \geq \gamma_1 \qquad \forall i = 1, 2, \ldots, b_1 \tag{10}$$

where γ_1 is the threshold for Q_1, b_1 is the total number of instances in E_1 and Φ_{1i} with $i = 1, 2, \ldots, b_1$ is defined by

$$\Phi_{1i} = \begin{cases} a_i, & \text{if } Q_1 \text{ is absolute} \\ a_i/b_2, & \text{if } Q_1 \text{ is relative} \end{cases} \tag{11}$$

with a_i being the number of E_2 instances related with the instance $e_i \in E_1$, and b_2 is the total number of instances in E_2.

2) Condition of Q_2 :

$$Q_2(\Phi_{2j}) \geq \gamma_2 \qquad \forall j = 1, 2, \ldots, b_2 \tag{12}$$

where γ_2 is the threshold for Q_2 and Φ_{2j} with $j = 1, 2, \ldots, b_2$ is defined by

$$\Phi_{2j} = \begin{cases} a_j, & \text{if } Q_2 \text{ is absolute} \\ a_j/b_1, & \text{if } Q_2 \text{ is relative} \end{cases} \tag{13}$$

with a_j being the number of E_1 instances related to the instance $w_j \in E_2$.

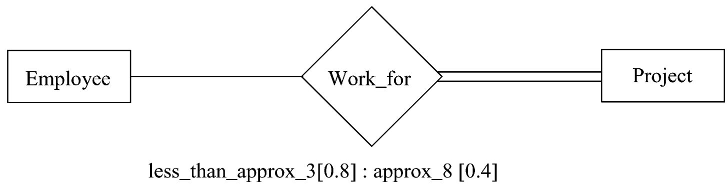

Fig. 5. Example 4: Fuzzy cardinality constraint in an ER model.

The warning area is similarly defined using δ_1 and δ_2, respectively. □

This fuzzy constraint has an effect on each instance and must be satisfied by each one. Generalizing, quantifier Q_1 must be satisfied for all quantification values Φ_{1i} with $i = 1, 2, \ldots, b_1$, i.e., Φ_{1i} must be in the allowed area of Q_1 with its threshold γ_1. On the other hand, quantifier Q_2 must be satisfied for all quantification values Φ_{2j} with $j = 1, 2, \ldots, b_2$, i.e., Φ_{2j} must be in the allowed area of Q_2 with its threshold γ_2.

Example 4: Following the previous example, if we suppose that the entity Employee is on the left of the relationship Works_for and the entity Project is on the right, a fuzzy cardinality constraint is shown in Fig. 5.

These constraints express the condition that *each* employee will work for a maximum of approximately three projects and each project will have approximately eight employees, requiring both constraints to be satisfied with the minimum fulfillment degrees indicated in square brackets. □

It should be noted that the fuzzy quantifier of this type of constraint can also be relative, although due to the meaning of a cardinality constraint, this will generally be absolute. In the case of a relative fuzzy quantifier, this quantifier will indicate the number of instances of the other entity to which each entity is related, with respect to the total number of instances of the other entity. Thus, if in Example 4 we use the quantifier "almost all" on the left, this means that "each employee must work for almost all the existing projects."

In some models it might even be useful to establish various fuzzy quantifiers on one or in both sides of a fuzzy cardinality constraint. Of course, in this case all the fuzzy quantifiers in the same constraint must be coherent.

If a relationship joins three or more entities (a relationship with degree greater than two) we can put the fuzzy cardinality quantifier associated with each entity at the side of the arc which joins this entity with the relationship. If there is already a quantifier for the fuzzy participation constraint, then in order to avoid ambiguity we must put the text "Card:" in front of the cardinality quantifier.

VII. Fuzzy (min,max) Notation on Relationships

Nonfuzzy participation and cardinality of an entity in a relationship can be expressed using this notation. This notation is more expressive both in classical and fuzzy modeling. In fuzzy modeling both min and max can have values which are fuzzy quantifiers in a similar way to the one explained previously.

Definition 3: Let E_1 and E_2 be two entities and R be a relationship between them. We denote the instances of E_1 as e_i with $i = 1, 2, \ldots, b_1$, and the instances of E_2 as w_j with $j = 1, 2, \ldots, b_2$. A **fuzzy (min,max) notation** of E_1 on R is represented using two fuzzy quantifiers in parenthesis $(Q_{\min}, Q_{\max})$ beside the line joining E_1 and R. This constraint establishes that

$$\lambda_{\min} \le \Phi_{\min,i} \wedge \lambda_{\max} \ge \Phi_{\max,i} \qquad \forall i = 1, 2, \ldots, b_1 \quad (14)$$

where b_1 is the total number of instances in E_1 and

$$\Phi_{\min,i} = \begin{cases} a_i, & \text{if } Q_{\min} \text{ is absolute} \\ a_i/b_2, & \text{if } Q_{\min} \text{ is relative} \end{cases} \quad (15)$$

$$\Phi_{\max,i} = \begin{cases} a_i, & \text{if } Q_{\max} \text{ is absolute} \\ a_i/b_2, & \text{if } Q_{\max} \text{ is relative} \end{cases} \quad (16)$$

with a_i being the number of E_2 instances related with the instance $e_i \in E_1$, and b_2 is the total number of instances in E_2. Furthermore

$$\lambda_{\min} = \min\left\{\alpha : \alpha = Q_{\min}^{-1}(\gamma_{\min})\right\} \quad (17)$$

$$\lambda_{\max} = \max\left\{\beta : \beta = Q_{\max}^{-1}(\gamma_{\max})\right\} \quad (18)$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are the minimum thresholds for $Q_{\min}$ and $Q_{\max}$, respectively. Thus, the allowed area is the interval $[\lambda_{\min}, \lambda_{\max}]$. The warning area is defined when the thresholds $\delta_{\min}$ and $\delta_{\max}$ are used for $Q_{\min}$ and $Q_{\max}$, respectively. The warning message must be shown when (14) is satisfied and the following equation is *not* satisfied:

$$\lambda'_{\min} \le \Phi_{\min,i} \wedge \lambda'_{\max} \ge \Phi_{\max,i} \qquad \forall i = 1, 2, \ldots, b_1 \quad (19)$$

where

$$\lambda'_{\min} = \min\left\{\alpha : \alpha = Q_{\min}^{-1}(\delta_{\min})\right\} \quad (20)$$

$$\lambda'_{\max} = \max\left\{\beta : \beta = Q_{\max}^{-1}(\delta_{\max})\right\}. \quad (21)$$

Hence, the warning area is the union of two intervals: $[\lambda_{\min}, \lambda'_{\min}] \cup [\lambda'_{\max}, \lambda_{\max}]$. We know that $\lambda_{\min} < \lambda'_{\min}$ and $\lambda'_{\max} < \lambda_{\max}$, because $\gamma_{\min} < \delta_{\min}$ and $\gamma_{\max} < \delta_{\max}$, respectively, and the quantifiers are defined with convex functions. Note that (20) and (21) are similar to (17) and (18) replacing $\gamma_{\min}$ and $\gamma_{\max}$ by $\delta_{\min}$ and $\delta_{\max}$, respectively.

In other words, (19) may be changed to obtain an Equation that must be satisfied. Applying De Morgan's law, the warning area is defined when the following equation is satisfied:

$$\lambda'_{\min} > \Phi_{\min,i} \vee \lambda'_{\max} < \Phi_{\max,i} \qquad \forall i = 1, 2, \ldots, n. \quad (22)$$

It should be noted that if a constraint exists on E_1 using (min,max) notation, then this constraint has an effect on each instance and must be satisfied by each one. These two quantifiers indicate, respectively, the *minimum* and *maximum* number of E_2 instances related with each E_1 instance.

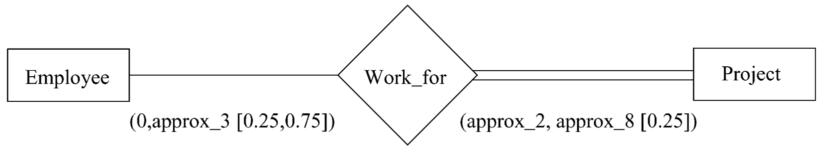

Fig. 6. Example 5: Fuzzy (min,max) notation in an ER model.

Some observations follow.

- If $Q_{\min}(0) \geq \gamma_{\min}$, then we cannot use $Q_{\min}[\gamma_{\min}]$. In this case, we must use 0 instead of $Q_{\min}$: $[0, Q_{\max}]$.
- If $Q_{\max}(\Psi) \geq \gamma_{\max}$, where Ψ is the maximum value in the underlying domain of $Q_{\max}$ (if $Q_{\max}$ is relative, $\Psi = 1$), then we cannot use $Q_{\max}[\gamma_{\max}]$. In this case, we must use the letter "N" instead of $Q_{\max}$, expressing a cardinality constraint "*to many*": $[Q_{\min}$,N].
- If $Q_{\min}$ and $Q_{\max}$ are of the same type (absolute or relative), then $\Phi_{\min,i} = \Phi_{\max,i}$.
- In conclusion, we must use fuzzy quantifiers which express a good constraint. An example of a bad constraint is: [approx_3_or_less, almost_all].

Example 5: In the context of the previous examples, we can use the following constraints using the fuzzy (min, max) notation. These constraints are represented in Fig. 6. On the employee side, the (min, max) constraint indicates that an employee may work for no projects (0 as *minimum*) and as a *maximum* can work for approximately three projects. The two values after the quantifier indicate that if this is fulfilled to a degree greater than or equal to 0.75 the operation will normally be permitted; if it is fulfilled to a degree between 0.25 and 0.75, then the user will be informed but the operation will be allowed; and if the constraint is fulfilled to a degree of less than 0.25, this means that the constraint is not being fulfilled because it reaches an intolerable degree and therefore the operation under way must not be permitted.

On the Project side in the same figure, we can find a constraint indicating that each project must have a *minimum* of approximately 2 employees working for it (with a degree of 0.5 as a minimum). If the quantifier approx_2 is defined as a triangle as in Fig. 1 with $\mathrm{n} = 2$ and margin $= 2$, then the value 0.5 is achieved with the minimum value 1, so that this quantifier with the minimum degree 0.5 guarantees the total participation of the entity Project in the relationship Works_for. This possibility is also indicated by the double line which connects the entity to the relationship. It should be noted that we are using two quantifiers for the min value but both are coherent.

At the same time, the number of employees in each project is restricted to a *maximum* of approximately 8 (with a minimum degree of 0.25). □

In the classical ER model, the (min,max) notation substitutes the other two notations for the participation and cardinality constraints, since if $\min = 0$ we are dealing with a partial participation and if $\min > 0$ we are dealing with a total participation. On the other hand, if $\max = 1$ the relationship will be 1:1 or 1:N (on the side of 1) and if $\max > 1$ (or $\max = \mathrm{N}$) we are dealing with a relationship N:M or 1:N (on the side of N).

However, in the ER model with fuzzy constraints, the fuzzy (min, max) notation adds expressiveness to the conceptual model, but it can only substitute fuzzy cardinality constraints, as we will see later.

A. Fuzzy (min, max) Notation and Fuzzy Cardinality Constraints

With regard to cardinality constraints (Section VI), it is clear that the semantic of both notions is different. It should be noted that in Example 4, the quantifier "approx_8" indicates that a Project must have approximately eight employees, whilst in Example 5, the same quantifier indicates that a Project must have a *maximum* of approximately eight employees.

In general, a fuzzy cardinality constraint with any type of quantifier can be represented using the fuzzy (min,max) notation so that both values, min and max, have the value of this fuzzy quantifier. The fuzzy cardinality constraint expressed in Example 4 can be expressed in fuzzy (min,max) notation so that the minimum value is equal to the maximum and both have the value of "approx_8."

Expressiveness is also equivalent on the other side with one exception depending on the types of both quantifiers.

- If the (min,max) notation uses **two quantifiers of the same type** (absolute or relative), then this constraint can be expressed by means of a fuzzy cardinality constraint using a quantifier which embraces both. For example, the constraint in Example 5 can be expressed with the fuzzy cardinality notation using a wider quantifier, instead of the quantifiers "approx_2" and "approx_8," such as "approx_between_2_and_8" (similar to the trapezoidal membership function in Fig. 3 with $\mathrm{a} = 2$ and $\mathrm{b} = 8$). It can be observed that the resulting wider quantifier may be automatically generated starting from the other two quantifiers (min and max).
- If the (min,max) notation uses **two quantifiers of different types** (one absolute and one relative), then this restriction cannot be expressed with a single fuzzy cardinality quantifier. This is because different types of quantifiers have different domains and they cannot be joined in another quantifier which embraces both.

It is important to note that this second kind of (min,max) notation should be uncommon because perhaps it is not intuitive to check that the two fuzzy quantifiers are not contradictory. They can also be contradictory after a DML sentence. For example, if we use approx_half instead of approx_2 in Example 5 (Fig. 6), both quantifiers in (min,max) notation (approx_half and approx_8) are not contradictory when the number of employees is six, for example, because the constraint is (approx_3,approx_8). However, if the number of employees is 200, for example, both quantifiers are contradictory, because approx_100 is clearly greater than approx_8.

As in fuzzy cardinality constraints, because of their meanings, the (min,max) notation will preferably use two absolute quantifiers, although two relative quantifiers are also accepted here.

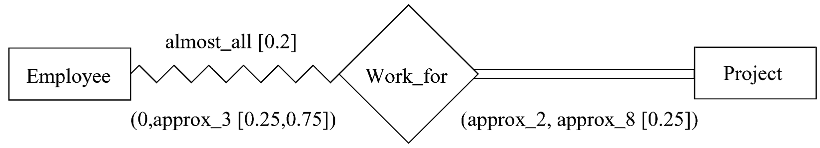

Fig. 7. Example 6: Fuzzy participation constraint with another constraint using fuzzy (min,max) notation in an ER model.

B. Fuzzy (min, max) Notation and Fuzzy Participation Constraints

On the other hand, the (min,max) notation and a fuzzy participation constraint (Section V) are not exclusive. While a fuzzy participation constraint establishes a condition on all entity instances globally, the (min,max) notation restricts the relationship of each instance (individually) with the other participating entity.

Example 6: If we merge Examples 3 and 5 we obtain the model in Fig. 7. It can be seen that both constraints are different and coherent. It is also important to note the different meanings even though we use the following (min,max) notation: (almost_all,almost_all). □

For these reasons, the most useful notations are the fuzzy (min,max) notation (mainly with absolute fuzzy quantifiers) and the notation for fuzzy participation constraints (mainly with relative fuzzy quantifiers). The notation for fuzzy cardinality restriction can be eliminated because this can be expressed using the fuzzy (min,max) notation.

In a new expression for fuzzy participation constraints, we can use fuzzy (min,max) notation instead of the quantifier Q in Definition 1. These minimum and maximum values restrict the quantity of entity instances related to the other entity. We must distinguish this notation for fuzzy participation and the usual fuzzy (min,max) notation. This new notation refers to the number of instances (in the constrained entity) related to the other entity. Usual fuzzy (min,max) notation refers to the number of instances in the other entity related to *each* instance in the constrained entity. This extension is not very useful but the formal definition for it is easy using Definitions 1 and 3.

VIII. Fuzzy Completeness Constraint on Specializations

In EER models, the relationship between a class and all its subclasses can be *total* or *partial* (Section II). In our fuzzy model, this constraint can be fuzzy mainly utilizing a relative fuzzy quantifier, although, as indicated in the case of participation constraints, they can also be absolute fuzzy quantifiers.

Definition 4: Let E be a superclass and $S_1, S_2, \ldots, S_n$ the set of its n subclasses. A **fuzzy completeness constraint** is represented by a zigzag line, labeled with a quantifier Q and its required thresholds. This constraint asserts the (5), with Φ defined by (4), where a is the number of E instances which belong to "any" subclass or subclasses, and b is the total number of instances in E.

The warning area is defined when (9) is satisfied. □

Example 7: Let us consider the model in Fig. 8 depicting an entity Employee which is a superclass with two subclasses defined by the attribute Contract_Type: Permanent and Temporary. The zig-zag line with the relative fuzzy quantifier "almost

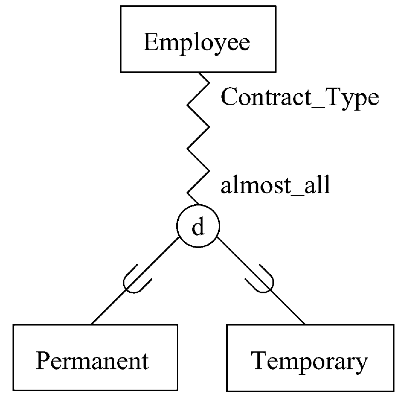

Fig. 8. Example 7: Fuzzy completeness constraint on an attribute-defined specialization with the defining attribute Contract_Type.

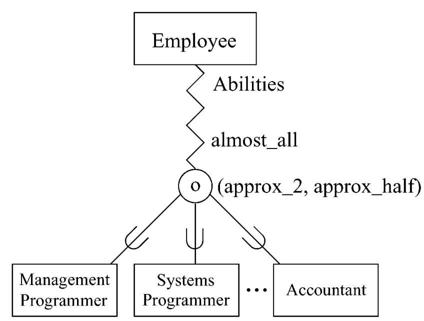

Fig. 9. Examples 8 and 9: Fuzzy completeness constraint and fuzzy cardinality constraint on an overlapping specialization.

all" (Fig. 2) indicates that "Almost all employees must have a Permanent or Temporary contract, but other minority contract types may exist (work experience, grants . . .)." These other contract types are not included in the model for various reasons (unknown types, types without own attributes . . .). □

In the previous example, the specialization is disjoint (with a "d" in the circle) since there cannot be an employee with various types of contracts. However, fuzzy constraints can also be applied to overlapping specializations (with an "o" in the circle) as shown in the following example.

Example 8: Let us consider an entity Employee which is a superclass with various subclasses defining the abilities of the employees: Management_Programmer, Systems_Programmer, Internet_Programmer, Analyst, Graphic_Designer, Accountant . . . just like Fig. 9. A relative fuzzy quantifier like "almost all" (Fig. 2) indicates that "Almost all employees must have one or *some* of the abilities expressed in the subclasses." □

In a new expression for fuzzy completeness constraints, we can use fuzzy (min,max) notation instead of the quantifier Q in Definition 4. These minimum and maximum values restrict the quantity of superclass instances which belong to "any" subclass.

This extension is not very useful but the formal definition for it is easy using Definitions 3 and 4.

IX. Fuzzy Cardinality Constraint on Overlapping Specializations

In an overlapping specialization, we can also establish the minimum and maximum number of subclasses to which each member of the superclass can belong in a flexible manner. This can easily be expressed using the fuzzy (min,max) notation.

Definition 5: Let E be a superclass and $S_1, S_2, \ldots, S_b$ the set of b subclasses of an overlapping specialization. In addition, we suppose that e_i with $i = 1, 2, \ldots, n$ are the instances of E. A **fuzzy cardinality constraint** on this **overlapping specialization** is represented with a fuzzy (min,max) notation, $(Q_{\min}, Q_{\max})$, next to the circle containing the letter "o" (overlapping). This constraint establishes that

$$\lambda_{\min} \leq \Phi_{\min,i} \wedge \lambda_{\max} \geq \Phi_{\max,i} \qquad \forall i = 1, 2, \ldots, n \quad (23)$$

where

$$\Phi_{\min,i} = \begin{cases} a_i, & \text{if } Q_{\min} \text{ is absolute} \\ a_i/b, & \text{if } Q_{\min} \text{ is relative} \end{cases} \quad (24)$$

$$\Phi_{\max,i} = \begin{cases} a_i, & \text{if } Q_{\max} \text{ is absolute} \\ a_i/b, & \text{if } Q_{\max} \text{ is relative} \end{cases} \quad (25)$$

with a_i being the number of subclasses to which instance e_i belongs. Furthermore, $\lambda_{\min}$ and $\lambda_{\max}$ are defined in the same way as in (17) and (18).

The warning area is defined when the thresholds $\delta_{\min}$ and $\delta_{\max}$ are used for $Q_{\min}$ and $Q_{\max}$, respectively. The warning message must be shown when (23) is satisfied and the following equation is *not* satisfied:

$$\lambda'_{\min} \leq \Phi_{\min,i} \wedge \lambda'_{\max} \geq \Phi_{\max,i} \qquad \forall i = 1, 2, \ldots, n \quad (26)$$

where $\lambda'_{\min}$ and $\lambda'_{\max}$ are defined in (20) and (21). We can also apply De Morgan's law here. □

This fuzzy constraint has an effect on each superclass instance and must be satisfied by each one. In general, both min and max should be absolute quantifiers, although relative quantifiers will also be accepted (with regards to the total number of subclasses, value b).

Example 9: Continuing with Example 8, we can establish a fuzzy cardinality constraint on the overlapping specialization, such as: (approx_2, approx_half).

This establishes the constraint whereby each employee must appear in a minimum of "approximately 2" skills and in a maximum of "approximately half" of the existing skills (or subclasses).

This schema is also depicted in Fig. 9. It should be noted that the fuzzy quantifier almost_all is a fuzzy completeness constraint (zig-zag line) and the (min,max) notation is used for a fuzzy cardinality constraint. □

Finally, it is important to note that the quantifiers can be of any type (absolute or relative). In this case each quantifier can also be followed, optionally of course, by one or two fulfillment degrees in square brackets $[\gamma, \delta]$, with the same meaning and default value as explained previously (Section IV).

X. Fuzzy Disjoint and Fuzzy Overlapping Constraints on Specializations

In specializations, the disjoint constraint specifies that the subclasses of the specialization must be disjoint. This means that an entity can be a member of at most one of the subclasses (zero or one). If the subclasses are not obliged to be disjoint, this is an overlapping specialization. Thus, it can be interesting to include to what extent the superclass instance belongs to each of the subclasses using linguistic labels ("a lot," "a little," ...) or, more simply, the membership degrees in the interval [0, 1].

It should be noted that it is to consider each subclass as a fuzzy subset of the superclass. As with all fuzzy sets, its elements are not clearly defined, since each element can belong to the fuzzy set with a certain degree. Therefore, we can define the concept of fuzzy entity in the following way.

Definition 6: Let E be an entity with n instances, $e_1, e_2, \ldots, e_n$. Entity E is a **fuzzy entity** if the membership function of all its instances to E is fuzzy. In other words, a fuzzy entity must define a membership function μ, where $\mu(e_i) \in [0, 1]$ is the membership degree of e_i to E with $i = 1, 2, \ldots, n$.

Fuzzy entities are represented using rectangles with dashed lines. □

This definition allows for two new definitions according to the specialization type.

Definition 7: Let E be a superclass with n subclasses, $S_1, S_2, \ldots, S_n$, in a disjoint specialization. This specialization is a **fuzzy disjoint specialization** when at least one of the subclasses is a fuzzy entity, and for any instance $e \in E$, there is zero or one subclass S_i with $i \in \{1, 2, \ldots, n\}$ such that

$$\mu_i(e) > 0 \quad (27)$$

where $\mu_i(e)$ is the membership degree of e to S_i. Thus, as with any disjoint specialization: $S_1 \cap S_2 \cap \ldots \cap S_n = \oslash$.

This constraint will be represented by the letter "f" (fuzzy) before the letter "d" in the circle, i.e., "fd." □

Of course, if S_i is a nonfuzzy subclass, then $\mu_i(e) = 1$ if e belongs to S_i and $\mu_i(e) = 0$ if e does not belong to S_i.

Definition 8: Let E be a superclass with n subclasses, $S_1, S_2, \ldots, S_n$, in an overlapping specialization. This specialization is a **fuzzy overlapping specialization** when at least one of the subclasses is a fuzzy entity, and for any instance $e \in E$, there are zero or more subclasses S_i with $i \in \{1, 2, \ldots, n\}$ such that $\mu_i(e) > 0$, where $\mu_i(e)$ is the membership degree of e to S_i.

This constraint will be represented by the letter "f" (fuzzy) before the letter "o" in the circle, i.e., "fo." □

Note that these definitions do not force all the subclasses to be fuzzy entities. Definition 8 is more flexible than Definition 7, since an instance $e \in E$ may belong to some subclasses with different membership degrees.

These definitions have two points of view.

1) **From the point of view of subclasses:** Subclasses are fuzzy sets and their underlying domain is all the superclass instances, i.e., each superclass instance has a membership degree to each subclass (including the value zero).

TABLE I
REPRESENTING FUZZY SETS ON A SPECIALIZATION WITH n SUBCLASSES AND m SUPERCLASS INSTANCES

Instances \ Subclasses	S_1	S_2	…	S_n
e_1	$\mu_{S_1}(e_1)$	$\mu_{S_2}(e_1)$	…	$\mu_{S_n}(e_1)$
e_2	$\mu_{S_1}(e_2)$	$\mu_{S_2}(e_2)$	…	$\mu_{S_n}(e_2)$
⋮	⋮	⋮	…	⋮
e_i	$\mu_{S_1}(e_i)$	$\mu_{S_2}(e_i)$	…	$\mu_{S_n}(e_i)$
⋮	⋮	⋮	…	⋮
e_m	$\mu_{S_1}(e_m)$	$\mu_{S_2}(e_m)$	…	$\mu_{S_n}(e_m)$

Let S be a subclass of E. Then the fuzzy set of S is represented by (using the format of (2)):

$$\{\mu_S(e_1)/e_1, \mu_S(e_2)/e_2, \ldots, \mu_S(e_m)/e_m\} \tag{28}$$

where e_i, with $i = 1, 2, \ldots, m$, are all the instances of superclass E, and $\mu_S(e_i)$ is the membership degree of e_i to subclass S.

2) **From the point of view of superclass instances:** Each superclass instance may belong to some subclasses. This membership is measured with a fuzzy set. The underlying domain of this fuzzy set is the set of all subclass names. Let S_j, with $j = 1, 2, \ldots, n$, be the n subclasses of E. Then the fuzzy set of instance e_i is:

$$\{\mu_{S_1}(e_i)/S_1, \mu_{S_2}(e_i)/S_2, \ldots, \mu_{S_n}(e_i)/S_n\} \tag{29}$$

where $\mu_{S_j}(e_i)$, with $j = 1, 2, \ldots, n$, is the membership degree of e_i to subclass S_j. It is important to note that in disjoint specializations the number of subclasses for a superclass instance is one.

Both points of view work with fuzzy sets with a different discrete underlying domain. Thus, it may be represented with the format of Table I, where the first point of view is represented by the fuzzy sets given by the columns and the second point of view is given by the rows.

Example 10: Fig. 10 indicates that our conceptual schema is also concerned with storing the extent to which each employee belongs to each of the subclasses. Thus, the set of system programmers is a fuzzy set (an employee can belong to this set with a certain membership degree), whereas we suppose that the set of accountants is not a fuzzy set (an employee can or cannot belong to this set). This is the first point of view.

The second point of view starts with a particular employee: an employee who is an expert at programming management applications, although he/she may also be skilled in other types of applications and less skilled as an analyst, could be represented in the database by the following fuzzy set: {1/Management_Programmer, 0.8/Systems_Programmer, 0.3/Analyst}. It should be noted that the underlying domain is the set of all the subclass names.

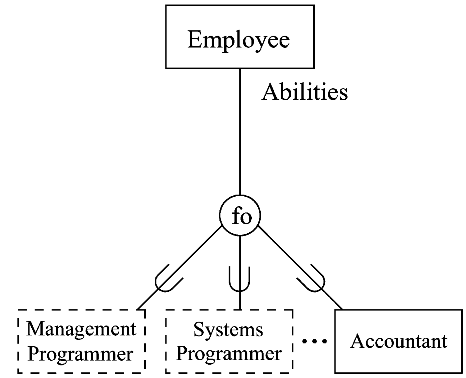

Fig. 10. Example 10: Fuzzy overlapping specialization.

This database model will allow us to make selections of the type: "Find the name of the *best* management applications programmer amongst those who are not assigned to *many* projects and who is at least a *regular* analyst." □

This constraint does not prevent the use of other fuzzy constraints (completeness or cardinality). These cases must be studied in order to define the method with which the DBMS ensures the fulfillment of these constraints:

1) If a **fuzzy completeness constraint** exists, then the DBMS must compute whether each superclass instance belongs to some subclass, for example, in order to decide if "almost all" superclass instances belong to some subclass. The problem is that membership is now fuzzy. The membership degree of an instance to the subclasses may be computed in various ways: 1) by using the greatest membership degree of this instance to any subclass, i.e., the height (function Hgt) [37] of the second point of view fuzzy set; or 2) by using the fuzzy set cardinality (function Card) [37] of the second point of view fuzzy set (adding all the membership degrees) or by using generalized measures, such as the fuzzy set energy [31]. We can certainly set a minimum threshold λ in order to decide whether a superclass instance belongs to some subclass.

Then, in order to compute the value of a in Definition 4 we must count how many instances of E have a fuzzy membership degree to "any" subclass or subclasses. The fuzzy membership degree is solved with the two previous options. The problem is then to count these elements. We propose the following four options, where b is the total number of instances in E, and a is computed by:

$$a = \sum_{i=1}^{b} \alpha_i. \tag{30}$$

The definition of α_i gives the following four options.

a) Option 1:

$$\alpha_i = \begin{cases} \mathrm{Hgt}(I_i), & \text{if}\, \mathrm{Hgt}(I_i) \geq \lambda \\ 0, & \text{in any other case} \end{cases} \tag{31}$$

where I_i is the fuzzy set for instance $e_i \in E$ from the point of view of superclass instances. Value λ is the limit or minimum threshold to reject instances with a very small membership degree.

b) Option 2:

$$\alpha_i = \begin{cases} \min(1, \mathrm{Card}(I_i)), & \text{if } \mathrm{Card}(I_i) \geq \lambda \\ 0, & \text{in any other case.} \end{cases} \quad (32)$$

c) Option 3:

$$\alpha_i = \begin{cases} 1, & \text{if } \mathrm{Hgt}(I_i) \geq \lambda \\ 0, & \text{in any other case.} \end{cases} \quad (33)$$

d) Option 4:

$$\alpha_i = \begin{cases} 1, & \text{if } \mathrm{Card}(I_i) \geq \lambda \\ 0, & \text{in any other case.} \end{cases} \quad (34)$$

These four options are sufficiently efficient and allow the system to be very flexible. With a fixed λ, we can sort the four options according to the results of the count operation:

$$\text{Option } 1 \leq \{\text{Option } 2, \text{Option } 3\} \leq \text{Option } 4. \quad (35)$$

Options 2 and 3 cannot be sorted, because even though $\mathrm{Card}(I_i) \geq \mathrm{Hgt}(I_i)$, Option 3 adds 1 (if the height is greater than or equal to λ), whereas Option 2 adds a value less than or equal to 1 (if the cardinality is greater than or equal to λ).

2) If a **fuzzy cardinality constraint** exists (only on overlapping specializations), then the DBMS must compute the number of subclasses to which each superclass instance belongs. For example, in order to decide if the number of subclasses of a superclass instance is between "approximately 2" and "approximately half" of the existing subclasses (using fuzzy (min,max) notation). However, this number is not simple as membership is now fuzzy. This problem may be solved in two ways: 1) by using the fuzzy set cardinality [37] of the second point of view fuzzy set or by using generalized measures, such as the fuzzy set energy [31]; or 2) by counting the number of subclasses with a membership degree greater than a minimum value (usually zero). Once the DBMS has computed this number, the system must check if this number satisfies the fuzzy cardinality constraint.

The cardinality of a fuzzy set can be a complex problem and it has been studied by various authors, especially Dubois and Prade [12] and Delgado *et al.* [11]. Nevertheless, in this application efficiency is very important (especially in large databases), but other methods can also be used.

This definition complements the definition of fuzzy types given in [33].

XI. Fuzzy Attribute-Defined Specializations

There are certain kinds of fuzzy attributes, summarized in [19]. Some models [34] and applications [16], [17] use the following ones. The so-called fuzzy attributes Type 1 are totally crisp (traditional), but they have some linguistic trapezoidal labels defined on them, which allow us to make the query conditions for these attributes more flexible (cold, warm, ...). With these attributes, we can use fuzzy queries in classic databases. Fuzzy attributes Type 2 admit crisp or fuzzy data over an ordered underlying domain. Fuzzy attributes Type 3 do not have an ordered underlying domain, for instance, the hair color. On these attributes some labels are defined (fair, brown, red-haired, ...) and on these labels a similarity relation has yet to be defined. Thus, each two labels are equal (or similar) with a similarity degree in [0, 1]. Moreover, fuzzy attributes Type 3 admit fuzzy sets (or possibility distributions) on theirs underlying domains. An example of these fuzzy sets is {1/brown, 0.5/red_haired, 0.2/fair}.

In some contexts, a fuzzy attribute Type 3 does not have a similarity relation defined in its domain. We call these attributes fuzzy attributes Type 4.

Definition 9: A **fuzzy attribute-defined specialization** is exactly the same as an attribute defined specialization in EER models [14] where this attribute is a fuzzy attribute. It is represented with an angled line joining the superclass with the circle. This line will be labeled with the name of fuzzy attribute Type n, preceded by the text "Tn:." □

This constraint establishes that every subclass instance has a valid value (in a certain fuzzy range) for that attribute and according to the subclass. In general, each subclass corresponds with one of the linguistic labels defined on this attribute. Thus, each subclass would be a fuzzy entity, but this is not mandatory. For example, in Fig. 10 the attribute Abilities would be considered as a fuzzy attribute Type 3. It should be noted that this makes it necessary to define a similarity relation on all the subclasses.

This definition is independent of all constraints like fuzzy or crisp disjoint or overlapping specializations. The classification of each instance e of superclass E is then an automatic process, according to the characteristics of the specialization.

- Fuzzy disjoint (fd): Instance e is assigned to one subclass S. Subclass S is the subclass with a greater value of $\mu_S(e)$ (membership degree of e to S). This membership degree is only stored if S is a fuzzy entity.
- Fuzzy overlapping (fo): Instance e is assigned to all subclasses S_i, such that $\mu_{S_i}(e) > 0$. These membership degrees are stored only in the subclasses which are fuzzy entities.
- Nonfuzzy disjoint (d): Instance e is assigned to one subclass S. Subclass S is the subclass with a greater value of $\mu_S(e)$, but this membership degree is not stored and it is considered as 1.
- Nonfuzzy overlapping (o): Instance e is assigned to all subclasses S_i, such that $\mu_{S_i}(e) > 0$, but these membership degrees are not stored and they are considered as 1.

These four cases may be used with the four fuzzy attribute types. Then, 16 different possibilities are produced.

The following example shows two fuzzy attribute-defined specializations (disjoint and overlapping). In one specialization, each pair of subclasses has a fuzzy similarity degree between them (Type 3). This property is useful for comparing them and for searching the more important instances in some queries. In the other specialization there is no similarity relation (Type 4).

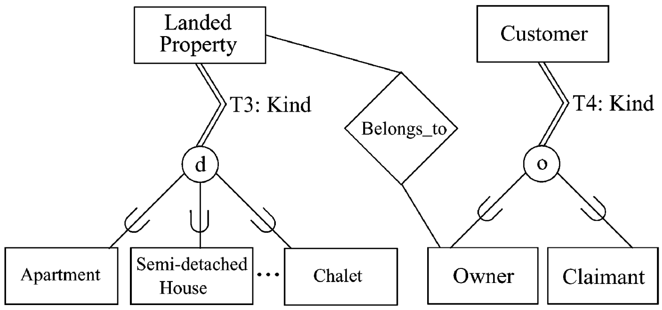

Fig. 11. Example 11: Fuzzy attribute-defined disjoint specialization with total participation constraint.

Example 11: The conceptual model represented in Fig. 11 states that in a real estate agency, every landed property belongs to one subclass, which has its own attributes. Thus, this is a total disjoint specialization (a double line and a "d" inside the circle). The attribute Kind is a fuzzy attribute Type 3, because if one person is looking for a chalet, for example, then this customer is possibly interested in semi-detached houses because these two types are similar. Thus, this is taken into account in order to show all the relevant properties to our customer. In this sense, fuzzy queries are studied in [16]–[18]. It should be noted that the subclasses are not fuzzy, because every landed property only belongs to one subclass.

Every landed property has an owner, who is a customer. Another kind of customer is a claimant who is looking for a landed property. The overlapping specialization results in the fact that one customer may be an owner and a claimant at the same time. The fuzzy attribute Type 4, Kind, allows to store possibility distributions over the subclasses, in order to express any fuzzy concept. In this example we are interested in measuring the urgency of the customer. Thus, a customer with the value $\{0.4/\text{Owner}, 1/\text{Claimant}\}$ is a customer who is urgently looking for a landed property and who is offering some property without urgency. It can be seen that the subclasses are not fuzzy, because a customer is or is not an owner and/or a claimant. □

Example 12: Fig. 13 includes another three examples of fuzzy attribute-defined specializations using two fuzzy overlapping specializations and one disjoint specialization. The first one is a specialization with a total participation constraint (double line) and it establishes that all employees must belong to one or more categories. In addition, Category is a fuzzy attribute Type 3.

The second one is a specialization with a fuzzy participation constraint with the fuzzy quantifier almost_all in the labeled arc: Almost all researchers must belong to one or more research lines. In addition, Research_Line is a fuzzy attribute Type 3. We use a labeled arc instead of a zig-zag line in the fuzzy participation constraint because in this case it is clearer.

The third one is a disjoint specialization with a total participation constraint and it establishes that all temporary employees are beginners or seniors, according to their seniority (or antiquity). Subclasses are not fuzzy because we do not want to store the membership degree. In addition, a temporary employee cannot belong to both subclasses. The antiquity is a crisp and known value but we can make flexible queries using this attribute, i.e., it is a fuzzy attribute Type 1. □

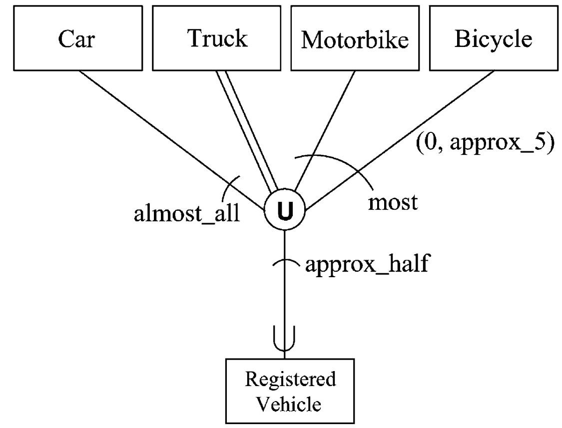

Fig. 12. Example 13: Fuzzy constraints on a Union Type or Category.

XII. Fuzzy Constraints in Union Types or Categories: Participation and Completeness

In the EER model, we can also find the union types or categories [13], [14]. This represents the case when some different superclasses may be members of a special subclass (called category) or not. By definition, each member of the subclass or category must be a member of at least one of the superclasses. Union types are represented with the union symbol inside a circle. Superclasses are joined to that circle by a line. The subclass or category is joined to that circle using a single line with the inclusion symbol. Furthermore, in partial categories it is possible that superclass instances do not belong to the category, because the category is a subset of the union of all superclasses.

It should be noted that the total categories (double line) indicate that all the superclass instances belong to the category. In this case, the category may be represented using a generalization in which the category is transformed into a superclass with total participation constraint.

In this type of specialization, it is possible to apply fuzzy constraints in two ways.

Definition 10: Let C be a category (or subclass) of a union type, with n superclasses: E_i with $i = 1, 2, \ldots, n$. A **fuzzy participation constraint in one or more superclasses** is represented by an arc crossing the lines which join the selected superclasses with the circle. The arc must be labeled with its fuzzy quantifier or with the fuzzy (min,max) notation.

The selected superclasses are those superclasses which are constrained. They are denoted by $E_j, \forall j \in J$ with $J \subseteq \{1, 2, \ldots, n\}$. The union of the selected superclasses is denoted by $\bigtriangledown$

$$\bigtriangledown = \bigcup_{j \in J} E_j. \tag{36}$$

1) If the arc is labeled with the quantifier Q, this constraint establishes (5), with Φ defined by (4), where a is the number of instances in $\bigtriangledown$ which belong to C, and b is the total number of instances in $\bigtriangledown$.

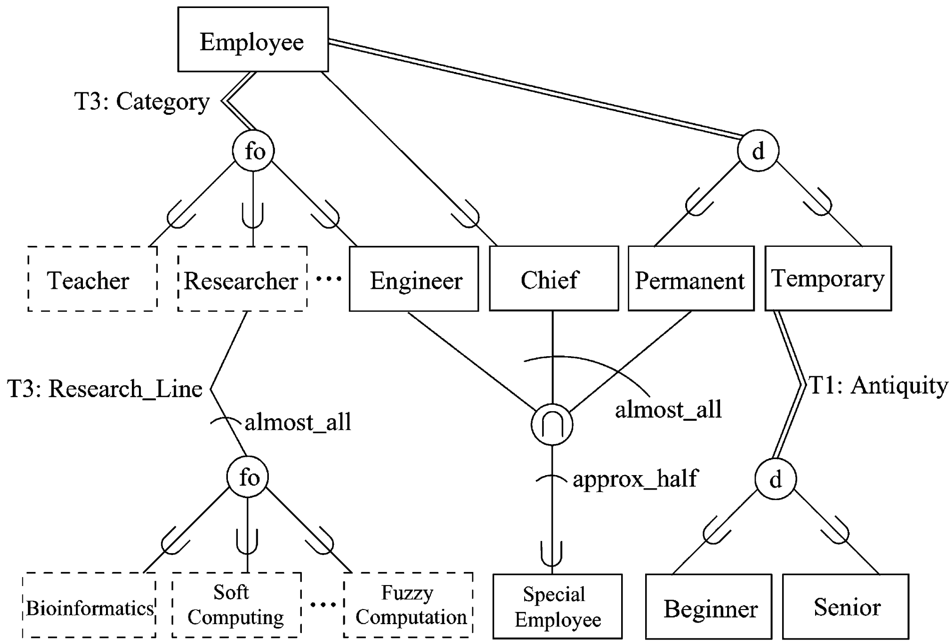

Fig. 13. Examples 12 and 14: Three fuzzy attribute-defined specializations and fuzzy constraints in a shared subclass.

2) If the arc is labeled with the fuzzy (min,max) notation $(Q_{\min}, Q_{\max})$ this constraint establishes that:

$$\lambda_{\min} \leq \Phi_{\min} \wedge \lambda_{\max} \geq \Phi_{\max} \tag{37}$$

where

$$\lambda_{\min} = \min\left\{\alpha : \alpha = Q_{\min}^{-1}(\gamma_{\min})\right\} \tag{38}$$
$$\lambda_{\max} = \max\left\{\beta : \beta = Q_{\max}^{-1}(\gamma_{\max})\right\} \tag{39}$$

where $\gamma_{\min}$ and $\gamma_{\max}$ are the minimum thresholds for $Q_{\min}$ and $Q_{\max}$, respectively, and

$$\Phi_{\min} = \begin{cases} a, & \text{if } Q_{\min} \text{ is absolute} \\ a/b, & \text{if } Q_{\min} \text{ is relative} \end{cases} \tag{40}$$
$$\Phi_{\max} = \begin{cases} a, & \text{if } Q_{\max} \text{ is absolute} \\ a/b, & \text{if } Q_{\max} \text{ is relative} \end{cases} \tag{41}$$

with a and b being the same values defined in previous case.

The warning area is similarly defined using δ_1 and δ_2 respectively. □

This constraint restricts the number of instances (in the union $\bigtriangledown$ of any group of superclasses), which belong to the category. The fuzzy quantifier will normally be relative. For example, with the quantifier "almost all" on one superclass the constraint states that: "almost all the superclass elements belong to the category." Another option is to join two or more superclasses with an arc indicating that the union of instances of these superclasses are constrained in participation. This constraint allows the use of the (min,max) notation indicating the minimum and maximum number of instances in $\bigtriangledown$ which belong to the category (using absolute or relative fuzzy quantifiers), and in this case we must perform the observations appearing in Section VII in order to express a good constraint.

Definition 11: A **fuzzy completeness constraint in the category** (on the union of all superclasses) is represented by an arc crossing the line which joins the category with the circle. The arc is labeled with one fuzzy quantifier, or with the fuzzy (min,max) notation. This constraint is a fuzzy participation constraint (Definition 10) embracing all superclasses: $J = \{1, 2, \ldots, n\}$. □

This constraint restricts the number of instances, of all superclasses (the union), which belong to the category. This fuzzy quantifier will normally be relative. For example, with the quantifier "almost all" on the category, the constraint states that: "almost all elements of all superclasses belong to the category." This constraint also allows the use of the fuzzy (min,max) notation, indicating the minimum and maximum number of all superclass instances which belong to the category. It should be noted that this second way always refers to all the superclasses instances, i.e., to the union of all the superclasses. Consequently, relative fuzzy quantifiers are preferable in this constraint.

Example 13: Let us consider four entity types for vehicles: Car, Truck, Motorbike, and Bicycle. Some vehicles may belong to the Registered Vehicle entity. Fig. 12 depicts this model with some participation constraints: Almost all the cars must be registered vehicles. All the trucks must also be registered. Moreover, the model allows a maximum of approximately five bicycles to be registered vehicles. The arc labeled with the fuzzy quantifier "most" indicates that most motorbikes or bicycles (its union) must be registered.

For the sake of simplicity, we introduce a fuzzy completeness constraint in the same specialization. This constraint establishes that approximately half of the existing vehicles must be registered vehicles. □

In real models, fuzzy constraints in the same specialization must be mixed with care.

XIII. Fuzzy Constraints in Shared Subclasses: Participation and Completeness

A shared subclass (or intersection type) is a subclass with several superclasses [14]. Each member of the subclass must be a member of all the superclasses, i.e., the subclass is a subset of the intersection of all of the superclasses. A shared subclass is represented joining it with all of its superclasses by a single line with the inclusion symbol. Another representation utilizes the intersection symbol inside a circle: Superclasses are joined to that circle by a line and the subclass is joined to that circle using a single line with the inclusion symbol.

As with union types, in this type of specialization it is possible to apply fuzzy constraints in two ways:

Definition 12: Let S be a shared subclass (of an intersection type), with n superclasses: E_i with $i = 1, 2, \ldots, n$. A **fuzzy participation constraint in one or more superclasses** is represented by an arc crossing the lines which join the selected superclasses with the circle. The arc must be labeled with its fuzzy quantifier or with the fuzzy (min,max) notation.

The selected superclasses are those superclasses which are constrained. They are denoted by $E_j, \forall j \in J$ with $J \subseteq \{1, 2, \ldots, n\}$. The intersection of the selected superclasses is denoted by $\triangle$

$$\triangle = \bigcap_{j \in J} E_j. \tag{42}$$

1) If the arc is labeled with the quantifier Q, this constraint establishes (5), with Φ defined by (4), where a is the number of instances in $\triangle$ which belong to S, and b is the total number of instances in $\triangle$.
2) If the arc is labeled with the fuzzy (min,max) notation $(Q_{\min}, Q_{\max})$ this constraint establishes the constraint expressed in (37) with $\Phi_{\min}$ and $\Phi_{\max}$ computed using the values a and b defined in the previous case of this definition.

The warning area is similarly defined using δ_1 and δ_2, respectively. □

This constraint restricts the number of instances, in the intersection of any group of superclasses ($\triangle$), which belong to the shared subclass. This fuzzy quantifier should be relative. For example, with the quantifier "almost all" on one superclass the constraint expresses that: "almost all the superclass elements belong to the shared subclass." Another option is to join two or more superclasses with the arc indicating that the intersection of instances of those superclasses are constrained in participation. This constraint allows the use of the fuzzy (min,max) notation indicating the minimum and maximum number of instances in $\triangle$ which belong to the shared subclass (using absolute or relative fuzzy quantifiers). Generally speaking, the participation constraint is not useful as one constraint on one superclass (or on several superclasses) depends on the membership of its instances to the other superclasses (it should be remembered that the subclass is a subset of the intersection).

Definition 13: A **fuzzy completeness constraint in a shared subclass** (on the intersection of all superclasses) is represented by an arc crossing the line which joins the shared subclass with the circle. The arc is labeled with one fuzzy quantifier, or with the fuzzy (min,max) notation. This constraint is a fuzzy participation constraint (Definition 12) embracing all superclasses: $J = \{1, 2, \ldots, n\}$. □

This constraint restricts the number of instances, in the intersection of all the superclasses which belong to the shared subclass. This fuzzy quantifier will normally be relative. For example, with the quantifier "almost all" on the shared subclass the constraint states that: "almost all the elements of the intersection of all the superclasses belong to the shared subclass." This constraint also allows the fuzzy (min,max) notation to be used, indicating the minimum and maximum number of instances in the intersection (of all the superclasses) which belong to the shared subclass.

Example 14: Let us consider an entity for Special Employees with its own attributes (extra payment, number of awards, motive, ...). A member of this shared subclass must be an engineer, a chief (boss) and a permanent employee. Fig. 13 depicts this model with the following participation constraint: Almost all the chiefs and permanent employees must be special employees. It is interesting to note how this constraint means that almost all the chiefs and permanent employees must also be engineers (because all special employees belong to the engineer superclass).

On the other hand, the fuzzy completeness constraint establishes that approximately half of the employees who are engineers, chiefs and permanent employees must be special employees. □

XIV. Conclusion and Further Research

Fuzzy logic allows us to bring the operation of information systems closer to the working methods of humans. People frequently deal with fuzzy concepts (terms like "almost all," "the majority," "approximately 8," etc.) which include a certain vagueness or uncertainty and which traditional information systems do not understand and therefore cannot use.

Fuzzy databases [17], [34], [38] have also been widely studied with the following main objectives: firstly, to allow imprecise or fuzzy data to be stored, and secondly, to allow the possibility of imprecise or fuzzy queries, using existing data (whether imprecise or not). Traditionally, the application of fuzzy logic to databases has paid little attention to the problem of conceptual modeling [5].

The extension of the ER model for dealing with fuzzy data has been studied in various publications [4], [5], [7], [8], [28], [32], [40], [45], [47], [53], but none of these refers to the possibility of extending constraints by using the tools offered by fuzzy sets theory. Another research line is to achieve notational constructs to allow a greater selection of other fuzzy integrity constraints. For example, relaxing the constraints proposed in [10].

In this paper, we have presented a system for expressing flexible constraints which can be used in a conceptual model utilizing the EER modeling tool [9], [14]. These restrictions can therefore be represented using fuzzy quantifiers [17], [19], [29], [30], [48], [51]. The constraints studied are: the fuzzy participation constraint, the fuzzy cardinality constraint, the fuzzy completeness constraint on specializations, the fuzzy cardinality constraint on overlapping specializations, fuzzy disjoint and fuzzy overlapping constraints on specializations, fuzzy attribute-defined specializations, fuzzy participation and completeness constraints in union types or categories and fuzzy participation and completeness constraints in shared subclasses.

In addition, we have studied the fuzzy (min,max) notation and shown how this notation can substitute fuzzy cardinality constraints and that a fuzzy cardinality constraint can only substitute the (min,max) notation if both quantifiers are of the same type (absolute or relative). Despite this equivalence in the majority of cases, we consider that it is preferable to use the (min,max) notation for greater clarity.

The studied constraints on specializations include and improve the types of constraints proposed in [46] which are not considered in other models [14]. Our proposal improves on these, since it uses the power and flexibility offered by fuzzy sets theory.

The fuzzy constraints have a novel meaning and offer great expressiveness to the conceptual model. Furthermore, the conceptual model continues to be an easy-to-understand system of expression even for nontechnical users, something which is fundamental in conceptual modeling.

This work is integrated with [41], [21], [43], and [44] in a complete fuzzy EER model, the fuzzy EER model. We must also study possible problems and improvements in the resulting model.

An interesting study to facilitate the task of using fuzzy quantifiers on the part of designers would be to classify the quantifiers which can be used in natural language, and study the relationship between them. As previously indicated, one constraint can be established with various fuzzy quantifiers and, in this case, the use of certain quantifiers conditions and limits the possibility of using others in the same constraint.

Other important future lines of works are: 1) to study the repercussions of a fuzzy relationship between two entities with fuzzy constraints; 2) to study the repercussions of the inheritance characteristic with fuzzy entities and constraints; and 3) to relax the universal quantifier which refers to all instances of any entity [for example, in (10), (12), (14), (19), (23), and (26)].

The next step will be to define of the transformation of this fuzzy conceptual model into a fuzzy DBMS. Two main extensions to this fuzzy DBMS must be carried out in order to preserve the semantics of the fuzzy conceptual model: create the necessary elements (e.g., triggers, assertions) in order to assure the fulfilment of the fuzzy constraints and the study of extension to the fuzzy SQL (FSQL) in order to query the stored data in compliance with these fuzzy constraints. FSQL is an extension of the popular SQL which allows for dealing with imprecise data [16], [17].

We are currently working on modeling a real application for a real estate agency, using all these ideas and several new ones, for example the contribution of fuzzy logic in the knowledge management [25], [26]. We started with the definition presented in [18] and one first approach is in [42], [44].

ACKNOWLEDGMENT

The authors would like to thank Prof. L. Jiménez and the anonymous referees for their useful revision and comment on this paper.

REFERENCES

[1] G. Bordogna, D. Lucarella, and G. Pasi, "A fuzzy object-oriented data model managing vague and uncertain information," *Int. J. Intell. Syst.*, vol. 14–7, pp. 623–651, 1999.

[2] B. P. Buckles and F. E. Petry, "Uncertainty models in information and database systems," *Inform. Sci.*, vol. 11, pp. 77–87, 1985.

[3] R. de Caluwe, Ed., *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*. Singapore: World Scientific, 1997.

[4] N. Chaudhry, J. Moyne, and E. A. Rundensteiner, "A design methodology for databases with uncertain data," in *Proc. 7th Int. Working Conf. Scientific Statistical Database Management*, Charlottesville, VA, 1994, www.mitexsolutions.com, pp. 32–41.

[5] ——, "An extended database design methodology for uncertain data management," *Inform. Sci.*, vol. 121, pp. 83–112, 1999.

[6] P. Chen, "The entity-relationship model—Toward a unified view of data," *ACM Trans. Database Systems (TODS) 1*, vol. 1, pp. 9–36, Mar. 1976.

[7] G. Q. Chen and E. E. Kerre, "Extending ER/EER concepts toward fuzzy conceptual data modeling," *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 2, pp. 1320–1325, 1998.

[8] G. Q. Chen, *Fuzzy Logic in Data Modeling: Semantics, Constraints and Database Design*. ser. The Kluwer International Series on Advances in Database Systems, A. K. Elmagarmid, Ed. Norwell, MA: Kluwer, 1998.

[9] T. Connolly and C. Begg, *Database Systems: A Practical Approach to Design, Implementation and Management*. Reading, MA: Addison-Wesley, 1998.

[10] J. P. Davis and R. D. Bonnell, "Modeling semantic constraints with logic in the EARL data model," in *Proc. 5th Int. Conf. Data Engineering*, 1989, pp. 226–233.

[11] M. Delgado, D. Sánchez, and M. A. Vila, "Fuzzy cardinality based evaluation of quantified sentences," *Int. J. Approx. Reason.*, vol. 23, pp. 23–66, 2000.

[12] D. Dubois and H. Prade, "Fuzzy cardinality and the modeling of imprecise quantification," *Fuzzy Sets Syst.*, vol. 16, pp. 190–230, 1985.

[13] R. Elmasri, J. Weeldreyer, and A. Hevner, "The category concept: An extension to the entity-relationship model," *Int. J. Data Knowledge Eng.*, vol. 1:1, May 1985.

[14] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 3rd ed. Reading, MA: Addison-Wesley, 2000.

[15] I. Fujishiro *et al.*, "The design of a graph-oriented schema for the management of individualized fuzzy data," *Jpn. J. Fuzzy Theory Syst.*, vol. 3, no. 1, pp. 1–14, 1991.

[16] J. Galindo, J. M. Medina, O. Pons, and J. C. Cubero, "A server for fuzzy SQL queries," in *Flexible Query Answering Systems*, T. Andreasen, H. Christiansen, and H. L. Larsen, Eds. New York: Springer-Verlag, 1998, pp. 164–174. Lecture Notes in Artificial Intelligence (LNAI) 1495.

[17] J. Galindo, "Tratamiento de la imprecisión en bases de datos relacionales: Extensión del modelo y adaptación de los SGBD actuales," Ph.D. dissertation, Univ. Granada, Granada, Spain, Mar. 1999, www.lcc.uma.es.

[18] J. Galindo, J. M. Medina, J. C. Cubero, and O. Pons, "Management of an estate agency allowing fuzzy data and flexible queries," in *Proc. EUSFLAT-ESTYLF Joint Conf.*, Palma de Mallorca, Spain, Sept. 1999, pp. 485–488.

[19] J. Galindo, J. M. Medina, J. C. Cubero, and M. T. García, "Relaxing the universal quantifier of the division in fuzzy relational databases," *Int. J. Intell. Syst.*, vol. 16, no. 6, pp. 713–742, 2001.

[20] J. Galindo, A. Urrutia, R. Carrasco, and M. Piattini, "Fuzzy constraints using the enhanced entity-relationship model," in *Proc. XXI Int. Conf. Chilean Computer Science Soc. (SCCC)*, Punta Arenas, Chile, Nov. 2001, pp. 86–94.

[21] J. Galindo and A. Urrutia, "Fuzzy extensions to EER specializations," in *Proc. 8th CAiSE/IFIP8, Int. Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD'03)*, Velden, Austria, June 2003, pp. 218–227.

[22] R. George, R. Srikanth, F. E. Petry, and B. P. Buckles, "Uncertainty management issues in the object-oriented data model," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 179–192, Apr. 1996.

[23] N. Van Gyseghem, R. De Caluwe, and R. Vandenberghe, "UFO: Uncertainty and fuzziness in an object-oriented model," in *Proc. IEEE 2nd Int. Conf. Fuzzy Systems*, vol. 2, San Francisco, CA, Mar. 1993, pp. 773–778.

[24] M. Hammer and D. McLeod, "Database description with SDM: A semantic data model," *ACM Trans. Database Systems (TODS)*, vol. 6, no. 3, pp. 351–386, Sept. 1981.

[25] L. Jiménez, "Gestion des Connaissances: CommonKADS étendue aux données imprécises et incertaines," in *École Doctorale Systèmes, 3e Congrès des Doctorants*, Blagnac, France, May 2002, pp. 108–113.

[26] L. Jiménez and A. Urrutia, "Extensión del Conocimiento del Dominio de CommonKADS con Lógica Difusa," in *Proc. 5th Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes Software (IDEAS 2002)*, Havana, Cuba, Apr. 2002.

[27] E. E. Kerre and G. Q. Chen, "An overview of fuzzy data models," in *Studies in Fuzziness: Fuzziness in Database Management Systems*, P. Bosc and J. Kacprzyk, Eds. Heidelberg, Germany: Physica-Verlag, 1995, pp. 23–41.

[28] E. E. Kerre and G. Chen, "Fuzzy data modeling at a conceptual level: Extending ER/EER concepts," in *Knowledge Management in Fuzzy Databases*, O. Pons, M. A. Vila, and J. Kacprzyk, Eds. Heidelberg, Germany: Physica-Verlag, 2000, pp. 3–11.

[29] Y. Liu and E. E. Kerre, "An overview of fuzzy quantifiers. (I). Interpretations," *Fuzzy Sets Syst.*, vol. 95, no. 1, pp. 1–21, 1998.

[30] ——, "An overview of fuzzy quantifiers. (II). Reasoning and applications," *Fuzzy Sets Syst.*, vol. 95, no. 2, pp. 135–146, 1998.

[31] A. De Luca and S. Termini, "Entropy of L-fuzzy sets," *Inform. Control*, vol. 24, pp. 55–73, 1974.

[32] Z. M. Ma, W. J. Zhang, W. Y. Ma, and G. Q. Chen, "Conceptual design of fuzzy object-oriented databases using extended entity-relationship model," *Int. J. Intell. Syst.*, vol. 16, no. 6, pp. 697–711, 2001.

[33] N. Marín, O. Pons, and M. A. Vila, "Fuzzy types: A new concept of type for managing vague structures," *Int. J. Intell. Syst.*, vol. 15, pp. 1061–1085, 2000.

[34] J. M. Medina, O. Pons, and M. A. Vila, "GEFRED. A generalized model of fuzzy relational databases," *Inform. Sci.*, vol. 76, no. 1–2, pp. 87–109, 1994.

[35] A. Motro, "Accommodating imprecision in database systems: Issues and solutions," *SIGMOD Record*, vol. 19, no. 4, pp. 69–74, Dec. 1990.

[36] ——, "Management of uncertainty in database system," in *Modern Database System the Object Model, Interoperability and Beyond*, W. Kim, Ed. Reading, MA: Addison-Wesley, 1995.

[37] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge, MA: MIT Press, 1998.

[38] F. E. Petry, *Fuzzy Databases: Principles and Applications*. ser. International Series in Intelligent Technologies, H.-J. Zimmermann, Ed. Norwell, MA: Kluwer, 1996.

[39] E. Rundensteiner and L. Bic, "Semantic database models and their potential for capturing imprecision," presented at the Conf. Management Data, COMAD'89, Hyderabad, India, 1989.

[40] E. Ruspini, "Imprecision and uncertainty in the entity-relationship model," in *Fuzzy Logic in Knowledge Engineering*, H. Prade and C. V. Negoita, Eds. Berlin, Germany: Verlag TUV Rheinland, 1986, pp. 18–22.

[41] A. Urrutia, J. Galindo, and L. Jiménez, "Representación de Información Imprecisa en un Modelo Conceptual EER Difuso," in *Proc. VIII Congreso Internacional de Investigación en Ciencias Computacionales (CIICC'01)*, Colima, México, Nov. 2001, pp. 15–27.

[42] A. Urrutia and J. Galindo, "Algunos Aspectos del Modelo Conceptual EER Difuso: Aplicación al Caso de una Agencia Inmobiliaria," in *XI Congreso Español sobre Tecnologías y Lógica Fuzzy (ESTYLF'2002)*, León, Spain, Sept. 2002, pp. 359–364.

[43] A. Urrutia, J. Galindo, and M. Piattini, "Modeling data using fuzzy attributes," in *Proc. XXII Int. Conf. Chilean Computer Science Soc.*, Nov. 2002, pp. 117–123.

[44] A. Urrutia, "Definición de un modelo conceptual para bases de datos difusas," Ph.D. dissertation, University of Castilla-La Mancha, Ciudad Real, Spain, July 2003.

[45] R. M. Vandenberghe, "An extended entity-relationship model for fuzzy databases based on fuzzy truth values," in *Proc. 4th Int. Fuzzy Systems Association World Congr. IFSA'91*, Brussels, Belgium, 1991, pp. 280–283.

[46] M. Varas, R. Contreras, and D. Campos, "Constraints in generalization structures in conceptual database schemes," presented at the Conf. Int. de la Sociedad Chilena de Ciencia de la Computación, SCCC'98 Antofagasta, Concepción, Chile, 1998, www.inf.udec.cl/~mvaras.

[47] G. Vert, A. Morris, M. Stock, and P. Jankowski, "Extending entity-relationship modeling notation to manage fuzzy datasets," in *Proc. 8th Int. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'2000*, Madrid, Spain, July 2000, pp. 1131–1138.

[48] R. R. Yager, "Quantified propositions of a linguistic logic," *Int. J. Man–Machine Stud.*, vol. 19, pp. 195–227, 1983.

[49] A. Yazici and R. George, *Fuzzy Database Modeling*. New York: Springer-Verlag, 1999.

[50] A. Yazici, B. P. Buckles, and F. E. Petry, "Handling complex and uncertain information in the exifo and NF^2 data models," *IEEE Trans. Fuzzy Syst.*, vol. 7, pp. 659–676, Dec. 1999.

[51] L. A. Zadeh, "A computational approach to fuzzy quantifiers in natural languages," *Comput. Math. Applicat.*, vol. 9, pp. 149–183, 1983.

[52] L. A. Zadeh, "Fuzzy sets," in *Inform. Control*, vol. 8, 1965, pp. 338–353.

[53] A. Zvieli and P. Chen, "Entity-relationship modeling and fuzzy databases," in *Proc. 2nd IEEE Int. Conf. Data Engineering*, 1986, pp. 320–327.

José Galindo received the Ph.D. degree in computer science from the University of Granada, Granada, Spain, in 1999.

He is currently a Professor of Computer Science with the School of Engineering, University of Málaga, Málaga, Spain. He is the author of several books and papers on computer science, databases, information systems, and fuzzy logic. His research interests include fuzzy logic, fuzzy databases, and ethical issues in the technological age. He is a Member of IDBIS Research Group and RITOS-2 Ibero-American Research Net.

Angélica Urrutia received the B.S. and M.S. degrees from Concepcion University and Santiago University, Chile, respectively, and the Ph.D. in computer science (*Sobresaliente Cum Laude*) from Castilla-La Mancha University, Ciudad Real, Spain, in 2003.

She is currently an Associate Professor with the Computer Science Department of Maule Catholic University, Chile. She is member of the Chilean Computer Science Society and RITOS-2 (Red Iberoamericana de Tecnologías del Software para la Década del 2000) working group of CYTED. She is the Founder and President of the Chilean Workshop on Databases. Her major research topics are fuzzy databases and infomation systems. In these areas, she has authored several original scientific papers.

Ramón A. Carrasco received the Ph.D. degree (*Sobresaliente Cum Laude*) from Granada University, Granada, Spain, in 2003.

He currently leads the Department of Knowledge Management (Data Warehouse: OLAP, Data Minig, and ETL) at the Spanish savings bank CajaGranada. He is the author of several papers on databases, data mining, information systems, and fuzzy logic. His research interests include data mining, fuzzy logic, neural networks, and fuzzy databases. He is a Member of the IDBIS Research Group.

Mario Piattini received the M.Sc. and Ph.D. degrees in computer science from the Politechnical University of Madrid, Madrid, Spain, in 1994.

He is a Certified Information Systems Auditor and Certified Information System Manager under the Information System Audit and Control Association (ISACA). He is currently a Full Professor with the Department of Computer Science at the University of Castilla-La Mancha, Ciudad Real, Spain. He is the author of several books and papers on databases, software engineering, and information systems. He leads the ALARCOS Research Group of the Department of Computer Science at the University of Castilla-La Mancha. His research interests include advanced database design, database quality, software metrics, object oriented metrics, and software maintenance.