PREMIER CONGRES INTERNATIONAL DE
SIGNAUX, CIRCUITS & SYSTEMES
Monastir du 18 au 21 Mars 2004

Some Notes on an Extended Query Language for FSM

Abdelkader Telmoudi[1], and Salem Chakhar[2]

[1]*Ecole Nationale d'Ingénieurs de Tunis, B.P. 37 Le Belvédère 1002 - Tunis, Tunisie.*
abdelkader.telmoudi@enit.rnu.tn

[2]*LAMSADE, Université Paris Dauphine,*
Place du Maréchal de Lattre de Tassigny, F-75775 Paris Cedex 16, France.
chakhar@lamsade.dauphine.fr

Abstract—FSM is a database model that has been recently proposed by the authors. FSM uses basic concepts of classification, generalization, aggregation and association that are commonly used in semantic modelling and supports the fuzziness of real-world at attribute, entity, class and relations intra and inter-classes levels. Hence, it provides tools to formalize and conceptualize real-world within a manner adapted to human perception of and reasoning about this real-word. In this paper we briefly review basic concepts of FSM and provide some notes on an extended query language adapted to it.

Keywords: — Fuzzy class, Fuzzy database, Fuzzy query language, Fuzzy semantic model.

1 INTRODUCTION

There have been several tentatives to develop data models that support the fuzziness of real-world [1][4][9]. Most of these works consider that each entity in real-world belongs to one and only one class, which is very restrictive in many applications (e.g. cosmology, archaeology, spatial modelling). In database literature, we may enumerate some recent extensions of object-oriented data models to support the uncertainty of real-world at the class definition level [5][10][7] and only few extensions of semantic data models have been proposed [3][6]. In the same direction of research, the authors have proposed a new database model, namely Fuzzy Semantic Model (FSM) [2], that supports fuzzy class definition and authorises an entity to be a member of one or more classes with different degrees of membership. In this paper we briefly review basic concepts of FSM and then provide some notes on an extended query language adapted to it.

2 FUZZY SEMANTIC MODEL

2.1 Constructs of FSM

Constructs of FSM (see [2] for a full description of FSM) are extensions of the Unifying Semantic Model [8] that are enriched with new concepts enabling the database system to support fuzziness of real-world.

The *space of entities E* is the set of all entities of the *domain of interest*. A *fuzzy entity* ***e*** in *E* is a natural or artificial entity that we cannot assign to an exact class. In other words, a fuzzy entity verifies only (partially) some extent properties of one or some classes. Classic entities are particular case of fuzzy entities since they are assigned to exactly one class.

A *fuzzy class K* in *E* is a collection of *fuzzy entities*: $K=\{(e,\mu_K(e)): e\in E; \mu_K(e) > 0\}$. μ_K is a *characteristic* or *membership function* and $\mu_K(e)$ represents the *degree of membership* (or 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 entity ***e*** can be considered as an element of fuzzy class *K*.

FSM contains several types of basic fuzzy classes that are summed up in Table 1. The elements of a fuzzy class are called *members*. Each fuzzy class *K* may have any number of three types of members: (*a*) *true-members* (i.e. entities ***e*** with $\mu_K(e)=1$), (*b*) *pseudo-members* (i.e. entities ***e*** with $0.17 \leq \mu_K(e) < 1$), and (*c*) *weak-members* (i.e. entities ***e*** with $0 < \mu_K(e) < 0.17$).

FSM supports four different relationships: property, decision-rule, membering and interaction. Property relationships relate fuzzy classes to domain classes. Each property relationship creates an *attribute* and each attribute has a unique datatype and may be *single-valued*, *unknown*, *undefined*, *null* or *multi-valued*. Decision rule relationships are implementation of the extents of fuzzy classes, i.e., the set of properties-based rules used to assign fuzzy entities to fuzzy classes (see 2.2). Membering relationships relate fuzzy entities to fuzzy classes through the definition of d.o.m. Interaction relationships relate members of one fuzzy class to other members of one or several fuzzy classes.

Name	***Description***
Complete fuzzy class	A fuzzy class that all its members have a d.o.m equal to 1
Non-complete fuzzy class	A fuzzy class that has at least one member with a d.o.m strictly less than 1
Strong fuzzy class	A fuzzy class whose members can exist on their own
Weak fuzzy class	A fuzzy class that the existence of its members depend on the existence of other classes
Compact fuzzy class	A complete and strong fuzzy class
Non-compact fuzzy class	A complete and weak fuzzy class
Fuzzy entity-class	A new entity that cannot be assigned to any pre-existing fuzzy class
Domain class	A space to which an attribute's values are mapped

Table 1: FSM basic fuzzy classes

In FSM there are several complex fuzzy classes (see Table 2), that permit to implement the semantics of real-world among objects in terms of generalization, specialization, aggregation, grouping and composition relationships, which are commonly used in purely semantic modelling.

2.2 Entity/Class Membership Function in FSM

As it is underlined above, a fuzzy class is a collection of many fuzzy entities having some similar properties. Fuzziness is thus induced whenever an entity verifies only (partially) some of these properties. We denote by $P_K=\{p^i_K\text{: } K \subset E;\ i >=1\}$ the set of these properties for a given fuzzy class K. P_K is the *extent* of class K. These properties may be derived from the attributes of the class and/or from common semantics. The extent to which each of these properties determines the class K is not the same. Indeed, there are some properties that are more discriminative than others. To ensure this, we associate to each property p^i_K a normalized weight w^i_K reflecting its importance in deciding whether or not an entity e is a member of a given class K. To keep the coherence of our model, we impose that $\Sigma_i w^i_K = 1$.

On the other hand, an entity may verify fully or partially extent properties of a given fuzzy class. Let D^i_K be the basic domain of extent property p^i_K values and P^i_K is a subset of D^i_K, which represents the set of possible values of property p^i_K. The partial membership function of an extent property value is $\rho_{p^i_K}$, which maps elements of D^i_K into [0,1]. For any attribute value $v \in D^i_K$, $\rho_{p^i_K}(v) = 0$ means that fuzzy entity e violets property p^i_K and $\rho_{p^i_K}(v) = 1$ means that this entity verifies fully the property value. The number v is the value of the attribute of entity e on which the property p^i_K is defined. More generally, the value of $\rho_{p^i_K}(v)$ represents the extent to which entity e verifies property p^i_K of fuzzy class K. Accordingly, for any fuzzy entity e, the global d.o.m $\mu_K(e)$ for a fuzzy class K is equal to $\Sigma_i\, w^i_K \cdot \rho_{p^i_K}(v)$.

Name	***Description***
Interaction fuzzy class	A fuzzy class that describe the interaction of two or more fuzzy classes
Fuzzy superclass	A generalization of one or many, simple or complex, fuzzy classes
Fuzzy subclass	A specialization of one or many, simple or complex, fuzzy classes
Composite fuzzy class	A fuzzy class that has other fuzzy classes as members
Aggregate fuzzy class	A fuzzy class that its members are heterogeneous and exhaustive collection from several fuzzy classes
Grouping fuzzy class	A fuzzy class that its members are homogenous collection of members from the same fuzzy class

Table 2: FSM complex fuzzy classes

3 AN EXTENDED QUERY LANGUAGE

In this section we present some notes on an ongoing conceptual query language for accessing FSM-based databases and illustrate some examples of data retrieve operations. All examples of this section rely on the database schema illustrated in Figure 1. For the sake of clarity, only necessary classes and relationships are depicted in this figure.

In the example database, GALAXY is an aggregate fuzzy class whose members are unique collections of members from COMETS, STARS, and PLANETS grouping fuzzy classes. These last ones are homogenous collections of members from strong fuzzy classes COMET, STAR, and PLANET, respectively. NOVAE and SUPERNOVAE are two attribute-defined fuzzy subclasses of STAR basing on *Type_of_Star* attribute. SCIENTIST is a collection of astronomists and DISCOVEY is an interaction fuzzy class between SUPERNOVAE and SCIENTIST.

The general definitions of a fuzzy class and a fuzzy subclass in FSM are as follows (note that attributes' definitions are partially inspired from [7]).

CLASS <*class_name*> WITH DOM OF <*gdom*>
{
SUPERCLASS:
OF <*sclass_name_1*> WITH DOM <*scdom_1*>
...
OF <*sclass_name_k*> WITH DOM <*scdom_k*>

EXTENT:
{<*Ext_attr_1*> | <*Ext_sphrase_1*>} WITH WEIGHT <w_1>
...
{<*Ext_attr_n*> | <*Ext_sphrase_n*>} WITH WEIGHT <w_n>

ATTRIBUTES:
Attr_1_name: [FUZZY] DOMAIN <*domaine_1*>: TYPE OF <*type_1*> WITH DOM OF <*dom_1*>: [REQUIRED][UNIQUE]
...
Attr_r_name: [FUZZY] DOMAIN <*domaine_r*>: TYPE OF <*type_r*> WITH DOM OF <*dom_r*>: [REQUIRED][UNIQUE]

CONTENTS:
[ENUMERATED COMPOSITION FROM <*class_list_1*>]
[SELECTED COMPOSITION ON ATTRIBUTES <*attr_list_1*> FROM <*class_list_2*>]
[AGGREGATION OF <*class_list_3*>]
[GROUPING FROM <*class_name_2*>]

INTERACTION:
<*inter_name_1*> WITH <*c_name_1*> INVERSE IS <*i_inter_name_1*>
...
<*inter_name_z*> WITH <*c_name_z*> INVERSE IS <*i_inter_name_z*>
}

SUBCLASS <*sclass_name*> WITH DOM OF <*scdom*>
{
SPECIALIZATION:
OF <*class_name_1*> WITH DOM <*scdom_1*>:
[BY ENUMERATION]
[ON ATTRIBUTES <*attr_list_1*>]
[ON SEMANTICS <*sphrase_list_1*>]
[BY INTERSECTION WITH <*class_list_1*>]
[BY DIFFERENCE WITH <*d_class_name_1*>]
OF <*class_name_q*> WITH DOM <*scdom_q*>:
...
EXTENT:
{<*Ext_attr*>|<*Ext_sphrase*>} WITH WEIGHT <w_1>
...
{<*Ext_attr*>|<*Ext_sphrase*>} WITH WEIGHT <w_s>

ATTRIBUTES:
Attr_1_name: [FUZZY] DOMAIN <*domaine_1*>: TYPE OF <*type_1*> WITH DOM OF <*dom_1*>: [REQUIRED][UNIQUE]
...
Attr_t_name: [FUZZY] DOMAIN <*domaine_t*>: TYPE OF <*type_t*> WITH DOM OF <*dom_t*>: [REQUIRED][UNIQUE]

INTERACTION:
<*inter_name_1*> WITH <*c_name_1*> INVERSE IS <*i_inter_name_1*>
...
<*inter_name_y*> WITH <*c_name_y*> INVERSE IS <*i_inter_name_y*>
}

Subclasses may have their own subclasses and if this is the case, they must have SUPERCLASS components.

As examples, fuzzy classes GALAXY, STAR and SUPERNOVAE in the example database may be implemented as follows:

CLASS *galaxy* WITH DOM OF *gdom*
{
EXTENT:
Location WITH WEIGHT 1.0

ATTRIBUTES:
Name: TYPE OF *string* WITH DOM 1.0: REQUIRED UNIQUE
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

CONTENTS:
AGGREGATION OF *comets*, *stars*, *planets*
}

CLASS *star* WITH DOM *sdom*
{
SUPERCLASS:
OF *supernovae* WITH DOM *scdom_sn*
OF *novae* WITH DOM *scdom_n*

EXTENT:
Luminosity WITH WEIGHT 0.7
Weight WITH WEIGHT 0.3

ATTRIBUTES:
S_Name: TYPE OF *string* WITH DOM 1.0: REQUIRED UNIQUE
Type_of_Star: TYPE OF *symbolic* (*novae*, *supernovae*) WITH DOM 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 1.0: REQUIRED
Luminosity: FUZZY DOMAIN [$0.0001L_s$-$100000L_s$]: TYPE OF *real* WITH DOM OF 1.0
Weight: FUZZY DOMAIN [$0.1W_s$-$100W_s$]: TYPE OF *real* WITH DOM OF 1.0: REQUIRED
}

SUBCLASS *supernovae* WITH DOM *sndom*
{
EXTENT:
Luminosity WITH WEIGHT 0.3
Weight WITH WEIGHT 0.7

ATTRIBUTES:
SN_Name: TYPE OF *string* WITH DOM 1.0: REQUIRED UNIQUE
Type_of_SN: TYPE OF *string* WITH DOM 1.0: REQUIRED

SPECIALIZATION:
OF *star* WITH DOM *scdom* ON ATTRIBUTES *Type_of_Star*

INTERACTION:
discoverer WITH *scientist* INVERSE IS *discovers*
}

The symbols L_s and W_s above are luminosity and weight of the sun, respectively; they are often used as measurement units. In the following, we provide some notes on an ongoing conceptual query language through some examples of data retrieve operations.

The syntax of a retrieve query in FSM is as follows:

[**FROM** ({<*class_name*> **WITH DOM** <op_1> <*class_level*> | <*member_type*> **MEMBER OF** <*class_name*>})]
RETRIEVE {<*attribute_list*>|(<*attr_ name*> **OF** <*inter_name*>)}
[**ORDER BY** <*order_list*>]
[**WHERE** (<*selection_express*> [**WITH DOM** <op_2> <*attr_level*>])]

The argument of the FROM statement is a list of classes' names with their respective levels of selection (*class_level*) or a specific type of members (*member_type*), which can take the value of True, Pseudo or Weak. Only members that have a global d.o.m verifying the arithmetic comparison ensured by the operator op_1 or have a type of *member_type* are considered in the selection process. The *attribute_list* in the RETRIEVE statement is the list of attributes to be delivered to the user. This statement permits also to select attributes of any class that is in interaction with one or more classes in the FROM statement. The ORDER BY statement is used to choose the way the list of entities is ordered. The *selection_express* 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, attributes-based conditions may be combined with appropriate selection levels (*attr_level*) and only entities that their attributes' values have a partial d.o.m verifying the arithmetic comparison ensured by the operator op_2 are selected. Some examples of data retrieve operations are enumerated below.

Query1. Retrieve the name and type of supernovae that have d.o.m equal or greater to 0.7 and have luminosity greater to $5L_s$ with partial d.o.m equal or greater to 0.9.

FROM *supernovae* WITH DOM ≥ 0.7
RETRIEVE *SN_Name*, *Type_of_SN*
WHERE *Luminosity* > $5L_s$ WITH DOM ≥ *0.9*

Query2. Retrieve the name of all true supernovae, and the name and laboratory of their discoverers.

FROM *True* MEMBER OF *supernovae*
RETRIEVE *SN_Name*, *Name* OF *discoverer, Laboratory* OF *discoverer*

Query3. Retrieve location, luminosity and weight of all stars having d.o.m. greater to 0.75 and of type supernovae of type SN II and of weight greater to 10 W_s with partial d.o.m greater to 0.5 .

FROM *star* WITH DOM > *0.75*
RETRIEVE *Location, Luminosity, Weight*
WHERE *Type_of_Star*=*supernovae* and *type_of_SN*= *SN II* and *weigtht* > $10W_s$ WITH DOM > *0.5*

Query4. Retrieve dates of discover and names of all supernovae of type SN I that are located in milky way galaxy with a d.o.m. greater to 0.5 and having luminosity greater to $17L_s$ with d.o.m less to 0.7.

FROM *discovery, supernovae, galaxy*
RETRIEVE *SN_Name*, *Discovry_Date*,
WHERE *Type_of_SN* = *SN I* and (*galaxy.Name*=*milky-way* and *galaxy.Location*= *supernovae.Location* WITH DOM >*0.5*) and *Luminosity* > $17L_s$ WITH DOM < *0.7*

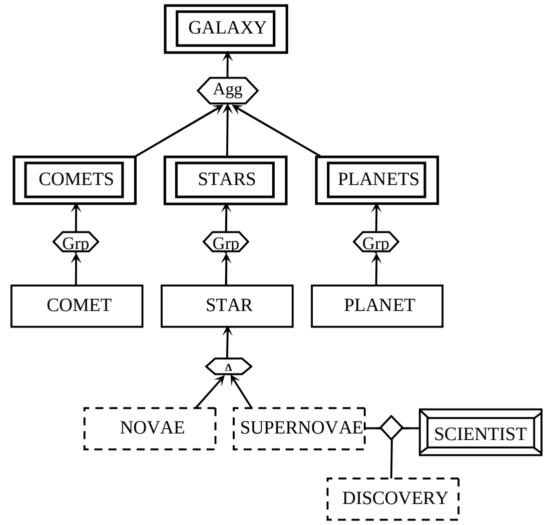

Figure 1: Graphical representation of the example database

4 CONCLUSION

In this paper, we have briefly reviewed basic concepts of FSM—a database model that has been recently proposed by the authors—and provide some notes on an extended query language adapted to it. In current time, we investigate the problem of the definition of partial membership functions through possibility and evidence theories. We are also concerned with several topics related to the implementation of a prototype of FSM.

REFERENCES

[1] W.P. Buckles, F.E. Petry, "A fuzzy model for relational databases," *Fuzzy Sets Systems*, vol. 33, pp. 213-226, 1992.

[2] S. Chakhar, A. Telmoudi, "Extending database capabilities: Fuzzy semantic model," *Proc. Int. Conf. Sci. of Electronic, Technologies of Inf. and Telecommunications 2004*, to appear.

[3] G.Q. Chen, E.E. Kerre, "Extending ER/EER concepts towards fuzzy conceptual data modeling," *Proc. of the 1998 IEEE Int. Conf. on Fuzzy Systems*, pp. 1320-1325, 1998.

[4] D. Dubois, H. Prade, "Using fuzzy sets in database systems: Why and how?," *Proc. Flexible Query Answering Systems*, pp. 89-103, 1996.

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

[6] W.Y. Liu, N. Song, "The fuzzy association degree in semantic data models," *Fuzzy Sets Systems*, vol. 117, pp. 203-208, 2001.

[7] Z.M. Ma, W.J. Zhang, W.Y. Ma, "Extending object-oriented databases for fuzzy information modeling," *Inf. Sys.*, in press.

[8] S. Ram, "Intelligent database design using unifying semantic model," *Inf. and Management*, vol. 29, pp. 191-206, 1995.

[9] A. Yazici, R. George, "*Fuzzy database modeling*," Heidelberg: Springer-Verlag, 1999.

[10] A. Yazici, R. George, D. Aksoy, "Design and implementation issues in fuzzy object-oriented data model," *Inf. Sci.*, vol. 108, pp. 241-260, 1998.