Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 4598–4620
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Database design and querying within the fuzzy semantic model

Rafik Bouaziz [a,1], Salem Chakhar [b,*], Vincent Mousseau [b,2], Sudha Ram [c,3], Abdelkader Telmoudi [d,4]

[a] Department of Computer Science, Faculty of Economic Sciences and Management, Route de l'Aérodrome, BP 1088, 3018 Sfax, Tunisia
[b] LAMSADE, University of Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, Paris, France
[c] Department of MIS, 430J McClelland Hall, Eller College of Management, University of Arizona, Tucson, AZ 85721, USA
[d] Department of Industrial Engineering, National School of Engineering of Tunis, B.P. 37 Le Belvédère 1002 Tunis, Tunisia

Received 20 January 2005; received in revised form 19 April 2007; accepted 6 May 2007

Abstract

Fuzzy semantic model (FSM) is a data model that uses basic concepts of semantic modeling and supports handling fuzziness, uncertainty and imprecision of real-world at the attribute, entity and class levels. The paper presents the principles and constructs of the FSM. It proposes ways to define the membership functions within all the constructs of the FSM. In addition, it provides a proposal for specifying FSM schema and introduce a query language adapted to FSM-based databases.

Keywords: Database design; Fuzzy database; Semantic modeling; Flexible querying

1. Introduction

In database research, there are several proposals to develop models that support handling fuzziness, uncertainty and imprecision of real-world [24,2,4]. Most efforts have been oriented towards the extension of the conventional relational database model [20,17,10] and towards the development of tools allowing flexible querying, most often in relational database contexts [23,5,19]. We also enumerate some extensions of semantic and object-oriented database models [21,3,8,16,14]. However, most of these extensions introduce fuzziness only at the attribute level and consider that entities are fully "encapsulated" into their classes, which means that they fully verify the properties of these classes. This is very restrictive in many data-intensive applications

* Corresponding author. Tel.: +33 1 44 05 49 18; fax: +33 1 44 05 41 11.
E-mail addresses: raf.bouaziz@fsegs.rnu.tn (R. Bouaziz), salem.chakhar@dauphine.fr (S. Chakhar), mousseau@lamsade.dauphine.fr (V. Mousseau), ram@eller.arizona.edu (S. Ram), abdelkader.telmoudi@enit.rnu.tn (A. Telmoudi).
[1] Tel.: +216 74 55 54 44; fax: +216 74 77 88 99.
[2] Tel.: +33 1 44 05 41 84; fax: +33 1 44 05 41 11.
[3] Tel.: +1 520 621 2748; fax: +1 520 621 2433.
[4] Tel.: +216 71 874 700; fax: +216 71 872 729.

0020-0255/$ - see front matter
doi:10.1016/j.ins.2007.05.013

(e.g. geographical and environmental information systems, decision support systems) in which one may find it difficult to assign an entity to a particular class, mainly when this entity partially verifies the class properties.

Several proposals for extending object-oriented and semantic database models to support the management of fuzziness, uncertainty and imprecision of real-world at the class definition level have been recently proposed [12,9,13,25,22,15]. In this respect, the authors have proposed a new data model, namely the fuzzy semantic model (FSM) [6,7], that authorizes an entity to be, albeit partially, a member of its class according to a given degree of membership. The latter reflects the level to which the entity verifies the properties of this class.

The basic constructs of FSM are extensions of the unifying semantic model (USM) [18]. USM is selected as basis of FSM instead of other ER/EER models for two main reasons. First, USM draws upon constructs found in several other semantic models and we believe that USM synthesizes and extends these constructs in a coherent manner. Second, in addition to the traditionally used abstractions of classification, generalization, aggregation and association, USM proposes concepts to represent constraints on relationships between subclasses and also distinguishes between the concepts of composite and group/aggregate classes. We think that these new concepts cope better with real-world semantics.

This paper reviews and refines FSM. More specifically, it presents ways to define membership functions within all the constructs of FSM. In addition, it provides a proposal to specify FSM schema and introduces a query language adapted to FSM-based databases. The paper includes several illustrative examples most of them rely on the database example illustrated in Fig. 3. Readers are invited to refer frequently to this figure to better appreciate these examples.

The rest of the paper is structured as follows. The next section presents the principles of FSM. Section 3 details the constructs of FSM. Section 4 provides a proposal to specify FSM schema. Section 5 presents an ongoing conceptual query language for accessing FSM-based databases and illustrates some examples of data retrieval operations. Section 6 compares our proposal to some other fuzzy semantic data models. Section 7 concludes the paper.

2. Basic elements of FSM

2.1. Basic idea

The *space of entities* E is the set of all entities of the interest domain. A *fuzzy entity* e in E is a natural or artificial entity such that one or several of its properties are fuzzy. In other words, a fuzzy entity verifies only partially some *extent properties* (see Section 2.2) of its class. A *fuzzy class* K in E is a collection of fuzzy entities: $K = \{(e, \mu_K(e)) : e \in E \wedge \mu_K(e) > 0\}$. μ_K is a characteristic or *membership function* and $\mu_K(e)$ represents the *degree of membership* (d.o.m.) of fuzzy entity e in fuzzy class K. Membership function μ_K maps the elements of E to the range $[0, 1]$ where 0 implies no-membership and 1 implies full membership. A value between 0 and 1 indicates the extent to which fuzzy entity e can be considered as an element of fuzzy class K.

FSM contains several basic and complex fuzzy classes that are illustrated in Fig. 2. They will be discussed in Section 3. First, we show how the entity/class membership function is defined.

2.2. Entity/class membership function

A fuzzy class is a collection of fuzzy entities having some common properties. Fuzziness is thus induced whenever an entity verifies only partially some of these properties. We denote by $X_K = \{p_1, p_2, \ldots, p_n\}$ (with $n \geqslant 1$) the set of properties for a given fuzzy class K. X_K is called the *extent set* of fuzzy class K and $p_i \in X_K (i = 1, \ldots, n)$ is an *extent property* associated with K. The extent properties may be derived from the attributes of the class and/or from common semantics. For example, the fuzzy class STAR in Fig. 3 may have two extent properties based on *luminosity* and *weight* attributes. The degree to which each of the extent properties determines fuzzy class K is not the same. Indeed, there are some properties that are more discriminative than others. To ensure this, we associate to each extent property p_i a non-negative weight w_i reflecting its importance in deciding whether or not an entity e is a member of a given fuzzy class K. We also impose that $\sum_{i=1}^{n} w_i > 0$.

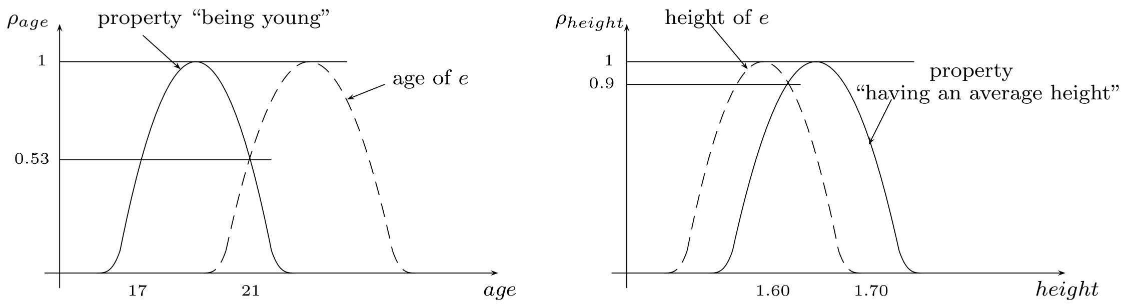

Fig. 1. Fuzzy properties "being young" and "having an average height".

An entity may verify fully or partially the extent properties of a given fuzzy class. Let D^i be the basic domain of extent property p_i values and P^i is a subset of D^i, which represents the set of possible values of property p_i. The *partial membership function* of an extent property value is $\rho_{P^i_K}$ which maps elements of D^i into $[0,1]$. For any attribute value $v_i \in D^i$, $\rho_{P^i_K}(v_i) = 0$ means that fuzzy entity e violates property p_i and $\rho_{P^i_K}(v_i) = 1$ means that this entity verifies fully the property. The number v_i is the value of the attribute of entity e on which the property p_i is defined. For extent properties based on common semantics, v_i is a semantic phrase and the partial d.o.m. $\rho_{P^i_K}(v_i)$ is supposed to be equal to 1 but the user may explicitly provide a value less than 1. More generally, the value of $\rho_{P^i_K}(v_i)$ represents the level to which entity e verifies property p_i of fuzzy class K. Thus, the *global* d.o.m. of fuzzy entity e in fuzzy class K is

$$\mu_K(e) = \frac{\sum_{i=1}^{n} \rho_{P^i_K}(v_i) \cdot w_i}{\sum_{i=1}^{n} w_i}. \quad (1)$$

Suppose that the fuzzy class YOUNG of young persons is defined through the attributes *age* and *height*. Accordingly, the extent set of this class is $X_{\text{Young}} = \{p_1, p_2\}$, where p_1 and p_2 properties are defined respectively on the *age* and *height* attributes. Clearly, the *age* attribute is more relevant in defining a young person. However, in many situations, it is not possible to determine the exact age of that person and the *height* attribute will be a good indicator. To ensure this, we assign to p_1 and p_2 the weights of $w_1 = 0.8$ and $w_2 = 0.3$, respectively. Now, suppose that we aim to calculate the global d.o.m. of a person e in the fuzzy class YOUNG. The two fuzzy properties of "being young" and "having an average height" are shown in Fig. 1. The exact age and height of e are not known but we suppose that they are as represented in Fig. 1. In this figure, it is easy to see that $\rho_{P^1_{\text{Young}}}(e.age) = 0.53$ and $\rho_{P^2_{\text{Young}}}(e.height) = 0.9$. Thus by applying Eq. (1), we get: $\mu_{\text{Young}}(e) = 0.630$.

To define the d.o.m. of an object in its class, the authors in [25] use a weighted sum of the inclusion degrees of the attribute values in the attribute ranges as they are defined at the class level. They use the relevance of attributes to classes as weights. In the proposal of [15], the authors use a weighted sum of the inclusion degrees of the attribute values in the attribute domains. They use the importance of attributes to classes as weights. The inclusion degrees are computed differently in these two proposals. In FSM, we use the partial d.o.m. instead of the inclusion degrees. The weights in the three proposals have similar interpretations. However, in [25,15] all the attributes of the class are used to compute the d.o.m. (although, one can give a weight of zero to one or several attributes to eliminate them from consideration) but in FSM only a subset of the attributes are used.

3. Constructs of FSM

3.1. Basic classes

A fuzzy class is a semantic collection of fuzzy entities. Each class has a list of characteristics or properties, called attributes. Each of these attributes takes its values in a *domain class* (Fig. 2c). Some of these attributes are used to construct the extent set X_K defined earlier. To be a member of a fuzzy class K, a fuzzy entity e must verify (fully or partially) at least one of the extent properties of K, i.e., $\mu_K(e) > 0$.

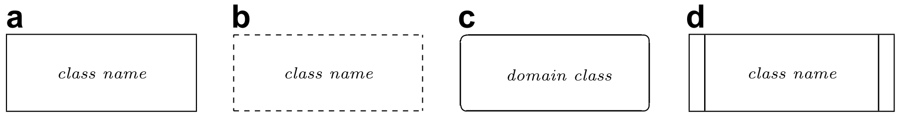

Fig. 2. FSM class symbols.

The classes in FSM may be simple (Fig. 2a) or complex (Fig. 2d). In addition, they are categorized as exact or fuzzy. An *exact class K* is a class that all its members have a d.o.m. equal to 1. A *fuzzy class K* is a class such that at least one of its members has a d.o.m. strictly inferior to 1. Furthermore, classes may also be categorized as strong or weak. A *strong fuzzy class* (Fig. 2a) is a fuzzy class whose members can exist on their own, i.e., they are not depending on other classes. A *weak fuzzy class* (Fig. 2b) is a fuzzy class whose members depend on the existence of other classes for their existence.

3.2. Members

The elements of a fuzzy class are called *members*. In FSM, α-MEMBERS denotes, for a given fuzzy class K, the set $\{e : e \in K \wedge \mu_K(e) \geqslant \alpha\}$ where $\alpha \in [0, 1]$. For instance, 0.17-MEMBERS of fuzzy class STAR is $\{e : e \in \text{STAR} \wedge \mu_{\text{Star}}(e) \geqslant 0.17\}$. The 1-MEMBERS may also be referred to *true* or *exact members*. In turn, α-MEMBERS with $0 < \alpha < 1$ are called *fuzzy members*. The concept of α-MEMBERS may be mapped to the concept of α-cut associated with fuzzy sets and which is defined for a fuzzy subset F as the set $F_\alpha = \{x : \mu_F(x) \geqslant \alpha\}$ with $0 \leqslant \alpha \leqslant 1$.

3.3. Relationships

3.3.1. Property relationships

A *property relationship* associates a fuzzy class with a domain class. Each property relationship defines an attribute (see Section 3.4). We distinguish two types of attributes: (i) *simple attributes*, which are defined by themselves (Fig. 4a), and (ii) *derived attributes*, which are obtained from other attributes (Fig. 4b). For example, *luminosity*, *weight* and *star-name* are three simple attributes which we may associate with the fuzzy class STAR in the database example of Fig. 3. We may also associate a derived attribute *age* with the class PERSON based on the *date-of-birth* attribute.

3.3.2. Decision rule relationships

To implement the extent sets associated with fuzzy classes, two new relationships are introduced in FSM formalism. The first is an *attribute-based decision rule relationship* (Fig. 4c) used to decide (through a binary comparison, for instance) whether or not a fuzzy entity is a member of a given class. The fuzzy class STAR in Fig. 3 may, for instance, have two decision rule relationships based on *luminosity* and *weight* attributes, respectively. The second is a *semantic decision rule relationship* (Fig. 4d), which is a semantic phrase used to specify the members of a specific class. Semantic decision rule relationships are mainly useful to define exact classes. For example, the class PERSON in Fig. 3 may be defined as "a set of persons".

Any basic fuzzy class must have at least one decision rule. In turn, complex (or non-basic) fuzzy classes may (e.g. attribute-defined fuzzy composite classes; see Section 3.7) or may not require (e.g. fuzzy interaction classes; see Section 3.3.4) decision rule relationships.

3.3.3. Membering relationships

The membering relationships relate fuzzy entities to fuzzy classes. Two types of membering relationships are defined: *true* (or *exact*) (Fig. 4e) and *fuzzy* (Fig. 4f) *membering relationships*. All these relationships are normally binary. However, in generalization/specialization relationships an entity may be—through the inheritance mechanism—a member of several fuzzy superclasses at the same time with different membership degrees.

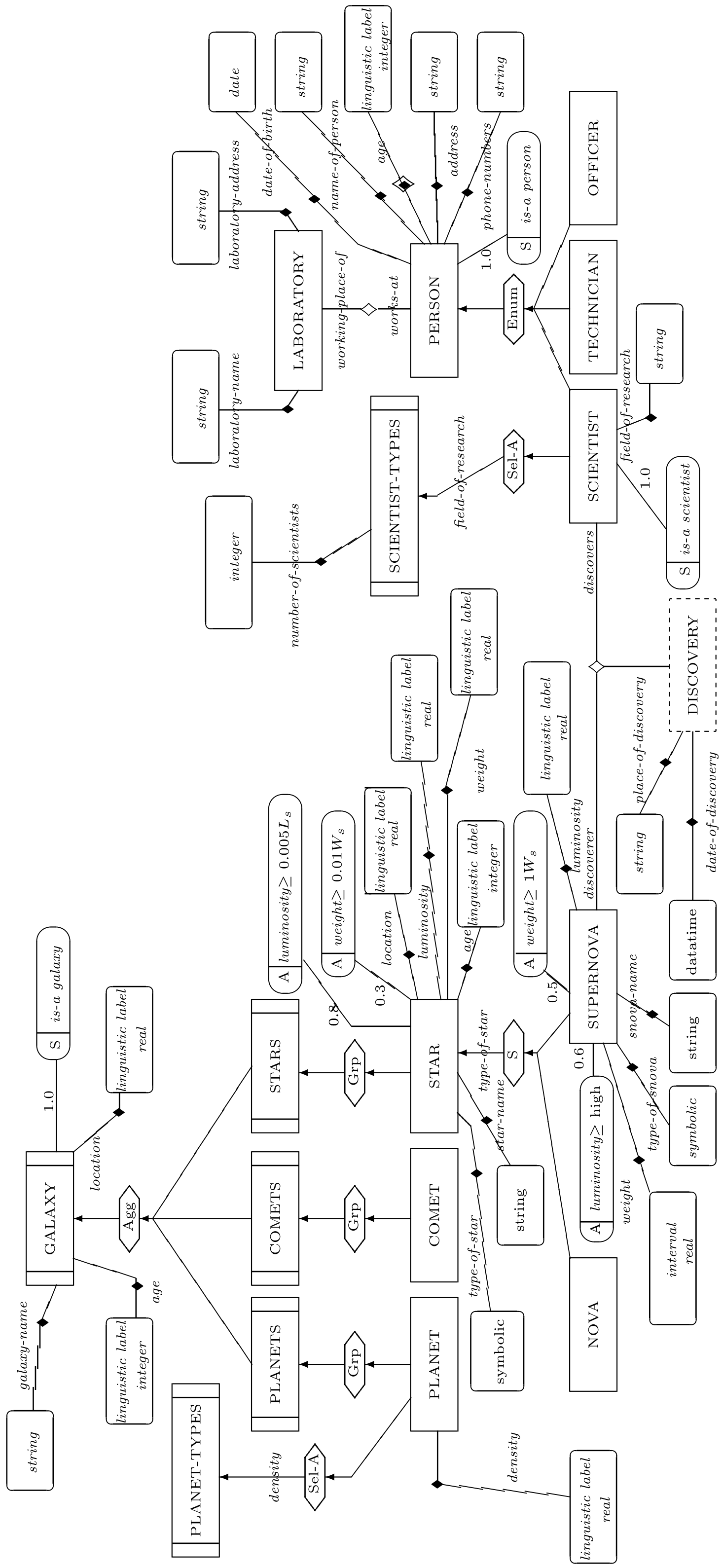

Fig. 3. Example of a FSM-based model.

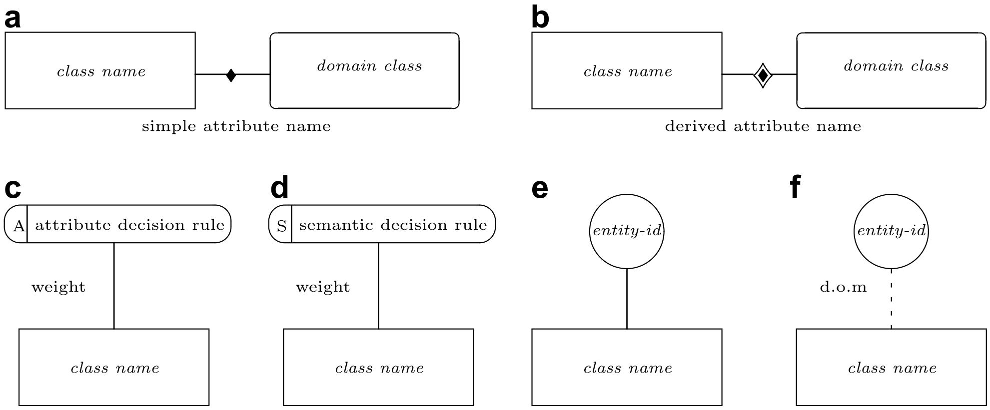

Fig. 4. FSM relationships.

3.3.4. Interaction relationships

An *interaction* (or *association*) relationship relates members of one fuzzy class to other members of one or many fuzzy classes. There are two types of interaction relationships: *binary interaction relationship* and *n-ary interaction relationship*. The binary interaction relationship relates two fuzzy classes. In addition, in binary interaction relationships two attributes are created, each one is the *inverse* of the other. For example, the binary interaction relationship relating SUPERNOVA and SCIENTIST classes in Fig. 3 requires the creation of two attributes, namely *discoverer* from the point of view of SUPERNOVA and its inverse attribute *discovers* from the point of view of SCIENTIST. The *n*-ary interaction relationship relates at least three members from three fuzzy classes.

The interaction relationship may (Fig. 5b) or may not require (Fig. 5a) the creation of new attributes that describe the interaction relationship. In the former case, a new (obligatory weak) *fuzzy interaction class* is generated. For instance, each member of the DISCOVERY fuzzy interaction class in Fig. 3 associates one member (may be several members when the discovery is accomplished by several scientists) from SCIENTIST class with one member from SUPERNOVA fuzzy class. This relationship may be further described with two attributes, *date-of-discovery* and *place-of-discovery*, that permit to handle some information concerning the date and the place of the discovery. An interaction relationship may also relate one member to other members of the same fuzzy class and forms thus a *reflexive* (or *recursive*) *interaction relationship* (Fig. 5c).

The fuzzy interaction class should not have extent properties since its members are fully defined in terms of the extent properties of the participant fuzzy classes. However, the d.o.m. of a member e of a fuzzy interaction class I relating m members $e_1, e_2, \ldots, e_m$ from m fuzzy classes $K_1, K_2, \ldots, K_m$ may be calculated as follows:

$$\mu_I(e) = \prod_{i=1}^{m} \mu_{K_i}(e_i). \tag{2}$$

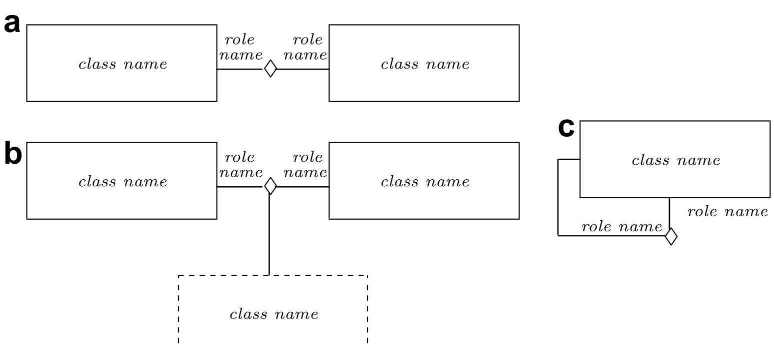

Fig. 5. FSM interaction relationships.

3.4. Attributes

Each property relationship relates a fuzzy class to a domain class. This relationship creates an *attribute* associated with the members of the fuzzy class. Attributes may also be created through interaction, composition or grouping relationships. Attributes may be *single* or *multi-valued*. Values may be crisp or fuzzy.

In FSM, it is not necessary that a fuzzy subclass inherits all the attributes of its fuzzy superclass. Attributes that are not obligatory are said to be *non-relevant*. The others are the *relevant* ones. In other terms, relevant attributes are neither common to all the fuzzy subclasses to be included in their (common) fuzzy superclass and inherited by all of them nor specific to only one of them to be included only in it.

Through the inheritance concept associated with subclass/superclass relationships, a class inherits all relevant attributes of its superclass. These attributes are seen to be attributes of both the subclass and the superclass. As in [11], an attribute is said to be an *immediate attribute* of the base class it is declared in. The authors in [11] also distinguish two types of attributes: *data-valued attributes* and *entity-valued attributes*. The data-valued attributes are equivalent to the attributes as presented above. The entity-valued attributes are specific, non-printable, binary relationships that describe the properties of each entity of a class by relating it to an entity (or entities) of another (or the same) class. In database literature, entity-valued attributes are also called *roles*. For example, *discoverer* and *discovers* are two entity-valued attributes associated with the fuzzy interaction relationship between SUPERNOVA and SCIENTIST. The concept of entity-valued attributes that is adopted here is mainly useful for data manipulation and retrieval operations.

3.5. Fuzzy classes relationships

FSM also supports two types of inter-classes relationships: specialization and generalization. The *specialization relationship* relates a fuzzy superclass to one or several simple or complex fuzzy subclasses. Such a relation advocates that all the members of the fuzzy subclass are members of its fuzzy superclass. Any specialization relationship creates implicitly a *generalization relationship*, which relates a fuzzy subclass to a fuzzy superclass. The same superclass may have one, two or more subclasses (e.g. class PERSON in Fig. 3) and the same fuzzy subclass may have more than one fuzzy superclass.

A fuzzy subclass may be attribute-defined, roster-defined or set-operation-defined. An attribute-defined fuzzy subclass (Fig. 6a) has one or several attribute values that are in accordance with some discriminative values which characterize perfectly its members. For instance, the fuzzy subclasses NOVA and SUPERNOVA in Fig. 3 are specializations of the fuzzy class STAR based on the attribute *type-of-star*. The attribute-defined fuzzy subclasses inherit all relevant attributes of their fuzzy superclasses.

A roster-defined fuzzy subclass is simply defined by an explicit enumeration of its members (Fig. 6b). These subclasses inherit all relevant attributes of their superclasses. For instance, SCIENTIST, TECHNICIAN and OFFICER in Fig. 3 are three roster-defined subclasses of PERSON.

A set-operation-defined fuzzy subclass may be defined as the set-difference (Fig. 6c) or the set-intersection (Fig. 6d) of two or more fuzzy classes. Members of a difference fuzzy subclass of two fuzzy superclasses are the fuzzy entities which are members of the first fuzzy superclass that are not members of the second one. The set-difference fuzzy subclass inherits only relevant attributes of the first fuzzy superclass. Members of a set-inter-

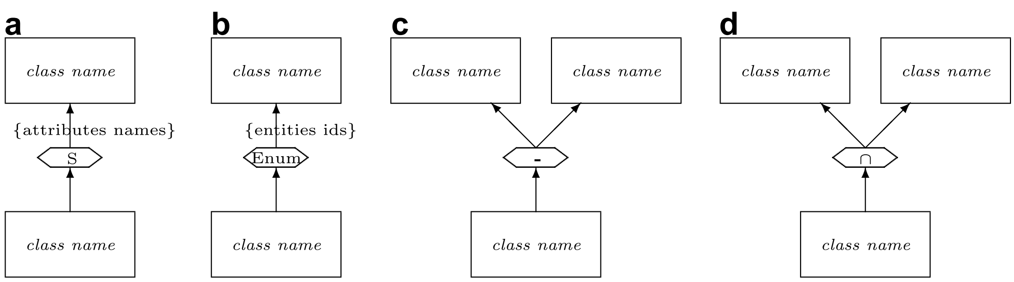

Fig. 6. FSM class relationships.

section fuzzy subclass of two or several fuzzy superclasses are members of each of these fuzzy superclasses. The set-intersection fuzzy subclass inherits all relevant attributes of all the participant fuzzy superclasses.

Fuzzy subclasses as well as superclasses have their own extent properties and the d.o.m. of their members may be calculated through Eq. (1). In [15], the authors distinguish two types of object/class relationships in object-oriented databases. The first is a *direct objet–class relationship* which applies when the object and the class have the same attributes. The second is an *indirect objet–class relationship* and is specific for subclass/superclass relationships where an object belonging to the subclass must belong to the superclass since a subclass is a specialization of the superclass. The authors propose a formula to calculate the d.o.m. of a member of the subclass in the superclass. The idea may be adapted to our FSM as follows. Let S_1 be a subclass of S_2. The *inheritance* concept associated with subclass/superclass relationships advocates that S_1 inherits some attributes from S_2, overrides some others, and adds some new ones. Then, let S_2 has the extent properties set $X_{S_2} = \{p_1, p_2, \ldots, p_k, \ldots, p_{k+1}, \ldots, p_m\}$ and S_1 has the extent properties set $X_{S_1} = \{p_1, p_2, \ldots, p_k, \ldots, p'_{k+1}, \ldots, p'_m, p_{m+1}, \ldots, p_n\}$ where $p'_{k+1}, \ldots, p'_m$ are overridden from $p_{k+1}, \ldots, p_m$ (i.e. for all $i = k+1$ to m, p'_i is based on the same attribute on which p_i is based) and $p_{m+1}, \ldots, p_n$ are specific for S_1. Thus, the d.o.m. of an entity e from fuzzy subclass S_1 in fuzzy superclass S_2 of S_1, written $\mu_{S_2}(e/S_1)$, is

$$\mu_{S_2}(e/S_1) = \frac{\sum_{i=1}^{k} \rho_{P^i}(v_i) \cdot w_i + \sum_{j=k+1}^{m} \rho_{P'^j}(v'_j) \cdot w'_j}{\sum_{i=1}^{k} w_i + \sum_{j=k+1}^{m} w'_j}, \tag{3}$$

where, for $i = 1$ to k, $P^i \subset D^i$ represents the set of possible values of extent property p_i, w_i is the weight of p_i and v_i is the value of the attribute of entity e on which property p_i is based; and for $j = k+1$ to m, $P'^j \subset D^j$ (note that p_i and p'_i should have the same domain) represents the set of possible values of extent property p'_j, w'_j is the weight of p'_j and v'_i is the value of the attribute of fuzzy entity e on which property p'_j is based.

For example, suppose that $X_{\text{Star}} = \{p_1, p_2\}$ where extent properties p_1 and p_2 are based on *luminosity* and *weight* attributes, respectively; and $X_{\text{Supernova}} = \{p'_1, p'_2, p_3\}$ where extent properties p'_1, p'_2 and p_3 are based on *luminosity*, *weight* and *age* attributes, respectively. Note that in the schema definition examples that will be introduced in Section 4 and provided in Appendix A, SUPERNOVA has only two extent properties which are based on *luminosity* and *weight* attributes. The *age* attribute is added here to illustrate how subclass/superclass d.o.m. is computed. Let $w_1 = w'_1 = 0.7$, $w_2 = w'_2 = 0.3$ and $w_3 = 0.2$. Next, suppose that a member e exists in SUPERNOVA for which $\rho_{P^1_{\text{Star}}}(e.luminosity) = 0.56$ and $\rho_{P^2_{\text{Star}}}(e.weight) = 0.60$; and $\rho_{P'^1_{\text{Supernova}}}(e.luminosity) = 0.35$, $\rho_{P'^2_{\text{Supernova}}}(e.weight) = 0.34$ and $\rho_{P^3_{\text{Supernova}}}(e.age) = 0.12$. Then, the d.o.m. of e in STAR is

$$\mu_{\text{Star}}(e/Supernova) = 0.347.$$

We notice that by applying Eq. (1), we get $\mu_{\text{Supernova}}(e) = 0.309$ which is inferior to $\mu_{\text{Star}}(e/Supernova) = 0.347$. This is in accordance with the rule proposed by several authors (e.g. [14,15,13]) and which postulates the fact that in fuzzy subclass/superclass relationships, the d.o.m. of an entity/object to a fuzzy subclass should be less or equal to its d.o.m. to the fuzzy superclass of this subclass. More generally, any entity e member of a fuzzy subclass S_1 of a fuzzy superclass S_2 should verify $\mu_{S_1}(e) \leqslant \mu_{S_2}(e)$.

3.6. Subclass/superclass membership function

In [15] the authors extend the notion of membership function to the subclass/superclass relationships in object-oriented database models. To calculate the d.o.m. of a subclass in a superclass, they use a weighted sum of the inclusion degrees of the attribute domains of the subclass in the attribute domains of the superclass. In this paper, we adopt a similar way. The only difference is that we will use the d.o.m. of entities relatively to their direct classes as weights. Formally, the d.o.m. of a fuzzy subclass S_j in its fuzzy superclass S_i, written $\mu(S_i, S_j)$, is equal to:

$$\mu(S_i, S_j) = \frac{\sum_{e_\alpha \in S_j} \mu_{S_i}(e_\alpha/S_j) \cdot \mu_{S_j}(e_\alpha)}{\sum_{e_\alpha \in S_j} \mu_{S_j}(e_\alpha)}, \tag{4}$$

where $\mu_{S_j}(e_\alpha)$ is the d.o.m. of fuzzy entity e_α in fuzzy class S_j calculated as in Eq. (1) and $\mu_{S_i}(e_\alpha/S_j)$ represents the d.o.m. of fuzzy entity e_α from S_j in S_i calculated as in Eq. (3).

For example, suppose that the fuzzy subclass SUPERNOVA contains three entities e_1, e_2 and e_3 with $\mu_{\text{Supernova}}(e_1) = 0.55$, $\mu_{\text{Supernova}}(e_2) = 1.0$ and $\mu_{\text{Supernova}}(e_3) = 0.67$. We suppose also that these entities verify the following:

- $\mu_{\text{Star}}(e_1/\text{Supernova}) = 0.37$.
- $\mu_{\text{Star}}(e_2/\text{Supernova}) = 0.15$.
- $\mu_{\text{Star}}(e_3/\text{Supernova}) = 0.90$.

Then, Eq. (4) gives: $\mu(\text{Star}, \text{Supernova}) = 0.430$.

The d.o.m. in subclass/superclass relationships as calculated here differs from the one proposed in [15,25] in the sense that it depends on the entities currently present in the database. This means that in FSM the value of $\mu(S_i, S_j)$ may evolve over time and it is not "static" as in [15] or [25].

3.7. Fuzzy composite classes

A fuzzy composite relationship defines a new fuzzy class that has other classes as its members. All such classes are strong. Specifically, a member of a *fuzzy composite class* is the set of members of some other fuzzy classes taken as a whole. The fuzzy classes that are members may be subclasses of a common fuzzy superclass, in such a case they are said to be *homogeneous*, or they may not, in such a case they are said to be *heterogeneous*. Each fuzzy composite class has a multi-valued attribute called *contents*, which permits to identify all of its members.

Fuzzy composite classes are needed to define *class attributes* or properties that describe a whole class rather than each individual entity in a class. To better appreciate the concept of composite class and its utility, consider the example shown in Fig. 7. In this example, a fuzzy composite class PLANET-TYPES is defined on the fuzzy class PLANET. PLANET-TYPES has three members: c_1, c_2 and c_3. These members result from the grouping of the members of PLANET according to the value of attribute *density*, called *selection attribute* (see Section 3.7.1). Without fuzzy composite class PLANET-TYPES, there is no way to handle the class attribute *number-of-planets*. In fact, it is not possible to include this attribute in fuzzy class PLANET. This example will be continued below (in Section 3.7.1).

There are two types of fuzzy composite classes: attribute-defined (Fig. 8a) and enumerated (Fig. 8b). They are discussed hereafter.

3.7.1. Attribute-defined fuzzy composite classes

Given a fuzzy class C with some attributes $a_1, a_2, \ldots, a_n$, we can define an attribute-defined fuzzy composite class based on C as having as its members those subclasses of the fuzzy class C in which all members have identical values for the attributes $a_1, a_2, \ldots, a_n$. That is, we can envision a set of attribute-defined fuzzy subclasses of C defined by the tuple of values $(a_1, a_2, \ldots, a_n)$ and the fuzzy composite class has those fuzzy subclasses as its members. The attributes $a_1, a_2, \ldots, a_n$ are called the *selection attributes* of the fuzzy composite class. Attribute-defined fuzzy composite classes are necessarily homogeneous ones. In the example introduced above, PLANET-TYPES is defined on the fuzzy class PLANET. This composition is based on *density* attribute. The three members of PLANET-TYPES are denoted c_1, c_2 and c_3, respectively. Each member of PLANET-TYPES is itself a class. In Fig. 7, the classes associated with c_1, c_2 and c_3 are called TERRESTRIAL, PLANETOIDS and GAS-GIANTS, respectively. Each of these fuzzy classes consists of a number of members, each of which is a member of the class PLANET. The composite class PLANET-TYPES really consists of a number of classes each of which is a subclass of the class PLANET.

The d.o.m. in attribute-defined fuzzy composite classes is computed as follows. Let C be an attribute-defined fuzzy composite class based on the selection attributes $a_1, a_2, \ldots, a_n$. The extent properties set of fuzzy composite class C is based on all or a subset of the selection attributes: $X_C = \{p_1, p_2, \ldots, p_q\}$ with $q \leqslant n$ and $p_1, p_2, \ldots, p_q$ are based on attributes $a_1, a_2, \ldots, a_q$, respectively. As mentioned earlier, members of an attribute-defined fuzzy composite class C are themselves fuzzy classes $K_1, K_2, \ldots, K_p$ that are subclasses of the same

PLANET (S)

entity-id	*name*	*density*	$\mu_S(e_i)$
e_1	Mercury	High	1.0
e_2	Uranus	Low	0.8
e_3	Earth	High	1.0
e_4	Mars	High	1.0
e_5	Saturn	Low	0.9
e_6	Pluto	Average	0.6
e_7	Venus	High	1.0
e_8	Neptune	Low	0.8
e_9	Charon	Average	0.6
e_{10}	Jupiter	Low	0.9

Composite class PLANET-TYPES (C)

entity-id	*class-name*	*density*	*number-of-planets*	$\mu(c_i) =$
c_1	TERRESTRIAL	High	4	$\mu(C, K_1) \cdot \mu(S, K_1)$=0.75
c_2	PLANETOIDS	Average	2	$\mu(C, K_2) \cdot \mu(S, K_2)$=0.5
c_3	GAS-GIANTS	Low	4	$\mu(C, K_3) \cdot \mu(S, K_3)$=0.65

TERRESTRIAL (K_1)

entity-id	*name*	*density*	$\mu_{K_1}(e_i)$	$\mu_S(e_i/K_1)$	$\mu_C(e_i/K_1)$
e_1	Mercury	High	1.0	1.0	0.75
e_3	Earth	High	1.0	1.0	0.75
e_4	Mars	High	1.0	1.0	0.75
e_7	Venus	High	1.0	1.0	0.75

PLANETOIDS (K_2)

entity-id	*name*	*density*	$\mu_{K_2}(e_i)$	$\mu_S(e_i/K_2)$	$\mu_C(e_i/K_2)$
e_6	Pluto	Average	0.6	1.0	0.5
e_9	Charon	Average	0.6	1.0	0.5

GAS-GIANTS (K_3)

entity-id	*name*	*density*	$\mu_{K_3}(e_i)$	$\mu_S(e_i/K_3)$	$\mu_C(e_i/K_3)$
e_2	Uranus	Low	0.8	1.0	0.65
e_5	Saturn	Low	0.9	1.0	0.65
e_8	Neptune	Low	0.8	1.0	0.65
e_{10}	Jupiter	Low	0.9	1.0	0.65

Fig. 7. Example of fuzzy composite class definition.

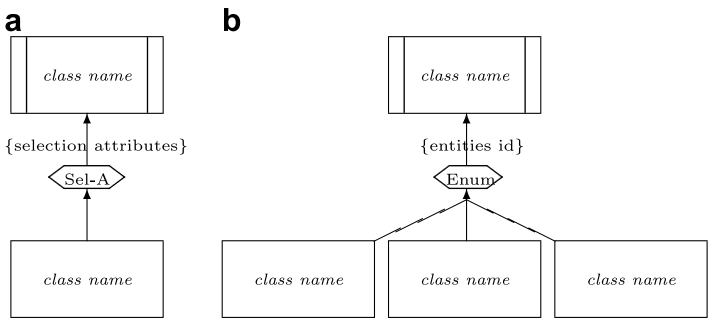

Fig. 8. FSM composite classes.

fuzzy superclass S. This means that each member e_i of C is in relation with all the members of one class, say K_i. Thus, since at least a subset of the selection attributes are common to the fuzzy composite class and to its members, the d.o.m. of e_i in C may be calculated as follows:

$$\mu_C(e_i) = \mu(C, K_i) \cdot \mu(S, K_i), \tag{5}$$

where $\mu(C, K_i)$ and $\mu(S, K_i)$ are the d.o.m. of the class K_i in fuzzy composite class C and in its fuzzy superclass S, respectively. They may be computed in a similar way to Eq. (4), that is

$$\mu(C, K_i) = \frac{\sum_{e_\alpha \in K_i} \mu_C(e_\alpha / K_i) \cdot \mu_{K_i}(e_\alpha)}{\sum_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)}, \tag{6}$$

and

$$\mu(S, K_i) = \frac{\sum_{e_\alpha \in K_i} \mu_S(e_\alpha / K_i) \cdot \mu_{K_i}(e_\alpha)}{\sum_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)}. \tag{7}$$

Note that $\mu_C(e_\alpha/K_i)$ and $\mu_S(e_\alpha/K_i)$ are computed in a similar way to Eq. (3). Let first explain Eq. (5). The first term in this equation ensures that all the members of the class K_i in relation with entity e_i of C are included in the computation of $\mu_C(e_i)$. The second term is used since members of K_i are initially members of S and then, the d.o.m. of K_i in S should also be included. On the other hand, because all the members of a fuzzy subclass of an attribute-defined fuzzy composite class share exactly the same values for the selection attributes on which extent properties set is based, they belong to the fuzzy composite class with the same d.o.m. (this is explained in the illustrative example that follows). This means that in Eq. (6) above, $\mu_C(e_\alpha/K_i)$ is the same for all e_α in K_i. Let for all e_α in K_i, $\mu_C(e_\alpha/K_i) = x_i$ with $x_i \in [0, 1]$. Then, Eq. (6) above will be as follows:

$$\mu(C, K_i) = \frac{x_i \cdot \sum_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)}{\sum_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)} = x_i. \tag{8}$$

Consequently, Eq. (5) becomes

$$\mu_C(e_i) = x_i \cdot \mu(S, K_i). \tag{9}$$

Illustrative example

Consider again the example of Fig. 7. First we comment the d.o.m. of the entities of the fuzzy class PLANET. Planets Mercury, Venus, Earth and Mars, also known as rocky planets, are composed primarily of rock and metal and have very high densities. They are also the closet to the Sun and are relatively well known. They take a d.o.m. equal to 1.0. The planets Jupiter, Saturn, Uranus and Neptune, also known as gas giants, are composed mainly of hydrogen and helium gases and have low densities. These four outer planets in our solar system are less known than the four previous ones and they take a d.o.m. less than 1.0 (in function of their distance from the Sun). The last two planets, Pluto and its moon Charon, are among thousands of objects orbiting in the outer regions of our solar system. They are too large to be asteroids and too small to be planets. They have an average density. The relatively small value of their d.o.m. is explained by (i) the importance of their distance from the Sun and (ii) the fact that many astronomers believe that Pluto and Charon may actually be Kuiper Belt objects and not planets. Kuiper Belt is a disk-shaped region past the orbit of Neptune extending roughly from 30 to 50 AU from the Sun containing many small icy bodies (AU is the abbreviation of astronomical unit and is a unit of length, which is approximately equal to the mean distance between the Earth and Sun). As mentioned above, the definition of fuzzy composite class PLANET-TYPES is based on *density* attribute. Then, $X_{\text{PLANET-TYPES}} = \{p\}$ where p is an extent property based on *density* attribute. Since the possible values for *density* attribute are "high", "average" and "low", PLANET-TYPES has three members c_1, c_2 and c_3. Each of these entities is in fact a subset of the entities of PLANET. As shown in Fig. 7, the classes corresponding to entities c_1, c_2 and c_3 are called TERRESTRIAL, PLANETOIDS and GAS-GIANTS, respectively. It is easy to see that all the members of each of these classes share the same value for the attribute *density*.

For the sake of clarity, in the rest of this example, fuzzy classes PLANET and PLANET-TYPES are denoted S and C, respectively. Equally, we denote the three subclasses TERRESTRIAL, PLANETOIDS and GAS-GIANTS by K_1, K_2 and K_3, respectively. These notations are also used in Fig. 7. Now, we show how $\mu_C(c_1)$ can be computed based on Eqs. (5)–(9). According to Eq. (5), we have:

$$\mu_C(c_1) = \mu(C, K_1) \cdot \mu(S, K_1).$$

By using Eq. (7) and the data of Fig. 7, we get: $\mu(S, K_1) = 1$ (explanation of this result is provided in the end of this example). Consider now the computing of $\mu(C, K_1)$. By using Eq. (6) we have:

$$\mu(C, K_1) = \frac{\mu_C(e_1/K_1) \cdot \mu_{K_1}(e_1) + \mu_C(e_3/K_1) \cdot \mu_{K_1}(e_3) + \mu_C(e_4/K_1) \cdot \mu_{K_1}(e_4) + \mu_C(e_7/K_1) \cdot \mu_{K_1}(e_7)}{\mu_{K_1}(e_1) + \mu_{K_1}(e_3) + \mu_{K_1}(e_4) + \mu_{K_1}(e_7)}.$$

Next, it is easy to see that entities e_1, e_3, e_4, and e_7 of K_1 have exactly the same value for the attribute *density*. Then, basing on Eq. (3), it comes that $\mu_C(e_1/K_1)$, $\mu_C(e_3/K_1)$, $\mu_C(e_4/K_1)$ and $\mu_C(e_7/K_1)$ must have the same value (which is equal to 0.75 in Fig. 7). This holds since the extent property set X_C of composite fuzzy class PLANET-TYPES is based on attribute *density* for which entities e_1, e_3, e_4, and e_7 of subclass K_1 share the same value (which is equal to High in Fig. 7). Accordingly, we get:

$$\begin{aligned}\mu(C, K_1) &= \frac{0.75 \cdot \mu_{K_1}(e_1) + 0.75 \cdot \mu_{K_1}(e_3) + 0.75 \cdot \mu_{K_1}(e_4) + 0.75 \cdot \mu_{K_1}(e_7)}{\mu_{K_1}(e_1) + \mu_{K_1}(e_3) + \mu_{K_1}(e_4) + \mu_{K_1}(e_7)} \\ &= 0.75 \cdot \frac{\mu_{K_1}(e_1) + \mu_{K_1}(e_3) + \mu_{K_1}(e_4) + \mu_{K_1}(e_7)}{\mu_{K_1}(e_1) + \mu_{K_1}(e_3) + \mu_{K_1}(e_4) + \mu_{K_1}(e_7)} = 0.75.\end{aligned}$$

Using similar reasoning, we obtain $\mu_C(c_2) = 0.5$ and $\mu_C(c_3) = 0.65$. To conclude this paragraph, we comment some of the data provided in Fig. 7. First, we remark that for all $e_i \in K_j$, we have $\mu_S(e_i) = \mu_{K_j}(e_i)$ $(i = 1, \ldots, 10)$ $(j = 1, 2, 3)$. In general, this holds when:

- the set of selection attributes and the attributes used to define the extent set of the basic class (S in this example) are exactly the same, and
- the extent set of the basic class and the subclasses (K_1, K_2 and K_3 in this example) are the same.

Both conditions hold in our example. This means that the definition of the properties ‘‘have a high density’’, ‘‘have an average density’’ and ‘‘have a low density’’ associated with subclasses K_1, K_2 and K_3, respectively, are the same as the ones associated with S, i.e., $X_S = X_{K_j}$ $(j = 1, 2, 3)$. Second, it is easy to see that for $e_i \in K_j$, we have $\mu_S(e_i/K_j) = 1$ $(i = 1, \ldots, 10)$ $(j = 1, 2, 3)$. This holds since $X_S = X_{K_j}$ $(j = 1, 2, 3)$. This also explains why $\mu(S, K_1) = \mu(S, K_2) = \mu(S, K_3) = 1.0$.

3.7.2. Enumerated fuzzy composite classes

An enumerated fuzzy composite class is defined by listing its members, that is, by naming as its members other classes which appear in the model. These members may be homogeneous or heterogeneous. Each member of an enumerated fuzzy composite class may have its own attributes in addition to the common ones, if any.

Two cases hold for computing the d.o.m. of enumerated fuzzy composite class members, along with the fact that they are homogeneous or heterogeneous:

- *Homogeneous enumerated fuzzy composite class*. The extent properties set of an enumerated fuzzy composite class C of s homogeneous fuzzy subclasses $K_1, K_2, \ldots, K_s$ is $X_C = \bigcap_{i=1}^{s} X_{K_i}$. Then, the d.o.m. $\mu_C(e_i)$ of an entity e_i in C is computed in similar way to the case of an attribute-defined fuzzy composite class (see Section 3.7.1). This is because we are sure that in this case $X_C \neq \emptyset$ and Eqs. (5)–(9) still apply.
- *Heterogeneous enumerated fuzzy composite class*. Let C be a fuzzy composite class of s heterogeneous fuzzy subclasses $K_1, K_2, \ldots, K_s$. The d.o.m. of a member e_i from C is computed as follows:

$$\mu_C(e_i) = \mu(C, K_i) \cdot \mu(S_i, K_i). \tag{10}$$

Note that S_i is the fuzzy superclass of K_i and that $\mu(S_i, K_i)$ and $\mu(C, K_i)$ are computed as in Eq. (4). Since $K_1, K_2, \ldots, K_s$ are heterogeneous fuzzy classes (they have different fuzzy superclasses), the composite fuzzy class C has no extent properties set. Thus, it is not possible to use Eq. (3) to compute $\mu_C(e_j/K_i)$. In addition, the definition of the extent set of C as $X_C = \bigcap_{i=1}^{s} X_{K_i}$ may provide $X_C = \emptyset$. To avoid these problems, we propose to use Eq. (11)—instead of Eq. (3)—to compute $\mu_C(e_j/K_i)$:

$$\mu_C(e_j/K_i) = \frac{\mu_{K_i}(e_j) - \min_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)}{\max_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha) - \min_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)}, \quad (11)$$

where $\min_{e_\alpha \in K_i} \mu_{K_i}(e_j)$ and $\max_{e_\alpha \in K_i} \mu_{K_i}(e_\alpha)$ are, respectively, the minimum and maximum values for $\mu_{K_i}(e_j)$. This new formula works whether X_C is empty or not. It gives 1 for entities e_j having the maximum value for $\mu_{K_i}(e_j)$ and 0 for those having the minimum value for $\mu_{K_i}(e_j)$. The values for the other entities will be in $]0,1[$.

3.8. Fuzzy grouping classes

A *fuzzy grouping class* is a collection of members from other fuzzy classes. We may distinguish two types of fuzzy grouping classes: aggregate or grouping. A member of a *fuzzy aggregate class* is a heterogeneous collection from different fuzzy classes in which each member (or *aggregate*) is composed of exactly one member from each of the fuzzy classes that are called *components* (Fig. 9b). In other words, members of a fuzzy aggregate class are (a subset of) the Cartesian product of the members of its components. A *fuzzy grouping class* is a homogeneous collection of members (or *groups*) from the same fuzzy class that is called *component* (Fig. 9a). In both cases, members of the fuzzy grouping or aggregate class are unique collections of the component class(es). In other words, the addition or the elimination of one member from the collection creates a new group or a new aggregate. For example, GALAXY is a fuzzy aggregate class whose members are unique collections of members from COMETS, STARS and PLANETS fuzzy grouping classes. These last ones are homogeneous collections of members from strong fuzzy classes COMET, STAR and PLANET, respectively. Finally, we mention that each fuzzy grouping or aggregate class has a multi-valued attribute called *contents* that refers to the members of each of its groups or aggregates.

The extent properties set of a fuzzy aggregate class is the union of the extent properties sets of its components. Mathematically, $X_A = \bigcup_{i=1}^{m} X_{K_i} = \{p_1^1, \ldots, p_1^{n_1}, \ldots, p_k^1, \ldots, p_k^{n_k}, \ldots, p_m^1, \ldots, p_m^{n_m}\}$ where A is an aggregation of m fuzzy classes $K_1, K_2, \ldots, K_m$ and n_k for $k = 1$ to m is the number of extent properties for fuzzy class K_k. Then, the d.o.m. of an entity e of an aggregate fuzzy class A that maps to m entities $e_1, e_2, \ldots, e_m$ of m fuzzy classes $K_1, K_2, \ldots, K_m$ is calculated as follows:

$$\mu_A(e) = \frac{\sum_{\alpha=1}^{m} \sum_{s=1}^{n_\alpha} \rho_{P^s_{K_\alpha}}(v_s^\alpha) \cdot w_s^\alpha}{\sum_{\alpha=1}^{m} \sum_{s=1}^{n_\alpha} w_s^\alpha}, \quad (12)$$

where w_s^α and v_s^α (with $s = 1, \ldots, n_\alpha$ and $\alpha = 1, \ldots m$) denote the weight of extent property number s, p_α^s, of fuzzy class K_α and the value of the attribute on which extent property p_α^s is based, respectively. The d.o.m. of an entity e_i from a fuzzy subclass K_i in the fuzzy aggregate class A, written $\mu_A(e_i/K_i)$, is computed in similar way to Eq. (11), i.e.:

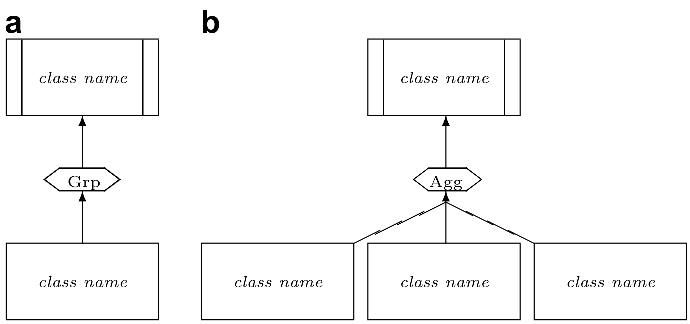

Fig. 9. FSM grouping classes.

$$\mu_A(e_i/K_i) = \frac{\mu_{K_i}(e_j) - \min_{e_\alpha \in K_i}\mu_{K_i}(e_\alpha)}{\max_{e_\alpha \in K_i}\mu_{K_i}(e_\alpha) - \min_{e_\alpha \in K_i}\mu_{K_i}(e_\alpha)}. \quad (13)$$

Then, the d.o.m. of fuzzy subclass K_i in the fuzzy aggregate class A is computed as follows:

$$\mu(A, K_i) = \frac{\sum_{e_\alpha \in K_i}\mu_A(e_\alpha/K_i) \cdot \mu_{K_i}(e_\alpha)}{\sum_{e_\alpha \in K_i}\mu_{K_i}(e_\alpha)}. \quad (14)$$

A fuzzy grouping class is a homogeneous collection of members from the same fuzzy class. The extent properties set of a fuzzy grouping class G is equal to the one of the fuzzy subclass K on which G is based, i.e., $X_G = X_K$. The d.o.m. of an entity e of a fuzzy grouping class G that groups m entities $e_1, e_2, \ldots, e_m$ from fuzzy class K is computed as follows:

$$\mu_G(e) = \frac{\sum_{i=1}^{m}\mu_K(e_i)}{m}. \quad (15)$$

Other formulae as $\mu_G(e) = \max_{1 \leqslant i \leqslant m}\mu_K(e_i)$ or $\mu_G(e) = \min_{1 \leqslant i \leqslant m}\mu_K(e_i)$ may also apply.

For each entity e in the component fuzzy class K that belongs to the fuzzy grouping class G, we have $\mu_G(e/K) = \mu_K(e)$. Then, the d.o.m. of fuzzy class K in G is $\mu(G, K) = 1$.

4. Schema definition in FSM

This section provides a proposal for specifying schema of FSM-based databases. All examples of this section rely on the database example of Fig. 3. In the example database, GALAXY is an aggregate fuzzy class whose members are unique collections of members from COMETS, STARS and PLANETS fuzzy grouping classes. The latter ones are homogeneous collections of members from strong fuzzy classes COMET, STAR and PLANET, respectively. NOVA and SUPERNOVA are two attribute-defined fuzzy subclasses of STAR based on *type-of-star* attribute. PLANET-TYPES is an attribute-defined fuzzy composite class. This composition is from PLANET fuzzy class based on the *density* attribute. PERSON is an exact class. It has three enumerated subclasses: SCIENTIST, TECHNICIAN and OFFICER. Each person is affiliated with at least one LABORATORY. SCIENTIST is a collection of scientists and DISCOVERY is an interaction fuzzy class between SUPERNOVA and SCIENTIST. SCIENTIST-TYPES is a fuzzy composite class from SCIENTIST based on *field-of-research* attribute.

In the generic definitions below we adopt the following conventions:

- [ ]: optional parameter(s).
- { }: list of parameters or values.
- |: the equivalent of the binary operator "xor".
- ⟨⟩: obligatory parameter(s).
- ( ): series of parameters connected with the "xor" operator.

The generic definition of a fuzzy class in FSM is as follows:

CLASS⟨*class-name*⟩ WITH DOM OF ⟨*dom*⟩
{
SUPERCLASS:
OF ⟨*class-name*⟩ WITH DOM OF ⟨*dom*⟩
...
INTERACTION CLASS OF ⟨*class-list*⟩

EXTENT:
⟨*ext-pr*⟩ WITH WEIGHT OF ⟨*w*⟩ DECISION RULE IS ((⟨*attr-name*⟩⟨*op*⟩ (⟨*attr-name*⟩ |⟨*value*⟩))|⟨*op*⟩ ⟨*sphrase*⟩)

...
ATTRIBUTES:
⟨*attr-name*⟩: [FUZZY] DOMAIN ⟨*domain*⟩: TYPE OF ⟨*type*⟩ WITH DOM OF ⟨*dom*⟩: [REQUIRED] [UNIQUE] [MULTI-VALUED]
...
CONTENTS:
[ENUMERATED COMPOSITION FROM (⟨*class-name:members-list*⟩)]
[SELECTED COMPOSITION ON ATTRIBUTES ⟨*attr-list*⟩ FROM ⟨*class-list*⟩]
[AGGREGATION OF (⟨*class-name:members-list*⟩)]
[GROUPING FROM ⟨*class-name:members-list*⟩]

INTERACTION:
⟨*inter-name*⟩ WITH (⟨*class-name*⟩ INVERSE IS ⟨*inv-inter-name*⟩ |⟨*inter-class-list*⟩) [CLASS IS ⟨*inter-class-name*⟩]
...
}

The SUPERCLASS component of the fuzzy class definition enumerates all the subclasses of the class along with their d.o.m. relatively to this class. This component is omitted if the fuzzy class has no fuzzy subclasses. The INTERACTION CLASS OF component is for fuzzy interaction classes only. It permits to specify the list of the participant classes for which the interaction class is defined. One, two and at least three class names are required for recursive, binary, and *n*-ary ($n \geqslant 3$) fuzzy interaction classes, respectively. Next in the EXTENT part, we list all the extent properties of the class. For each extent property, we indicate the name, the weight and the decision rule on which this extent property is based. As it is quoted earlier, decision rules may be attribute-based or semantic phrase-based. The left-side of the attribute-based decision rule indicates the attribute name on which the rule is based. The *op* operator may be a scalar comparator or a set-operator. The right-side of the attribute-based decision rule may be a crisp (e.g. $age = 21$) or fuzzy (e.g. $age =$ young) value. For semantic phrase-based decision rules, the *op* is an "is-a" operator and the right-side is a semantic phrase (e.g. the decision rule "is-a person" may be associated with the class PERSON in Fig. 3). The semantic phrase-based decision rules are optional—but recommended to make the database schema more comprehensible. For instance, we may have the following extent properties definitions:

p_1 WITH WEIGHT OF 0.8 DECISION RULE IS *luminosity* = very high
p_2 WITH WEIGHT OF 0.3 DECISION RULE IS *weight* in $[0.01W_s - 1W_s]$
p_3 WITH WEIGHT OF 0.5 DECISION RULE IS *age* = young
p_3' WITH WEIGHT OF 0.5 DECISION RULE IS *age* in [17–21]
p_4 WITH WEIGHT OF 1.0 DECISION RULE IS is-a galaxy
p_5 WITH WEIGHT OF 1.0 DECISION RULE IS is-a person

The symbol "W_s" above is the weight of the Sun; it is often used as a measurement unit. The four first decision rules are attribute-based ones while decision rules p_4 and p_5 are semantic phrase-based ones. These two last ones may be associated with classes GALAXY and PERSON, respectively.

In the ATTRIBUTES component, we specify the list of the attributes of the fuzzy class. We note that attributes definition schema is partially inspired from [15]. This definition of attributes apply for both exact and fuzzy ones. An exact attribute requires the definition of a datatype (e.g. integer, string) and a domain as a range of possible values for the attribute. A fuzzy attribute requires the definition of a fuzzy type and a fuzzy domain. The fuzzy types are based on simple (e.g. integer) or complex types (e.g. set-valued types, entity-valued attributes) that allow the representation of imprecise information. Fuzzy domains may be represented simply as a list of fuzzy linguistic terms (e.g. young, near). Other ways may also apply as for example possibility theory (e.g. the age of a young person may be represented through a possibility distribution as *age* = 0.1/17 + 1.0/18 + 0.2/19). In addition, attributes may be specified as required, unique or multi-valued. All required and unique attributes may serve as *identifiers*. For example, we may have the following declarations of attributes:

location: FUZZY DOMAIN {*in, near, very near, distant, very distant*}: TYPE OF *real* WITH DOM OF 1.0
age: FUZZY DOMAIN {*very old, old, young, very young*}: TYPE OF *integer* WITH DOM OF 1.0
star-name: TYPE OF *string* WITH DOM OF 1.0: REQUIRED UNIQUE
phone-numbers: TYPE OF *string* WITH DOM OF 1.0: MULTI-VALUED

According to these declarations, *location* and *age* attributes may have either exact values (e.g. *location* = 12LY; *age* = 55) or fuzzy values (e.g. *location* = very distant; *age* = old). The "LY" symbol is the abbreviation of light year. The *star-name* attribute may have only exact values (e.g. *star-name*= "Vega"). In addition, *star-name* attribute may be used as an identifier since it is required and unique. The *phone-numbers* attribute is an exact and multi-valued one.

The next component of fuzzy class definition is specific for fuzzy composite and grouping classes. It is an implementation of the *contents* attribute. For enumerated composition, we indicate the list of classes and for each one we specify the entities that are member of the fuzzy composite class. For attribute-defined composition, we fix the list of the selection attributes and the list of the classes from which selection is accomplished. For fuzzy aggregated classes we indicate the list of the classes that are part of the aggregation and for each one we specify the entities that are members of the fuzzy aggregate class. And finally for fuzzy grouping classes, we indicate the name of the class from which grouping is realized along with the list of members.

The last part of fuzzy class definition indicates the eventual interaction relationship(s) of the fuzzy class. As mentioned earlier, interaction relationships may be binary or *n*-ary. In both cases a name should be provided. Binary interaction relationships also require the name of the other participating fuzzy class and the name of the inverse attribute. For *n*-ary interaction relationships, we need to mention the list of the classes that participate in this interaction. In both cases and when it is necessary, the name of the fuzzy interaction class can be specified with the CLASS IS clause.

Since subclasses may have their own subclasses, they have the same components as for fuzzy classes. In particular, they may have SUPERCLASS components that indicate the list of their own subclasses. In turn, subclasses have a specific component, called SPECIALIZATION, that is designed to indicate their fuzzy superclasses. The generic definition of a fuzzy subclass in FSM is as follows (only the SPECIALIZATION component is provided; the definitions of the other components are similar to the ones of the fuzzy class and they are not reproduced):

SUBCLASS ⟨*class-name*⟩ WITH DOM OF ⟨*dom*⟩
{
SPECIALIZATION :
OF ⟨*class-name*⟩ WITH DOM OF ⟨*dom*⟩:
[BY ENUMERATION ⟨*members-list*⟩]
[ON ATTRIBUTES ⟨*attr-list*⟩]
[BY INTERSECTION WITH ⟨*class-list*⟩]
[BY DIFFERENCE WITH ⟨*class-name*⟩]
OF ⟨*class-name*⟩ WITH DOM OF ⟨*dom*⟩:
...
}

For each superclass of the subclass, we indicate the name of the superclass and the d.o.m. of the subclass in this superclass. Enumerated fuzzy subclasses require the enumeration of the fuzzy classes that participate in the generalization relationship along with the list of members. For attribute-defined subclasses, we should indicate the list of the attributes on which the ISA relationship is defined. For set-intersection-defined subclasses, we simply indicate the list of the other superclasses that participate in the intersection. Finally for set-difference-defined subclasses, we mention the name of the other fuzzy class that participates in the difference operation.

Several illustrative examples (based on Fig. 3) showing FSM schema definition are provided in Appendix A.

5. An extended query language

In this section, we introduce an ongoing conceptual query language adapted for FSM-based databases. First, we introduce the notions of perspective class and qualification, which are of great importance in FSM query formulation.

5.1. Perspective class and qualification

The notion of *perspective class* is introduced in [11]. It is simply defined as the class in which the user is primarily interested when formulating his/her query. It simplifies query formation and allows users with different interests to approach the database from points of view appropriate to their needs [11]. The perspective class can be associated with an appropriate syntactic process, called *qualification*, allowing immediate attributes of other classes to be treated as if they were attributes of the perspective class. This process may be extended through the entity-valued attributes concept to the attributes related by more than one level of qualification. These attributes are called *extended attributes*. For example, in Fig. 3, *field-of-research* is an immediate attribute of SCIENTIST and *name-of-person* is an inherited attribute of SCIENTIST from PERSON. If we suppose that classes SCIENTIST and TECHNICIAN in Fig. 3 are related and two entity-valued attributes *supervises* (from the point of view of SCIENTIST) and *supervisor* (from the point of view of TECHNICIAN) are defined for them, then with TECHNICIAN as perspective class, the *name-of-person* of *supervisor* refers to the name of a technician's supervisor(s) (i.e. a scientist entity). This last qualification is an extended attribute of TECHNICIAN in this example.

Apart from that, the notion of perspective class can be combined with generalization hierarchies to simplify query formation. For example, consider again the hypothetical binary relationship between SCIENTIST and TECHNICIAN, henceforth the list of technicians and the name of their supervisors can simply be obtained as follows (the syntax of a retrieve query in FSM is provided below):

FROM *technician* RETRIEVE *name-of-person*, *name-of-person* OF *supervisor*

In this example, TECHNICIAN is the perspective class. This query lists the name of all technicians and for each one it provides the name of his/her supervisor. But if a technician has no supervisor, whose name should be returned. In this case, the supervisor's name will take a "null" value. In this example, the qualification avoids the necessity to put the SCIENTIST class in the FROM clause since the entities of this class are "reached" through the entity-valued relationship.

The general syntax of qualification of an attribute is as follows [11]:

⟨*attr-name*⟩ {OF ⟨*entity-valued-attribute-name*⟩ [AS ⟨*class-name*⟩]}
OF ⟨*perspective-class-name*⟩ [AS ⟨*class-name*⟩]

The *attr-name* is either a data-valued or an entity-valued attribute. The "AS" clause specifies subclass/superclass role conversion (from a superclass to a subclass) in the same generalization hierarchy and may be best thought of as "looking down" a generalization hierarchy [11]. The following are some examples of qualification from Fig. 3:

name-of-person OF *person* AS *officer*
name-of-person OF *discoverer* OF *supernova*
laboratory-address OF *works-at* OF *person*
laboratory-name OF *works-at* OF *person* AS *scientist*

In the second qualification, the perspective class is SUPERNOVA. All the other qualifications use PERSON as the perspective class. The first one returns the names of persons who are officers. The second one lists for each supernova the name(s) of its discoverer(s). The third one returns for each person in the database the address of the laboratories s/he works at. The fourth qualification is like the previous one but it returns the working place name of persons who are scientists only, i.e., persons who are technicians or officers are not considered. The second example of qualification is used in the third and sixth queries below. The fourth example is used in the second query.

A last example of qualification is as follows. Suppose that a reflexive relationship called *spouse* is defined on the class PERSON. Then, the qualification:

field-of-research OF *spouse* AS *scientist* OF*scientist*

returns the field of research of scientist's spouse only if the spouse is also a scientist. Clearly, the perspective class here is SCIENTIST. This example is used in the seventh query below.

Finally, it is important to mention that it may be necessary to use more than one perspective class (as in queries 5 and 6 below) and that it is not necessary to qualify each attribute.

5.2. Syntax of retrieve queries

The generic syntax of a retrieve query in FSM is inspired from [11] and it makes use of the concept of perspective class:

[FROM {(⟨*perspective-class-name*⟩ [WITH DOM ⟨op_1⟩ ⟨*class-level*⟩]|
  ⟨α-MEMBERS OF *perspective-class-name*⟩)}]
RETRIEVE ⟨*target-list*⟩
[ORDER BY ⟨*order-list*⟩]
[WHERE ⟨*selection-expression*⟩ [WITH DOM ⟨op_2⟩⟨*attr-level*⟩]]

The argument of the FROM statement is a list of perspective class names (*perspective-class-name*) with their respective levels of selection (*class-level*) or a specification of the α-MEMBERS to be considered. Only members that have a global d.o.m. verifying the arithmetic comparison ensured by the operator op_1 (when WITH DOM is used) or have a d.o.m. greater or equal to α (when α-MEMBERS is used) are considered in the selection process. We notice that the WITH DOM part in the FROM clause is facultative and when omitted, all the entities of *perspective-class-name* that verify the condition(s) specified in the WHERE clause are returned. Thus, there is no necessity to introduce the WITH DOM > 0 condition when no restriction is imposed on the global d.o.m. of the entities as in queries 2, 5 and 6 hereafter. The *target-list* in the RETRIEVE statement is a list of expressions made up of constants, immediate, inherited and extended attributes of the perspective class, and aggregate and other functions applied on such attributes. The ORDER BY statement is used to choose the way the list of entities is ordered. The *selection-expression* in the WHERE statement is a set of symbolic, numerical or logical conditions that should be verified by the attributes of all selected entities. When it is necessary, attribute-based conditions may be combined with appropriate selection levels (*attr-level*) and only entities whose attributes values have a partial d.o.m. verifying the arithmetic comparison ensured by the operator op_2 are selected.

The following are some illustrative examples of data retrieval operations.

Query 1. Retrieve the name and the type of supernova that have global d.o.m. equal to or greater than 0.7 and have luminosity greater than $15L_s$ with partial d.o.m. equal to or greater than 0.9. (L_s is the luminosity of the Sun.)

FROM *supernova* WITH DOM $\geqslant 0.7$
RETRIEVE *snova-name*, *type-of-snova*
WHERE *luminosity* $> 15L_s$ WITH DOM $\geqslant 0.9$

Query 2. Retrieve the name of laboratory of all scientists in the database.

FROM *person*
RETRIEVE *laboratory-name* OF *works-at* OF *person* AS *scientist*

This query uses the fourth qualification cited above. Here, the perspective class is PERSON and it is not necessary to add a WHERE clause (to ensure that only persons who are scientists are listed) since the qualification permits to eliminate persons who are not scientists from the result.

Query 3. Retrieve the name of all true supernovae and the names of their discoverers.

FROM 1-MEMBERS OF *supernova*
RETRIEVE *snova-name*, *name-of-person* OF *discoverer* OF *supernova*

This query uses the second example of qualification cited earlier. Here, the qualification permits to avoid the necessity of adding the class SCIENTIST in the FROM clause. In addition, it avoids the necessity of a WHERE clause. Without qualification, a WHERE clause should be added and the response to query 3 will be as follows (note that the same supernova may be discovered by several scientists and that the same scientist may discover several supernovae):

FROM 1-MEMBERS OF *supernova*, *scientist*
RETRIEVE *snova-name*, *name-of-person*
WHERE *supernova.discoverer* in (
  FORM *scientist*
  RETRIEVE *name-of-person*
  WHERE *scientist.discovers=supernova.snova-name*)

Query 4. Retrieve location, luminosity and weight of each supernova having a global d.o.m. greater than 0.75 and is of type II-P and of weight greater than $10W_s$ with partial d.o.m. greater than 0.5.

FROM 0.75-MEMBERS OF *supernova*
RETRIEVE *location*, *luminosity*, *weight*
WHERE *type-of-snova*=II-P and *weight* $> 10W_s$ WITH DOM > 0.5

Query 5. Retrieve date of discovery and name of all supernovae of type Ia that are located in the milky-way galaxy with a partial d.o.m. greater than 0.5 and having high luminosity with d.o.m. less than 0.7.

FROM *discovery*, *supernova*, *galaxy*
RETRIEVE *snova-name*, *date-of-discovery*
WHERE *type-of-snova* = Ia and (*galaxy-name* = milky-way and
  galaxy.location = *supernova.location* WITH DOM > 0.5) and
  luminosity = high WITH DOM < 0.7

In this query example as in the next one, several perspective classes are used. In addition, the WITH DOM part is omitted from the FROM clause and so the conditions of the WHERE clause will be verified for all the entities.

Query 6. Retrieve the name, the date of discovery and the discoverer of all supernovae which are not located in the milky-way galaxy with d.o.m. not less than 0.5.

FROM *supernova*, *discovery*
RETRIEVE *snova-name*, *date-of-discovery*, *name-of-person* OF *discoverer* OF *supernova*
WHERE *supernova.location* not in (
  FROM *galaxy*
  RETRIEVE *location*
  WHERE *galaxy-name* = milky-way) WITH DOM $\geqslant 0.5$

This example illustrates an imbricated query in FSM. It shows again (in the outer block of the query) how the second example of qualification may be used.

Query 7. Retrieve all the scientists who are married and who have the same field of research as their spouses (we suppose that the *spouse* relationship defined on the PERSON class exists).

FROM *scientist*
RETRIEVE *name-of-person*
WHERE *field-of-research* OF *spouse* AS *scientist* OF *scientist* = *field-of-research*

This last query uses the fifth example of qualification cited earlier.

6. Related work

In this section, we compare FSM with some recent proposals for extending semantic data models.

Based on fuzzy set and possibility distribution, the author in [14] introduces fuzziness in the different constructs of the semantic IFO data model, including printable type, abstract type, free type, grouping, aggregation, fragment and ISA relationship. The obtained system, denoted IF_2O, is mapped to a fuzzy relational database model. It is fruitful to remark that several fuzzy constructs of IF_2O (e.g. abstract types with the fuzziness at the schema level) can not be mapped into the fuzzy relational database model. In addition, the author does not provide the ways in which the different d.o.m. are computed; thus, an effective comparison with FSM is not possible.

The authors in [16] use fuzzy set theory and possibility distribution to extend the EER model into a fuzzy EER (denoted FEER) one to cope with imperfect as well as complex objects. FEER supports fuzziness at model/type (i.e. class), type values (i.e. instances) and attribute levels. In addition, the authors distinguish two interpretations of fuzzy sets for modeling incomplete information within multi-valued attributes: distinctive fuzzy sets and conjunctive fuzzy sets. Furthermore, a formal design methodology for fuzzy object-oriented databases (FOODB) from a FEER model is detailed. But the ways in which the different d.o.m. are computed are not provided and several constructs of semantic modeling (e.g. grouping and composition) are not extended to cope with imperfect information.

Another proposal for extending the well-known ER database model to support fuzziness is reported in [8]. The possibility-based Fuzzy ER data model supports fuzziness and uncertainty at attribute, entity, relationships and instance/entity relationships. However, the Fuzzy ER described in [8] does not support fuzziness at subclass/superclass level. The paper does not present the ways in which the different d.o.m. are computed. In turn, it includes a fuzzy entity-relationship methodology (FERM) for the design and development of fuzzy relational databases.

In [22] and based on similarity relations, the IFO model is extended to the ExIFO (Extended IFO) to represent uncertainty as well as precise information. ExIFO supports uncertainty at the attribute, entity, and instance/class levels. The authors also provide an algorithm for mapping the schema of the ExIFO model to an extended NF^2 database model. The paper also includes some extended algebra operators along with an extended SQL-like query language. However, the ways in which the different d.o.m. are computed are not mentioned in the present paper. Moreover, it does not discuss the support of imperfect information within several constructs of semantic modeling including grouping, aggregation and subclass/superclass relationships.

On the basis of fuzzy set and possibility distribution, extension of the major constructs of the well-known ER/EER models to support uncertainty and imprecision of real-world at model/type, type/value and attribute values levels is introduced in [9]. The paper also discusses the attribute inheritance concept within fuzzy context. It deals especially with derived attribute inheritance, multiple inheritance associated with superclass/subclass relationships and selective inheritance of attributes associated with categories. The support of fuzziness at relationship constraints (inheritance constraint, participation constraint and cardinality constraint) is also investigated.

Finally, an extension of the graph-based IFO database model is provided in [21]. The paper first discusses the problem of vagueness, imprecision and uncertainty representation within fuzzy databases. Basing on IFO, the paper then proposes solutions for handling ill-defined values including values with semantic representation, values with semantic representation and disjunctive meaning and values with semantic representation and conjunctive meaning. The uncertainty is supported at attribute, object and class levels. The paper includes a graphical representation of some constructs of IFO. However, the paper lacks a discussion of several important concepts in semantic modeling such as entity/class and subclass/superclass relationships.

In comparison with the above-cited proposals, FSM has several merits:

- FSM is semantically richer since it is based on the USM model [18] that coherently synthesizes and extends constructs found in several other conventional semantic models.
- FSM introduces fuzziness within all the constructs of semantic modeling: attribute, entity, composition, aggregation, and grouping semantics as well as entity/class, subclass/superclass and more generally class/class relationship levels.
- FSM supports almost all kinds of imperfect information: vague, imprecise, uncertain and incomplete (i.e null, undefined and unknown) (see [1]). This ensures the high flexibility of FSM. Furthermore, all these types are uniformly represented through possibility distribution, which facilitates data manipulation and computing while giving the maximum flexibility to the users.

7. Conclusion

In database research, there are several proposals to develop database models that support the management of fuzziness, uncertainty and imprecision of real-world. In this paper, we have reviewed and refined the FSM, a fuzzy semantic model recently proposed by the authors. In addition, we have provided a proposal for specifying FSM schema and introduced a query language adapted to FSM-based databases. Furthermore, we have discussed/compared FSM with several recent fuzzy semantic data models.

In FSM, we need to compute and handle both d.o.m. of entity/class and subclass/superclass relationships. This will complicate DBMS design process. Another drawback of FSM (and nearly all other proposals) is related to compensatory nature of the weighted sum technique used to define and calculate global membership functions. Indeed, the low values of one or many partial membership functions may be compensated with high values of one or many other partial membership functions. Thus, other, non-compensatory, aggregation operators are required.

Currently, we are concerned with several topics related to the enhancement of the developed prototype. Further attentions are devoted to enrich FSM with other tools that may be used to represent integrity constraints as well to extend conventional data manipulation operators (e.g. product, union, etc.) with fuzzy concepts.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which improved the technical and the presentation of the paper. The second author would like to thank Prof. M. Manouvrier for her valuable comments on an earlier version of this paper.

Appendix A

This Appendix contains the definition of several classes taken from Fig. 3.

CLASS *star* WITH DOM OF *dom*
{
SUPERCLASS:
OF *supernova* WITH DOM OF *dom*
OF *nova* WITH DOM *dom*
EXTENT:
sp_1 WITH WEIGHT OF 0.8 DECISION RULE IS *luminosity* $\geqslant 0.005L_s$

sp_2 WITH WEIGHT OF 0.3 DECISION RULE IS $weight \geqslant 0.05W_s$
ATTRIBUTES:
star-name: TYPE OF *string* WITH DOM OF 1.0: REQUIRED UNIQUE
type-of-star: TYPE OF *symbolic*(*nova, supernova*) WITH DOM OF 1.0: REQUIRED
age: FUZZY DOMAIN {*very young, young, old, very old*}: TYPE OF *integer* WITH DOM OF 1.0: REQUIRED
location: FUZZY DOMAIN {*in, near, very near, distant, very distant*}: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
luminosity: FUZZY DOMAIN {*very low, low, medium, high, very high*}: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
weight: FUZZY DOMAIN $[0.01Ws - 100W_s]$: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
}
SUBCLASS *supernova* WITH DOM OF *dom*
{
SPECIALIZATION :
OF *star* WITH DOM OF *dom*:
ON ATTRIBUTES *type-of-star*
EXTENT:
snp_1 WITH WEIGHT OF 0.6 DECISION RULE IS *luminosity* ⩾ high
snp_2 WITH WEIGHT OF 0.5 DECISION RULE IS $weight \geqslant 1W_s$
ATTRIBUTES:
snova-name: TYPE OF *string* WITH DOM OF 1.0: REQUIRED UNIQUE
type-of-snova: TYPE OF *symbolic* (*Ia, Ib, Ic, Ib/c, Ic/b, II-P, II-L*) WITH DOM OF 1.0: REQUIRED
luminosity: FUZZY DOMAIN {*high, very high*}: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
weight: FUZZY DOMAIN $[1W_s - 100W_s]$: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
INTERACTION:
discoverer WITH *scientist* INVERSE IS *discovers* CLASS IS *discovery*
}
CLASS *person* WITH DOM OF 1.0
{
SUPERCLASS:
OF *scientist* WITH DOM OF 1.0
OF *technician* WITH DOM OF 1.0
OF *officer* WITH DOM OF 1.0
EXTENT:
pp_1 WITH WEIGHT OF 1.0 DECISION RULE IS set of persons
ATTRIBUTES:
name-of-person: TYPE OF *string* WITH DOM OF 1.0: REQUIRED
age: FUZZY DOMAIN {*very young, young, old, very old*}: TYPE OF integer WITH DOM OF 1.0: REQUIRED
address: TYPE OF *string* WITH DOM OF 1.0: REQUIRED
phone-numbers: TYPE OF *string* WITH DOM OF 1.0: MULTI-VALUED
INTERACTION:
works-at WITH *laboratory* INVERSE IS *working-place-of*
}
CLASS *discovery* WITH DOM OF *dom*
{
INTERACTION CLASS OF *supernova*, *scientist*
ATTRIBUTES:
date-of-discovery: TYPE OF *datetime* WITH DOM OF 1.0
place-of-discovery: TYPE OF *string* WITH DOM OF 1.0
}

SUBCLASS *scientist* WITH DOM OF 1.0
{
SPECIALIZATION:
OF *person* WITH DOM OF 1.0:
BY ENUMERATION *name-of-person-1,...,name-of-person-n*
EXTENT:
scp_1 WITH WEIGHT OF 1.0 DECISION RULE IS set of scientists
ATTRIBUTES:
field-of-research: TYPE OF *string* WITH DOM OF 1.0: REQUIRED
INTERACTION:
discovers WITH *supernova* INVERSE IS *discoverer* CLASS IS *discovery*
}

References

[1] A. Bahri, R. Bouaziz, S. Chakhar, Y. Naïja, Implementing imperfect information in fuzzy databases, in: Proceedings of The International Symposium on Computational Intelligence and Intelligent Informatics, Tunis-Gammarth, Tunisia, October 14–16, 2005.
[2] G. Bordogna, G. Pasi (Eds.), Recent Issues on Fuzzy Databases, Physica-Verlag, Heidelberg, GD, 2000.
[3] G. Bordogna, G. Pasi, D. Lucarella, A fuzzy object-oriented data model for managing vague and uncertain information, International Journal of Intelligent Systems 14 (1999) 623–651.
[4] P. Bosc, D. Kraft, F. Petry, Fuzzy sets in database and information systems: status and opportunities, Fuzzy Sets and Systems 156 (2005) 418–426.
[5] P. Bosc, O. Pivert, SQLf query functionality on top of a regular relational database management system, Studies in Fuzziness and Soft Computing 39 (2000) 171–191.
[6] S. Chakhar, A. Telmoudi, Extending database capabilities: fuzzy semantic model, in: Proceedings of The International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT'04), Sousse, Tunisia, March 15–20, 2004.
[7] S. Chakhar, A. Telmoudi, Conceptual design and implementation of the fuzzy semantic model, in: Proceedings of The 11th International Conference on Information Processing and Management of Uncertainty (IPMU 2006), vol. III, Paris, France, July 2–7, 2006, pp. 2438–2445.
[8] N. Chaudhry, J. Moyne, E.A. Rundensteiner, An extended database design methodology for uncertain data management, Information Sciences 121 (1999) 83–112.
[9] G.Q. Chen, E.E. Kerre, Extending ER/EER concepts towards fuzzy conceptual data modeling, in: Proceedings of The IEEE International Conference on Fuzzy Systems, vol. 2, Anchorage, Alaska, USA, May 4–9, 1998, pp. 1320–1325.
[10] S.K. De, R. Biswas, A.R. Roy, On extended fuzzy relational database model with proximity relations, Fuzzy Sets and Systems 117 (2001) 195–201.
[11] B.L. Fritchman, R.L. Guck, D. Jagannathan, J.P. Thompson, D.M. Tolbret, SIM: design and implementation of a semantic database system, in: Proceedings of The ACM SIGMOD Conference, 1989, pp. 46–55.
[12] R. George, R. Srikanth, F.E. Petry, B.P. Buckles, Uncertainty management issues in object-oriented data model, IEEE Transaction on Fuzzy Systems 4 (1996) 179–192.
[13] N.V. Gyseghem, R.D. Caluwe, Imprecision and uncertainty in UFO database model, Journal of American Society of Information Sciences 49 (3) (1998) 236–252.
[14] Z.M. Ma, A conceptual design methodology for fuzzy relational databases, Journal of Database Management 16 (2) (2005) 66–83.
[15] Z.M. Ma, W.J. Zhang, W.Y. Ma, Extending object-oriented databases for fuzzy information modeling, Information Systems 29 (2004) 421–435.
[16] Z.M. Ma, W.J. Zhang, W.Y. Ma, G.Q. Chen, Conceptual design of fuzzy object-oriented databases using extended entity-relationship model, International Journal of Intelligent Systems 16 (2001) 697–711.
[17] J.M. Medina, O. Pons, M.A. Vila, GEFRED: a generalized model of fuzzy relational databases, Information Sciences 76 (1994) 87–109.
[18] S. Ram, Intelligent database design using unifying semantic model, Information and Management 29 (4) (1995) 191–206.
[19] D. Rocacher, P. Bosc, The set of fuzzy rational numbers and flexible querying, Fuzzy Sets and Systems 155 (2005) 317–339.
[20] S. Shenoi, A. Melton, An extended version of the fuzzy relational database model, Information Sciences 52 (1) (1990) 35–52.
[21] M.A. Vila, J.C. Cubero, J.M. Medina, O. Pons, A conceptual approach for dealing with imprecision and uncertainty in object-based data models, International Journal of Intelligent Systems 11 (1996) 791–806.
[22] A. Yazici, B.P. Buckles, F.E. Petry, Handling complex and uncertain information in the ExIFO and NF^2 data models, IEEE Transactions on Fuzzy Systems 7 (6) (1999) 659–676.
[23] A. Yazici, D. Cibicel, An access structure for similarity-based fuzzy databases, Information Sciences 115 (1999) 137–163.
[24] A. Yazici, R. George, Fuzzy Database Modeling, Springer-Verlag, Heidelberg, 1999.
[25] A. Yazici, R. George, D. Aksoy, Design and implementation issues in fuzzy object-oriented data model, Information Sciences 108 (1/4) (1998).