On Modeling the Behavior of Comparators for Complex Fuzzy Objects in a Fuzzy Object-Relational Database Management System

Juan M. Medina [1] **, Carlos D. Barranco** [2] **, Jesús R. Campaña** [1] **, Sergio Jaime-Castillo** [1]

[1] Department of Computer Science and Artificial Intelligence, University of Granada
C/ Periodista Daniel Saucedo Aranda s/n,
Granada 18071, Spain
{medina,jesuscg,sjaime}@decsai.ugr.es

[2] Division of Computer Science, School of Engineering, Pablo de Olavide University
Utrera Rd. Km. 1,
Seville 41013 , Spain.
cbarranco@upo.es

Received 10 July 2012
Accepted 18 July 2012

Abstract

This paper proposes a parameterized definition for fuzzy comparators on complex fuzzy datatypes like fuzzy collections with conjunctive semantics and fuzzy objects. This definition and its implementation on a Fuzzy Object-Relational Database Management System (FORDBMS) provides the designer with a powerful tool to adapt the behavior of these operators to the semantics of the considered application.

Keywords: Fuzzy Databases, Fuzzy Object-Relational Databases, Complex Fuzzy Objects Comparison

1. Introduction

Fuzzy database models and systems have evolved from being extensions of the relational model to be extensions of the object-oriented and object-relational database models. These two last approaches deal with complex fuzzy datatypes, where the semantics of the fuzzy operators involved in complex object retrieval is dependent on the considered application. In previous work[1] we use a FORDBMS to represent and store dominant color descriptions extracted from images stored in the database. To perform flexible retrieval of the images based on their dominant colors, it is necessary to use implementations of fuzzy operators that compute the inclusion degree of a set of dominant colors into another set. Additionally, if we are interested in the retrieval of images with a similar dominant color description, the system must provide an implementation for the resemblance operator for conjunctive fuzzy collections and for fuzzy objects. In another previous work[2] we use a FORDBMS to represent description of curves in spines suffering a deformation called scoliosis. To obtain appropriate results in the fuzzy search, it is necessary to use an implementation of the operators involved in complex data retrieval that is different from the one used in a dominant color based image retrieval application. These facts prove that, for complex objects, it is necessary to provide a parameterized approach to adapt the behavior of the object comparison operations to the se-

Published by Atlantis Press

mantics of the each application.

There are several proposals to model comparison on sets, like the work of Hallez et al. [3], which are based on a framework by Dubois and Prade [4] for comparison of fuzzy sets, where they integrate the element similarities in the comparison of the sets. Interpreting a class as a set of attributes, Marín et al.[5] propose a recursive way of comparing two complex fuzzy objects of a given class, avoiding cycles. Additionally, this work proposes some operators for computing the resemblance between fuzzy sets of fuzzy objects. An extension to this model, proposed by Berzal et al.[6], provides a wide variety of options to model the behavior of the fuzzy operators for comparison of complex objects. A later work by Marín et al.[7] proposes an approach for computing the fuzzy inclusion in a flexible way, using OWA operators as aggregation method.

This paper proposes a framework, starting from the proposal of Marn et al.[5], for the operators involved in comparison operations on fuzzy collections of elements with conjunctive semantics, and on complex fuzzy objects. The proposed definition for these operators integrates several parameters that allow us to adapt the behavior of the comparator to the semantics of the application. This extension is designed for integration into a FORDBMS. Therefore, the paper includes a proposal for the integration of these operators into a FORDBMS, describing the structures and methods that provide the designer with the necessary mechanisms to set the desired behavior of these operators in accordance to the semantics of a particular application.

The paper is organized as follows. Section 2 describes the general structure of our FORDBMS. Section 3 introduces the new definition proposed for the comparators on complex fuzzy datatypes. The extension of the catalog to parameterize these comparators is described in Section 4. An example illustrating a real world application of the proposal is shown in Section 5. Finally, main conclusions and future work are summarized in Section 6.

2. The Fuzzy Object-Relational Database Management System

In previous work[8,9], we introduced the basis for the implementation of our FORDBMS, that is based on the extension of a market leader RDBMS (Oracle®) by using its advanced object-relational[10,11] features. This strategy let us take full advantage of the host RDBMS features (high performance, scalability, availability, etc.) adding the capability of representing and handling fuzzy data provided by our extension.

Following the above strategy, the FORDBMS is built over the host DBMS taking advantage of its really powerful extension capabilities. The representation of fuzzy data is made through user-defined datatypes (UDT) and some convenience user-defined operators that allow users to create SQL sentences to query fuzzy data and apply flexible conditions. The necessary logic for handling fuzzy data is encapsulated as methods in the UDTs. All this within the framework of SQL:2003[12]. The following subsections depict the capabilities of such a FORDBMS.

2.1. Fuzzy Data Support

Our FORDBMS includes a wide variety of UDTs that allow to easily model any sort of fuzzy data. Particularly, the system is able to represent the following types of fuzzy data:

- Atomic fuzzy types (AFT), fuzzy data represented as possibility distributions over ordered (OAFT) or non ordered (NOAFT) domains.
- Fuzzy collections (FC), fuzzy sets of objects (from an homogeneous domain, crisp or fuzzy) with conjunctive (CFC) or disjunctive (DFC) semantics.
- Fuzzy objects (FO), whose attribute types can be crisp or fuzzy, and where each attribute is associated with a degree to weigh its importance in object comparison.

The above types of fuzzy data can be divided in two groups. FC and FO data is named *complex* fuzzy data as each data item is composed of other

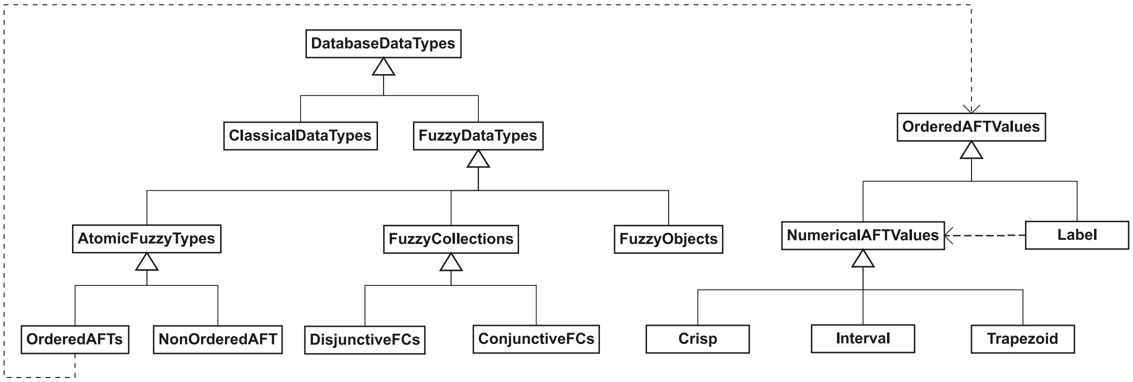


Fig. 1. Datatype hierarchy for the FORDBMS

data items that can be, in turn, fuzzy. Likewise, all AFT data is named *simple* fuzzy data, as the underlying domain on which a data item is defined must be crisp. Particularly, it is worth noticing that a *collection* is defined as a set of objects that, in turn, may represent other complex fuzzy structures, even including other nested collections. Therefore, following the common terminology in Object Oriented Programming and for the purposes of this paper, we use the term *collection* in favor of the term *set* to stress that the elements of this kind of structure can be complex fuzzy objects.

The UDTs available in the FORDBMS for representing the above types of fuzzy data are depicted in Fig.1, which shows an UML diagram illustrating these datatypes and their hierarchical relations. The diagram includes two abstract datatypes that, even though they are not part of the set of the UDTs for fuzzy data representation, are helpful to understand how the fuzzy UDTs are integrated in the host datatype system. The first one is *DataBaseDataTypes*, which represents a common ancestor for all datatypes in the host DBMS. The second is *ClassicalDataTypes*, which represents all the datatypes of the host DBMS that are not a fuzzy UDT.

Among the fuzzy UDTs, *AtomicFuzzyTypes* is the common ancestor for all UDTs representing an AFT. Likewise, *OrderedAFTs* is the common ancestor for all UDTs representing an OAFT. The different forms in which OAFT data can be represented in the FORDBMS are implemented by the UDTs whose common ancestor is the *OrderedAFTValues* datatype. One of the forms in which an OAFT can be represented is numerical: trapezoidal possibility distributions (by means of the UDT *Trapezoid*), intervals (*Interval* UDT) or crisp values (*Crisp* UDT). All the numerical representations have the same common ancestor, the *NumericalAFTValues* UDT. Alternatively, an OAFT can be represented as a linguistic label (by means of the UDT *Label*), which in turn is linked to some sort of OAFT numerical value.

Regarding the rest of the UDT hierarchy for representing fuzzy data, the *NonOrderedAFTs* UDT is devoted to represent NOAFT data. *FuzzyCollections* UDT is the common ancestor for all types of FC data. DFC data is represented by means of *DisjunctiveFCs* UDT instances and CFC data by instances of *ConjunctiveFCs* UDT. Finally, *FuzzyObjects* UDT is used to represent FO data.

2.2. Fuzzy Data Handling

All fuzzy types define a Fuzzy Equal comparator (FEQ) that computes the degree of fuzzy equality for each pair of instances. Each fuzzy datatype has its own implementation of this operator in accordance with its nature. Moreover, the FORDBMS provides parameters to adjust the fuzzy equality computation to the semantics of the data handled.

For AFT data, OAFT and NOAFT data subtypes, the system uses a possibility measure to implement FEQ. This comparator is defined as shown in Def. 1 and Def. 2 for OAFT and NOAFT data, respectively.

Definition 1. *(Fuzzy Equality Comparator for OAFT* *data).* Let A and B be two OAFT instances representing two possibility distributions defined over a finite reference universe $\mathscr{U}$ and modeled as fuzzy sets, μ_A and μ_B the membership functions of these fuzzy sets. The fuzzy equality $FEQ(A,B)$ of A and B is defined as:

$$FEQ(A,B) = \max_{x \in \mathscr{U}} \min(\mu_A(x), \mu_B(x)) \qquad (1)$$

Definition 2. *(Fuzzy Equality Comparator for NOAFT data).* Let A and B be two NOAFT instances representing two possibility distributions (modeled as two fuzzy sets whose membership functions are μ_A and μ_B) defined over a finite reference universe $\mathscr{U}$, where a resemblance relation $S(x,y), x, y \in \mathscr{U}$ is defined for all its elements. The fuzzy equality $FEQ(A,B)$ of A and B is defined as:

$$FEQ(A,B) = \max_{x \in \mathscr{U}, y \in \mathscr{U}} (S(x,y) \cdot \mu_A(x) \cdot \mu_B(x)) \quad (2)$$

Additionally, for OAFT data the system implements other fuzzy comparators based on the order relation of the underlying domain such as FGT (Fuzzy Greater Than), FGEQ (Fuzzy Greater or Equal), FLT (Fuzzy Less Than) and FLEQ (Fuzzy Less or Equal), using a possibility measure. Moreover, for each possibility measure based comparator of AFT datatypes, there is a necessity measure based counterpart: NFEQ, NFGT, NFEGQ, NFLT and NFLEQ. We refer the reader to Galindo's work[13] for more details on the above comparators as its particularities are beyond the scope of this work.

Finally, as in the case of AFT types, the FORDBMS defines FEQ comparators for complex fuzzy datatypes. Details on these comparators are provided in the next section.

3. Comparators for Complex Fuzzy Datatypes

Our FORDBMS provides complex fuzzy datatype structures to model complex problems from the real world. In order to properly capture the rich semantics present in such real objects, it is necessary to provide a flexible mechanism to model the way the system retrieves instances of the datatypes. In other words, the FORDBMS must provide a parameterized way to adapt the behavior of the flexible comparators on complex fuzzy datatypes instances to the semantics of the real world objects modeled by these instances.

The complex fuzzy datatypes that the FORDBMS provides are fuzzy collections (FC) and fuzzy objects (FO). The implementation of flexible comparators for each of these datatypes is not straight-forward as they must return a degree that represents the whole resemblance for each pair of datatype instances. The problem is that each instance has a complex structure and a fuzzy equality degree must be computed for each of their component values, and then, perform an aggregation of all these degrees. There are several options available for each step in the computation of the resemblance degree between a pair of complex fuzzy datatype instances. Depending on the alternative used, the semantics of the comparison may vary substantially. In this section we will provide a general definition for the flexible comparators for these datatypes.

3.1. Comparators for Conjunctive Fuzzy Collections

This fuzzy datatype models collections of elements with the same type, where each element can belong to the collection with a degree between 0 and 1. The semantics of the collection is conjunctive. The FORDBMS must provide an operator that computes to which degree an instance of a CFC data type is included into another.

3.1.1. Fuzzy Inclusion Operator

The operator `FInclusion(A,B)` calculates the inclusion degree of $A \subseteq B$, where A and B are instances of CFC. There are some proposals for this operator like the *Resemblance Driven Inclusion Degree* [5], which is defined as follows:

Definition 3. *(Resemblance Driven Inclusion Degree).* Let A and B be two fuzzy sets defined over a finite reference universe $\mathscr{U}$, μ_A and μ_B the membership functions of these fuzzy sets, S the resemblance relation defined over the elements of $\mathscr{U}$, $\otimes$ be a t-norm, and I an implication operator. The inclusion degree of A in B driven by the resemblance relation S is calculated as follows:

$$\Theta_S(B|A) = \min_{x\in\mathscr{U}} \max_{y\in\mathscr{U}} \theta_{A,B,S}(x,y) \tag{3}$$

where

$$\theta_{A,B,S}(x,y) = \otimes(I(\mu_A(x),\mu_B(y)),\mu_S(x,y)) \tag{4}$$

For some applications, like X-Rays image retrieval[2], the operator defined in Def. 3, performs better using the min as t-norm and the Gödel implication operator. However, in other applications like color based image retrieval[1], we obtain a better result if we use the Gaines implication operator defined as follows:

$$I(x,y) = \begin{cases} 1, & \text{if } x \leqslant y \\ y/x & \text{otherwise} \end{cases} \tag{5}$$

Additionally, we substitute in Eq. 3 the minimum aggregation operator by a weighted mean aggregation operator, whose weight values are the membership degrees in A of the elements of $\mathscr{U}$, divided by the fuzzy cardinality of A, $|\widetilde{A}|$, resulting in the following expression:

$$\Theta_S(B|A) = \sum_{x\in\mathscr{U}} \frac{\mu_A(x)}{|\widetilde{A}|} \cdot \max_{y\in\mathscr{U}} \theta_{A,B,S}(x,y) \tag{6}$$

where

$$|\widetilde{A}| = \sum_{x\in A} \mu_A(x). \tag{7}$$

As this kind of operations require the use of an implication operator, a t-norm and a aggregation function, we will propose a definition that provides freedom in the choice of the two first elements, and that includes the use of the well known OWA operators[14] to model the aggregation task. In this way, the particular semantics of each application can be taken into account choosing the right operators (i.e. implication, t-norm and aggregation) to compute the resemblance. This is the proposed definition:

Definition 4. *(Generalized Resemblance Driven Inclusion Degree).* Let A and B be two fuzzy sets defined over a finite reference universe $\mathscr{U}$, μ_A and μ_B the membership functions of these fuzzy sets, S the resemblance relation defined over the elements of $\mathscr{U}$, $\otimes$ be a t-norm, I an implication operator, F an OWA operator, and $K(A)$ an aggregation correction factor depending on A. The generalized inclusion degree of A in B driven by the resemblance relation S is calculated as follows:

$$\Theta_S(B|A) = \begin{cases} 1, \text{ if } A = \varnothing \\ 0, \text{ if } A \neq \varnothing \wedge B = \varnothing \\ K(A)\cdot F_{x\in A}(\mu_A(x)\cdot \max_{y\in B}\theta_{A,B,S}(x,y)), \text{ otherwise} \end{cases} \tag{8}$$

where

$$K : \widetilde{\mathscr{P}}(\mathscr{U}) \to \mathbb{R}^+ \tag{9}$$

and,

$$\theta_{A,B,S}(x,y) = \otimes(I(\mu_A(x),\mu_B(y)),\mu_S(x,y)) \tag{10}$$

When average is used as aggregation, the term *K(A)* can be used to take into account the CFC elements' membership degrees, to normalize the aggregation result of Eq. 8

Note that, if we select in Eq. 8 the min OWA operator F_* and set $K(A) = \mathit{1}$ we get:

$$\Theta_S(B|A) = F_{*x\in A}(\mu_A(x)\cdot \max_{y\in B}\theta_{A,B,S}(x,y)) \tag{11}$$

which has a similar behavior as Eq. 3 when $\mu_A(x) = 1, \forall x \in A$.

The following example, taken form Marín et. al work[5], illustrates the use of the introduced operator:

Example 1. Let $A = \{(a,0.9),(d,1)\}$ and $B = \{(a,1),(b,0.7),(c,0.9)\}$ be two fuzzy sets, where $\{a,b,c,d\}$ is the reference universe $\mathscr{U}$ over which a resemblance relation S is defined, such that $\mu_S(a,b) = \mu_S(a,c) = \mu_S(a,d) = \mu_S(b,c) = \mu_S(b,d) = 0$ and $\mu_S(c,d) = 0.7$.

According to Eq. 8, the inclusion degree of A into B is computed as follows:

$$\begin{aligned}&\Theta_S(B|A) = K(A) \cdot F_{x \in A}(\\&\mu_A(a) \cdot \max\{\bigotimes(I(\mu_A(a), \mu_B(a)), \mu_S(a,a)),\\&\ldots,\\&\bigotimes(I(\mu_A(a), \mu_B(c)), \mu_S(a,c))\},\\&\mu_A(d) \cdot \max\{\bigotimes(I(\mu_A(d), \mu_B(a)), \mu_S(d,a)),\\&\ldots,\\&\bigotimes(I(\mu_A(d), \mu_B(c)), \mu_S(d,c))\})\end{aligned} \tag{12}$$

If we use the product as t-norm, $K(A) = 1$, the min OWA operator F_* whose associated vector is, in this case, $W_* = [0,1]^T$ and, Gaines implication as the implication operator then:

$$\begin{aligned}&\Theta_S(B|A) = 0 \cdot \mu_A(a) \cdot \max\{1,0,0,0\}+\\&1 \cdot \mu_A(d) \cdot \max\{0,0,0.63,0\} = 0.63\end{aligned} \tag{13}$$

We can use Eq. 8 to express the *modified resemblance* defined in Eq. 6 by the following:

$$\begin{aligned}&\Theta_S(B|A) =\\&K(A) \cdot F_{ave\, x \in A}(\mu_A(x) \cdot \max_{y \in B} \theta_{A,B,S}(x,y))\end{aligned} \tag{14}$$

where F_{ave} is the averaging OWA operator, $K(A) = |A|/|\widetilde{A}|$ and $|A|$ stands for the *crisp* (i.e. non fuzzy) cardinality of A.

The following example illustrates the use of the previous equation to compute the *Generalized Resemblance Driven Inclusion Degree*.

Example 2. We start from the sets, resemblance relation, t-norm and implication operator used in example 1. In such case, $|A| = 2$, $|\widetilde{A}| = \mu_A(a) + \mu_A(d) = 1.9$ and $W_{ave} = [1/2, 1/2]^T$. Then, if we apply Eq. 12 with $K(A) = 2/1.9$ and the averaging OWA operator, the result is:

$$\begin{aligned}&\Theta_S(B|A) = \tfrac{2}{1.9} \cdot (\mu_A(a) \cdot \max\{1,0,0,0\}/2+\\&\mu_A(d) \cdot \max\{0,0,0.63,0\}/2) = \tfrac{0.9+0.63}{1.9} \approx 0.8\end{aligned} \tag{15}$$

As we have shown, with the use of OWA operators we can model "orness" (F^*), "andness" (F_*), average (F_{ave}), and other semantics for the aggregation.

3.1.2. Fuzzy Equality Operator

When A and B are two instances of CFC, the generalized resemblance between two fuzzy sets is calculated using the concept of double inclusion proposed by Marin et al.[5]

Definition 5. (Generalized resemblance between fuzzy sets). Let A and B be two fuzzy sets defined over a finite reference universe $\mathscr{U}$, over which a resemblance relation S is defined, and $\otimes$ be a t-norm. The generalized resemblance degree between A and B restricted by $\otimes$ is calculated by means of the following formula:

$$\beth_{S,\otimes}(A,B) = \otimes(\Theta_S(B|A), \Theta_S(A|B)) \tag{16}$$

Therefore, the implementation of the operator `FEQ(A,B)`, when A and B are instances of CFC, aggregates the results of `FInclusion(A,B)` and `FInclusion(B, A)` using a t-norm. In some cases it may be necessary to apply a factor of correction to take into account the relative cardinalities of the CFCs. This factor is defined[5] as follows:

Definition 6. (Cardinality ratio). Let A and B be two fuzzy sets. We define the cardinality ratio between A and B as a measure of the relative resemblance between their cardinalities. This ratio is calculated by means of the following formula:

$$\Phi(A,B) = \begin{cases} 1, & \text{if } A = \varnothing \wedge B = \varnothing \\ \frac{\min(|A|,|B|)}{\max(|A|,|B|)} & \text{otherwise} \end{cases} \tag{17}$$

By combining the *Generalized resemblance between fuzzy sets* and the optional *Cardinality ratio*, we define the fuzzy equality operator for two instances, A and B, of CFC as follows:

$$FEQ_{\phi=b}(A,B) = \begin{cases} \beth_{S,\otimes}(A,B) & \text{if } \phi = 0 \\ \Phi(A,B) \cdot \beth_{S,\otimes}(A,B) & \text{otherwise} \end{cases} \tag{18}$$

Published by Atlantis Press

3.2. Comparators for Fuzzy Objects

For this kind of fuzzy datatypes, our FORDBMS provides the operator `FEQ(A,B)`, that computes the resemblance of two instances of the same subclass of *FO*. The definition of the operator proposed in this section aims to provide the designer with a flexible framework to express the specific semantics of the considered problem. First, we will introduce a parameterized version of the FEQ operator for OAFT datatypes, to allow to fuzzify crisp values (in order to represent data imprecision) and to relax fuzzy values in flexible comparisons.

Definition 7. (Relaxation Function) Let $s,k \geqslant 0$ be two real numbers that represent support and kernel increments. Let A be a trapezoidal possibility distribution defined on a numerical domain $\mathscr{U}$ whose characteristic function is given by $[\alpha,\beta,\gamma,\delta]$. The relaxation function $r_{k,s} : \widetilde{P}(\mathscr{U}) \mapsto \widetilde{P}(\mathscr{U})$ of A, $r_{k,s}(A)$, is a fuzzy set whose membership function is defined by:

$$\begin{aligned}&\mu_{r_{k,s}(A)} = [\min(\alpha\cdot(1-s),\beta\cdot(1-k)),\beta\cdot(1-k),\\&\gamma\cdot(1+k),\max(\gamma\cdot(1+k),\delta\cdot(1+s))],\\&\forall A \in \widetilde{P}(\mathscr{U}) | \mu_A = [\alpha,\beta,\gamma,\delta]\end{aligned} \tag{19}$$

Note that $r_{0,0}(A) = A$.

Example 3. Let A be a trapezoidal possibility distribution given by $[30,35,45,50]$, and $s = 0.7$, $k = 0.4$ the support and kernel increments, respectively. Then, $\mu_{(0.7,0.4)}A(x)$ is given by the trapezoidal distribution:

$$\begin{aligned}&[\min(30\cdot(1-0.7),35\cdot(1-0.4)),35\cdot(1-0.4),\\&45\cdot(1+0.4),\max(45\cdot(1+0.4),50\cdot(1+0.7))]\\&=[9,21,63,85]\end{aligned} \tag{20}$$

Definition 8. (Relaxed numerical resemblance) Let A and B be two trapezoidal possibility distributions defined on a numerical domain $\mathscr{U}$, with membership functions $\mu_A(x)$ and $\mu_B(x)$, respectively, and $k,s \geqslant 0$ two real numbers that represent the kernel and support increments, respectively, then we define the relaxed numerical resemblance, $FEQ_{k,s}(A,B)$, as follows:

$$\begin{aligned}&FEQ_{k,s}(A,B) = FEQ(r_{k,s}(A),r_{k,s}(B)) =\\&\sup_{x\in\mathscr{U}}\min(\mu_{(k,s)}A(x),\mu_{(k,s)}B(x)))\end{aligned} \tag{21}$$

Note that $FEQ_{0,0}(A,B) = FEQ(A,B)$ and, because of the use of a possibility measure, $FEQ_{k,s}(A,B) = FEQ_{k,s}(B,A)$.

Definition 9. (Parameterized Object Resemblance Degree). Let C be a Class, n the number of attributes defined in the class C, $\{a_i : i \in 1,2,...,n\}$ the set of attributes of C that does not generate cycles in subclass definition, o_1 and o_2 be two objects of the class C, $o_j.a_i$ the value of the i-th attribute of the object o_j, $rel(a_i) \in [-1,1]$, a value whose absolute value means the relevance degree of the i-th attribute of the class C, and that indicates a discriminant attribute in the resemblance comparison if the degree is negative, m a parameter that establishes the minimum number of attributes whose comparison degree must be greater than 0. The Parameterized Object Resemblance Degree is recursively defined as follows:

$$OR(o_1,o_2) = \begin{cases} 1, \text{ if } o_1 = o_2 \\[1ex] FEQ_{k,s}(o_1,o_2), \text{ if } o_1,o_2 \in \text{subClassOf(OAFT)} \\[1ex] FEQ(o_1,o_2), \text{ if } o_1,o_2 \in \text{subClassOf(NOAFT)} \\[1ex] FEQ_{\phi=b}(o_1,o_2), \text{ if } o_1,o_2 \in \text{subClassOf(CFC)} \\[1ex] 0, \text{ if} \\ (\exists i \in [1,n] : OR(o_1.a_i,o_2.a_i) = 0 \wedge rel(a_i) < 0) \\ \vee(|\{OR(o_1.a_i,o_2.a_i) > 0 : i \in [1,n]\}| < m) \\[1ex] K(C)\cdot F(OR(o_1.a_i,o_2.a_i).|rel(a_i)|), \text{ otherwise} \end{cases} \tag{22}$$

where, $k,s \geqslant 0$ are two real numbers that represent the kernel and support increments for the considered class (if not defined, both take 0 as value), b a parameter indicating if the cardinality ratio correction factor is applied for CFC comparisons, F is an OWA operator, that aggregates the comparisons of attributes and has an associated weight vector $W = [w_1, \cdots, w_n]^T$, and $K(C) > 0$ an aggregation correction factor set for each Class C, whose value can be used to adjust the resemblance degree computed, taking into account the kind of OWA operator used and the relation between n, the number of attributes of C, and the sum of their respective relevance degrees $\sum_{i=1}^{n} |rel(a_i)|$.

The previous definition provides the designer with a parameterized rich framework to model the semantics of complex object comparisons. The definition offers the following design alternatives:

- To set a relaxation percentage for elements of a given subclass of OAFT, using the k and s parameters. This allows to perform flexible comparisons on crisp values that are not exactly equal, as well as to relax fuzzy data in these kind of comparisons.
- To apply or not the *cardinality ratio* on CFC comparisons.
- The possibility to determine that a given object attribute is discriminant in the whole comparison of two objects, in the sense that if the comparison of two objects returns a 0 value for this attribute, the whole object comparison must also return 0.
- To set the number of comparisons of attributes for an object that needs to be distinct from 0 to return a whole object comparison distinct from 0. For some kinds of problems it is better to return a 0 value for the whole object comparison if there are a certain number of comparisons of attributes that return 0.
- To set the relevance of each object attribute, $rel(a_i)$, in the whole object comparison.
- To choose the OWA operator F and the aggregation correction factor $K(C)$ that best matches the semantics of the modeled problem. When the semantics of the problem requires a normalized average approach, an F_{ave} OWA operator can be used in combination with a correction factor given by $K(C) = n / \sum_{i=1}^{n} |rel(a_i)|$, that takes into account the resemblance degrees in the computation. In contrast, if the semantics is more "andness" oriented, an OWA operator with this semantics and a neutral correction factor like $K(C) = 1$ can be used. There are other possible intermediate alternatives for choosing the combination of the OWA operator and the correction factor depending of the semantics of the problem.
- To set the FEQ parameters and behavior for each subclass involved in a complex object comparison.

3.3. Related work

As we have mentioned, our proposal takes as starting point Marín et al.'s model[5], and extends it to adapt the operators' behavior to the semantics of each particular application, keeping always in mind the need for a seamless integration in a FORDBMS. There exists a number of proposals in the literature to model the different aspects involved in complex fuzzy data comparison: implication operators, aggregation functions, t-norms, t-conorms, etc. In the introduction we referred to Berzal et al.[6] and the 2009 work of Marín et al.[7], which propose two extensions related to this work. However, our work is aimed at its integration into a FORDBMS, which makes this proposal select different options and implement them in a different way. Next, we analyze the most relevant aspects of our proposal with respect to the other two.

Regarding the definitions involved in the computation of fuzzy inclusion for Conjunctive Fuzzy Collections, Marin et al. in 2009[7] propose the use of the OWA operator *"almost all quantifier"* to compute the *approximate inclusion degree of B in A* into their proposal of fuzzy division. In this sense, our proposal of *Generalized Resemblance Driven Inclusion Degree* (Def. 4), that also integrates the use of OWA operators, is different from the above because it allows the use of any OWA operator that, along with the correction factor $K(A)$, makes it adaptable to a large variety of semantics.

The 2009 work of Marin et al.[7] also introduces a new proposal in addition to the one in Def. 3, the *Resemblance based tolerant inclusion degree* ($\Psi_S(A \mid B)$), whose idea is to expand the fuzzy set A. It means that the objects of the reference class U similar to an object initially present in A, are added to A. That paper also includes a fuzzy inclusion operator, based on this idea of inclusion, that uses the

OWA operator *almost all quantifier*. This way of computing the fuzzy inclusion could be studied to be incorporated into our framework.

As to fuzzy objects comparisons, Berzal et al.[6] analyze all the aspects involved and the alternatives available to model them. It presents an important study about the aggregation alternatives used to compute the compatibility degree of two fuzzy object instances taking into account the comparison degree obtained for each one of its attributes, the set of relevant attributes and its corresponding relevance degree. For such aggregation, it proposes operators based on linguistic quantifiers, such as *most* (e.g. *Most of the important attributes of the class exhibit similar values in the objects*). It also uses the *ordered weighted min* (OWmin) OWA operator that models the concept of *required majority*. To integrate the attribute relevance into the overall aggregation calculation, it uses implication operators, replacing $S_{a_i}(o_1,o_2)$ with $p_{a_i} \rightarrow S_{a_i}(o_1,o_2)$, being $S_{a_i}(o_1,o_2)$ the compatibility degree computed on the attribute a_i and p_{a_i} the relevance of the attribute a_i. It also analyzes how some of these operators work, such as the Kleene-Dienes, Gödel and Goguen implication operators.

Our proposal of *Parameterized Object Resemblance Degree* (Def. 9) for fuzzy object comparison reformulates the one shown in Marín et al.'s model[5], incorporating the use of OWA operators along with the aggregation correction factor $K(C)$ as a mechanism to model the aggregation of attribute comparisons. The attribute importance in the comparison is managed by two mechanisms: a) the *product* t-norm applied to the attribute relevance and the compatibility degree for such attribute, by means of which we can affect the weight of an attribute in the comparison; b) setting certain attributes as discriminant, requiring its compatibility degree to be greater than 0, by setting $rel(a_i) < 0$. Finally, the possibility of establishing the minimum number of attributes whose comparison degree must be greater than 0, and the integration of the mechanism for the relaxation of OAFT instance comparison, are other contributions included in our proposal that help to adjust the behavior of this operator to the problem's semantics.

4. FORDBMS Elements to Control the Comparison Behavior

Our FORDBMS implements the complex object comparison introduced in the previous section by means of the datatype structure shown in Fig. 1, where the definition and implementation of methods, constructors and operators take into account a set of parameters, stored in a specific database catalog, to determine their behavior. This section will describe this structure in relation with each kind of datatype considered.

4.1. Conjunctive Fuzzy Collections

The following are the tables with the parameters that provide the behavior of the FInclusion operator:

- `CFC_FInclusion(type_name,oper_name)`, where the first attribute identifies the subtype of CFC and the second stores the associated implementation. There is a predefined implementation labeled as: "min" (the default value) that implements the definition 1, using the Gödel implication. The designer can provide other implementations whose definitions are parameterized in the following tables.
- `CFC_FInclusion_def(type_name,oper_name,tnorm,implication,owa,ka)`, where the first two attributes identify the FInclusion operator whose parameters must be set, `tnorm` identifies the t-norm used (the min t-norm is used by default), the user can also select the product t-norm; the attribute `implication` sets the implication operator used, using by default the Gödel implication, but other operators are available in the implementation; the last attributes are usually related and define the OWA operator and the aggregation correction factor used. By default the FORDBMS provides an implementation with the values used in the Eq. 11. If the designer wants to provide his/her own OWA operator and aggregation correction factor, the adequate values must be set in the following tables:
- `OWA_def(owa_name,weight)`, this table stores all weights for each OWA operator defined.

- `Ka_def(type_name,ka_name,user_function)`, the user must define and implement a user function in the FORDBMS that computes the value of the aggregation correction factor, and set the identifier of this function in the column `user_function`.

To parameterize the FEQ operator on CFC, the FORDBMS provides the following table:

- `CFC_FEQ(type_name,tnorm,same_size)`, where the second attribute sets the t-norm used for the double fuzzy inclusion (the min by default), the third can take one of the following three values: if it is set to 1 (the default value) then the *cardinality ratio* is applied, if set to 2 no *cardinality ratio* is applied, and if set to 0 then the FEQ operator returns 0 if the compared CFC instances have different number of elements.

4.2. *Fuzzy Objects*

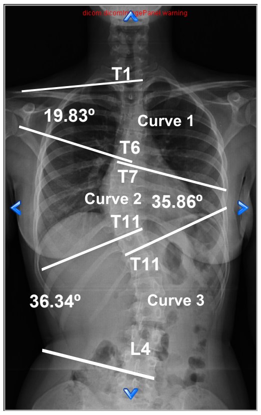
The following catalog tables are created to store the parameters that establish the behavior of the FEQ operator on instances of FO:

- `FTYPE_Relax(type_name,k,s,active)`, by means of this table the designer sets the parameters `k` and `s` that relax the instances of OAFTs subtypes in FEQ comparisons; if the attribute `active` is set to 'true' then this relaxation is applied in further FEQ comparisons, if it is set to 'false', this relaxation is not considered.
- `FO_FEQ_Aggr(type_name,owa,k_a,min_gt_0)`, this table stores the identifier of the OWA operator and the aggregation correction factor used for the subclass identified in the column `type_name`. By default, the FORDBMS implements and uses the F_{ave} OWA operator and the aggregation correction factor: $K(C) = n / \sum_{i=1}^{n} |rel(a_i)|$. The description and implementation for other operators must be set in the tables `OWA_def` and `Ka_def` described above. Besides, this table allows to establish the minimum number of attributes whose comparison degree must be greater than 0, so that the whole object comparison does not return 0.
- `FO_FEQ_Attrib(type_name,name,relevance)`, by means of this table, the designer can set the relevance values for each attribute of the considered subclass of FO, if the relevance value for a given attribute is lower than 0, this means that this attribute is discriminant.

According to the structure shown in Fig. 1, the FORDBMS provides a set of classes with methods and operators to create the necessary data structures for the considered problem, to set the behavior of each comparator and to instantiate, to store, to handle and to flexibly query the data. A detailed description of these language elements is out of the scope of this paper (for more details see Barranco's work[15]). Therefore, we consider more illustrative the use of an example to show the use of these elements.

5. An Example of Modeling Complex Fuzzy Datatypes and Flexible Comparators

To illustrate the use of the complex fuzzy datatypes handled by the proposed FORDBMS and the way the designer can adjust the behavior of comparators for them, we will use an example based on a flexible representation of the structure of a spine with scoliosis[2]. This pathology consists of a three-dimensional deformation of the spine. An anteroposterior (AP) X-ray of a spine with scoliosis projects the three dimensional deformation to a bidimensional image that shows several curves on the spine. To measure the severity of this disease, physicians measure, on the AP X-ray, the Cobb angle[16] for each curve in the spine. Each Cobb angle measurement is characterized by means of four values: angle value, superior vertebra, inferior vertebra and direction of the curve (left or right). Another parameter that characterizes a spine curve is the apical vertebra. The whole spine measurement comprises a set of curves, each one represented by the previously mentioned parameters.



Parameters of Curve 1:
Direction of the curve: "Right",
Cobb angle: 19.83º,
Superior vertebra: "T1",
Apex vertebra: "T3",
Inferior vertebra: "T6"

Parameters of Curve 2:
Direction of the curve: "Left",
Cobb angle: 35.86º,
Superior vertebra: "T7",
Apex vertebra: "T9",
Inferior vertebra: "T11"

Parameters of Curve 3:
Direction of the curve: "Right",
Cobb angle: 36.34º,
Superior vertebra: "T11",
Apex vertebra: "L2",
Inferior vertebra: "L4"

Fig. 2. Example of Cobb angle measurement for a spine with three curves. On the right, the parameter values for each curve

Figure 2 shows an example of Cobb angle measurement performed on an AP X-ray and the resulting characteristic values of the curves. For the diagnosis and treatment of scoliosis it is useful to have medical files of other similar cases. In order to gather this data, the possibility to retrieve X-rays of patients with similar parameters for the deformation of the spine is interesting. Therefore, a physician should be able to formulate queries to the FORDBMS looking for images of spines that include a given curve (or a set of curves). Note that the queries must be flexible in the sense of retrieving images with similar values for the parameters, but not exactly the same values.

5.1. Data Definition and Behavior of the Comparators Setting

Firstly, we need to create all subtypes needed to represent the structure of the data. According to Fig. 2, we use a subtype of OAFT to model the Cobb angle and subtypes of the NOAFT to model the boundary vertebrae of the Cobb angle and its direction. To model a whole curve measurement we select a FO subtype, and to model the whole spine measurement we use a CFC subtype. The definitions of these datatypes using the DDL provided by the FORDBMS, are shown bellow:

```
EXECUTE OrderedAFT.extends(’CobbAngleT’);
EXECUTE NonOrderedAFT.extends(’CurvDirectionT’);
EXECUTE NonOrderedAFT.extends(’VertbSetT’);
-- This type has two values: LEFT and RIGHT
EXECUTE NonOrderedAFT.defineLabel(
                        ’CurvDirectionT’,’LEFT’);
EXECUTE NonOrderedAFT.defineLabel(
                      ’CurvDirectionT’,’RIGHT’);
-- This set of sentences defines labels
-- for the 24 vertebrae
EXECUTE NonOrderedAFT.defineLabel(user,
                           ’VertbSetT’,’L5’);
..
EXECUTE NonOrderedAFT.defineLabel(user,
                            ’VertbSetT’,’C1’);
-- Creates the subtype for
-- a whole Cobb angle measurement
CREATE OR REPLACE TYPE CobbCurvT UNDER FO (
  Direction CurvDirectionT, Angle CobbAngleT,
  SupVertb VertbSetT,ApexVertb VertbSetT,
  InfVertb VertbSetT););
-- Creates the subtype for
-- the whole spine measurement
EXECUTE ConjunctiveFCs.extends(’SpineCurves’,
                              ’CobbCurvT’,4);
```

The type VertbSetT represents the 24 vertebrae of the spine. To perform comparisons it is necessary to provide an order relation for this set. To do this we define the following mapping: ’L5’ → 1, ’L4’ → 2 · · ’C1’ → 24. With this order relation we define a static function on the type VertbSetT to relax the proximity value of two vertebrae in FEQ comparisons. This function has the form create_nearness_vert(k,s) and generates and stores a nearness relation based on the parameters k and s. The former extends *k vertebrae* the kernel for a VertbSetT value, and the second extends *s vertebrae* the support. In this example, we will model that a given vertebra when compared with itself and adjacent ones returns 1, when compared with vertebrae located two positions away returns 0.67, 0.33 if they are three positions away and 0 if they are four or more positions away. To get this behavior we need to execute:

```
VertbSetT.create_nearness_vert(1,3)
```

The following statements set the behavior for the comparison of instances of the data structure defined:

```
-- Extends the kernel (k) and support (s)
-- in FEQ comparisons. The value ’Y’ in the
-- third parameter activates the relaxation.
-- Values are inserted into the catalog table
-- FTYPE_Relax(type_name,k,s,active)
execute orderedAFT.setRelax(’CobbAngleT’, 0.4,
                                    0.7, ’Y’);
-- Set the relevance for CobbCurvT attributes.
-- The first two are discriminant because of
-- their negative value.
-- Values are inserted into the catalog table
-- FO_FEQ_Attrib(type_name,name,relevance)
execute fo.setAttributeRelevance(’CobbCurvT’,
                              ’Direction’,-1);
```

```
execute fo.setAttributeRelevance('CobbCurvT',
                              'Angle',-1);
execute fo.setAttributeRelevance('CobbCurvT',
                              'SupVertb',1);
execute fo.setAttributeRelevance('CobbCurvT',
                              'ApexVertb',1);
execute fo.setAttributeRelevance('CobbCurvT',
                              'InfVerteb',1);
-- Set the min attributes nonzero
-- Values are inserted into the catalog table
-- FO_FEQ_Aggr(type_name,owa,k_a,min_gt_0)
execute fo.setMin_gt_0('CobbCurvT',3);
```

For the aggregation of the attributes of CobbCurvT we select the default implementation (see Sect. 4), then we do not need to set any values. The same is valid for the implementation used for the FInclusion operator. For FEQ comparisons on spine instances we want that spines with different number of curves return a 0 value. To do this, we execute the sentence:

```
EXECUTE ConjunctiveFCs.set('SpineCurves',null,0)
```

Now, we can create a table that stores instances of SpineCurves with the X-rays images:

```
create table APXRay(image# number, xray bfile,
SpineDescription SpineCurves);
```

The following sentence inserts the data related to the X-ray shown in Fig. 2:

```
Insert into apxray values (313701,
BFILENAME('APXRays','313701.gif'),SpineCurves(
1,cobbCurvT(
   CurvDirectionT('RIGHT'),
   CobbAngleT(trapezoid(18.83)),
   VertbSetT('T1'),
   VertbSetT('T3'),
   VertbSetT('T6')),
1,cobbCurvT(
   CurvDirectionT('LEFT'),
   CobbAngleT(trapezoid(35.86)),
   VertbSetT('T7'),
   VertbSetT('T9'),
   VertbSetT('T11')),
1,cobbCurvT(
   CurvDirectionT('RIGHT'),
   CobbAngleT(trapezoid(36.34)),
   VertbSetT('T11'),
   VertbSetT('L2'),
   VertbSetT('L4'))));
```

5.2. *Querying*

Using the behavior configured for FEQ on CobbCurvT and SpineCurves subtypes, we can retrieve images showing a spine curve pattern similar to a given one, for example, the image shown in Fig. 2. To perform this query we execute the following sentence:

```
SELECT ap1.image#,ap1.xray,ap1.cdeg(1)
FROM apxray ap1, apxray ap2
WHERE ap1.image#=313701 AND
FCOND(FEQ(ap1.spinedescription,
ap2.spinedescription),1)>0 order by cdeg(1) desc;
```

To process this query, the FORDBMS, taking into account the parameters defined in the previous section that determine the behavior of the fuzzy comparators involved in the query, evaluates the comparison of the collection of curves present in the image 313701 with respect to the collection of curves present in each other image. We refer the reader to Medina et al. work[17] for a step-by-step detail on the processing of this query. As can be seen in Fig. 3, the result set holds that the more exact the curve pattern matching is between each result image and the sample image, the higher the compliance degree assigned to the result image.

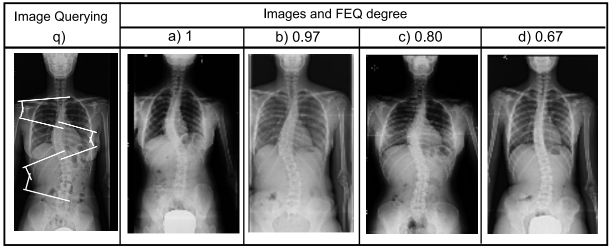


Fig. 3. Searching images that present similar spine curvature to image q).

6. Concluding Remarks and Future Work

This paper proposes a parameterized behavior for the operators FInclusion on CFC and FEQ on CFC and FO. This idea is motivated by the need to adapt the behavior of these comparators to the specific semantics of the considered applications. The FORDBMS is designed to support these changes through the implementation of a catalog that stores the parameters that define the behavior of these comparators, and defining and implementing the types, methods and operators that provides that functionality. This approach is proved useful by some example applications. Although some alternatives for the operators are implemented by default into the FORDBMS, future work will be oriented to extend the number of variants of operators supported and to implement techniques to improve the performance of retrieval operations based on these operators.

Acknowledgment

This work has been supported by the "Consejería de Innovación Ciencia y Empresa de Andalucía" (Spain) under research projects P06-TIC-01570 and P07-TIC-02611, and the Spanish (MICINN) under grants TIN2009-08296 and TIN2007-68084-C02-01.

1. Chamorro-Martínez, J., Medina, J.M., Barranco, C.D., Galán-Perales, E., Soto-Hidalgo, J.M.: Retrieving images in fuzzy object-relational databases using dominant color descriptors. Fuzzy Sets and Systems **158** (2007) 312–324
2. Medina, J.M., Jaime-Castillo, S., Barranco, C.D., Campaña, J.R.: Flexible retrieval of x-ray images based on shape descriptors using a fuzzy object-relational database. In: Proc. IFSA-EUSFLAT 2009. 20-24 july 2009. Lisbon (Portugal). (2009) 903–908
3. Hallez, A., Bronselaer, A., De Tré, G.: Comparison of sets and multisets. Internat. J. Uncertain. Fuzziness Knowledge-Based Systems **17**(suppl. 1) (2009) 153–172
4. Dubois, D., Prade, H.: A unifying view of comparison indices in a fuzzy set-theoretic framework. In: Recent Developments in Fuzzy Set and Possibility Theory. Pergamon Press (1982) 3–13
5. Marín, N., Medina, J., Pons, O., Sánchez, D., Vila, M.: Complex object comparison in a fuzzy context. Information and Software Technology **45**(7) (2003) 431–444
6. Berzal, F., Cubero, J.C., Marín, N., Vila, M.A., Kacprzyk, J., Zadrozny, S.: A general framework for computing with words in object-oriented programming. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **15**(Supplement-1) (2007) 111–131
7. Marín, N., Molina, C., Pons, O., Vila, M.A.: Semantically-driven flexible division in fuzzy object oriented models. In: IFSA/EUSFLAT Conf. (2009) 1039–1044
8. Cubero, J., Marín, N., Medina, J., Pons, O., Vila, M.: Fuzzy object management in an object-relational framework. In: Proc. 10th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'2004. (2004) 1767–1774
9. Barranco, C.D., Campaña, J.R., Medina, J.M.: Towards a fuzzy object-relational database model. In Galindo, J., ed.: Handbook of Research on Fuzzy Information Processing in Databases. IGI Global (2008) 435–461
10. Stonebraker, M., Moore, D.: Object-Relational DBMSs: The Next Great Wave. Morgan Kaufmann (1996)
11. Melton, J.: Advanced SQL:1999 Understanding Object-Relational and Other Advanced Features. Morgan Kaufmann (2003)
12. Eisenberg, A., Melton, J., Kulkarni, K., Michels, J.E., Zemke, F.: Sql:2003 has been published. SIGMOD Rec. **33**(1) (2004) 119–126
13. Galindo, J., Urrutia, A., Piattini, M.: Fuzzy Databases: Modeling, Design and Implementation. Idea Group Publishing, Hershey, PA, USA (2006)
14. Yager, R.: Families of owa operators. Fuzzy Sets and Systems **59** (1993) 125–148
15. Barranco, C.D.: Bases de datos objeto-relacionales difusas: modelo, arquitectura y aplicaciones. PhD thesis, University of Granada (2010)
16. Cobb, J.: Outline for the study of scoliosis. Am Acad Orthop Surg Inst Course Lect **5** (1948) 261–275
17. Medina, J.M., Jaime-Castillo, S., Jiménez, E.: A dicom viewer with flexible image retrieval to support diagnosis and treatment of scoliosis. Expert Systems with Applications **39** (2012) 8799 – 8808