# Fuzzy Domains with Adaptable Semantics in an Object-Relational DBMS

J.T. Cadenas[1], N. Marín[2⋆], and M.A. Vila[2]

[1]Department of Computation and I.T.,
Simón Bolívar University, 1080A, Caracas, Venezuela
[2]Intelligent Databases and Information Systems Research Group,
Department of Computer Science and A.I.,
University of Granada, 18071, Granada, Spain
jtcadenas@usb.ve,{nicm,vila}@decsai.ugr.es
http://idbis.ugr.es

**Abstract.** In this paper, we focus on the management of the user's context in the field of Fuzzy Database applications. Concretely, we propose an approach that uses fuzzy techniques to represent vague concepts with an adaptable semantics in order to meet user's preferences. We enhance previous work on the design and implementation of fuzzy data models, allowing a flexible use of different sets of linguistic labels for the same domain, which are transparently chosen according to user's context. This way, applications that operate on fuzzy databases get an additional piece of semantic power that makes possible an automatic customization in relation to user's preferences. This initial capability is proposed within a general architecture that aims at the management of the user context from a more general and interdisciplinary point of view. We complete our contribution with the description of the implementation of our proposal using the Oracle Object-Relational DBMS.

**Keywords:** Fuzzy Databases, Adaptable Semantics, User's Context, Object-Relational DBMS

## 1 Introduction

Real world data are frequently imperfect (vague and/or imprecise) and this imperfection makes data management difficult in conventional database management systems (DBMS). Fuzzy Sets Theory of Zadeh [22] has proven to be a useful tool in order to manage this data imperfection as demonstrated in de Tré and Zadrozny [19] and Bosc et al. [5]. As a consequence, fuzzy database modeling and processing are important areas in database research.

In the literature, there are significant proposals using the relational data model, e.g. [11], [6], [13]. Additionally, there are also remarkable proposals for

⋆ The research reported in this paper was partially supported by the Andalusian Government (Junta de Andalucía, Consejería de Economía, Innovación y Ciencia) under project P07-TIC03175.

fuzzy objects oriented models, e.g. [21], [4], [16], [2], [18]. In [15] the interested reader can find a recent overview of Fuzzy Conceptual Data Modeling; [10] can also be used for a compendium of research on Fuzzy Information Processing in Databases.

The extension of object-orientation in order to deal with imperfect data can also be considered in modern programming platforms [3], [1]. In the recent years, fuzzy extensions have also been considered in Object-Relational DBMS like Oracle [17] and PostgreSQL [9].

Part of the success of almost all the mentioned approaches is due to the use of linguistic labels, in order to represent imperfect values stored in the databases or to make possible the use of terms in queries that resemble the natural language used by humans (flexible querying). The link between the linguistic labels and the representation of imperfect values in the system, is provided by the definition of the semantics according to well known results of the Fuzzy Sets Theory.

Although the proposals cited above have proven to be suitable to address the representation and manipulation of the imperfection in the field of information systems, the semantics for fuzzy concepts is unique and shared by all the database users, and this fact is a clear drawback from a more human centered point of view: that is, the semantics defined to represent and manipulate imperfection is determined by the system designer. However, many other users may have to interact with the designed database; in this sense, it is clear that semantics largely depends on each different user or, moreover, on each user's context. Therefore, this semantics must appropriately change in order to suitably adapt to user's preferences.

For example, a user $A$ inserts data related to a *big* object that can then be offered to another user $B$ when asking for *medium* objects. Moreover, semantics can change even for the same person depending of his/her interest; i.e., John is 1.75 meters tall, a group of persons would say *John is medium height*, but the criteria may change if he is being selected to ride horses as a jockey or to play basketball; then, the same person would label John as *tall* in the first case and as *short* in the second situation. That is, linguistic labels are relative to the user context. In fact, it is assumed the absence of universal meaning in most of the descriptive terms used by humans for use in database systems.

In our opinion, if we want our fuzzy database systems advance from the human friendliness perspective, we will have to provide tools to represent vague concepts that have the capability of adapting its semantics to the user context and to do it in a transparent manner.

Some proposals focused on this issue can be found in the literature. Though they are initial approaches to this challenging problem, some interesting results are offered. The extent of user queries with preferences deduced from fuzzy rules that describe the current user context is given by Hadjali et al. [12]; this work shows the importance of context and customization of applications in database environments. Additionally, Ughetto et al. [20] presents techniques of data summaries and customized systems, allowing users to choose their own vocabulary when they are querying a database, using linguistic variables defined for each user's profile and getting the required level of granularity in results.

The approach we introduce in this paper aims at the customization of the fuzzy database systems according to the necessities of each user at every moment. Hence, two important contributions are the representation of fuzzy information regarding the user context; and, the provision of flexibility in database querying based on the user's preferences, including linguistic terms with adaptable semantics in human centered queries, retrieving results with matching degrees.

We focus on these issues and provide a first progress in this direction: by means of techniques of representation and manipulation of imperfect objects based on the theory of Fuzzy Sets, we present an approach to build different fuzzy domains in a database whose semantics is automatically adapted to the user, eliminating the need to define the desired semantics at the moment of database querying.

This is an initial contribution within a more challenging goal: we want that our fuzzy database management system adaptively respond to the user context, with context understood in a broader sense. For this reason, our approach is integrated within an architecture for building a system with these characteristics that is also described in this paper.

The work is organized as follows: next section presents a general architecture for a fuzzy database system based on user context. Section three is focused on the representation of fuzzy domains with adaptable semantics in the framework of the proposed architecture. Section four is deserved to describe a proof of concept implementation of this capability within a conventional ORDBMS as Oracle; in this section is showed the transparent use of our model through sentences of standard SQL:99, but with adaptive semantics based on the user's preferences. Finally, section 5 concludes the paper and presents some guidelines for future work.

## 2 A General Architecture for a Fuzzy Database System Adaptive to the User's Context

Nowadays, people have to use more and more effectively the Information and Communication Technologies (ICT). To achieve this effectiveness these technologies must adapt to users as much as possible: ICT systems must be sensitive to the user's context.

Most sensitive systems are developed by Ubiquitous Computing at low level, closely tied to implicit inputs gathered by sensors, without regard things such as desires, objectives and the user's emotional state. These nuances related to the context are defined in a higher level of abstraction.

Moreover, uncertainty and ambiguity are present in various aspects of the user's context, such as the semantics of the used concepts, data captured by devices (sensors) or inferred through diverse methods. These facts produce a wide survey of challenges not only in the representation of information but also in the context modeling using Fuzzy Databases and the development of context-aware systems.

To the best of our knowledge, most Information System are designed to be used mostly independently of user's context; the programmer and/or designer imposes a built-in criteria that is unable to take into account the multiple future contexts that users may have: who the user is, when and where the application works, the level of expertise of the user (beginner/expert), whether he/she is alone or in company, among others.

Usually, in all these situations, the software is designed to works in the same way; the programmers make certain assumptions that are suitable for a given pre-defined context, but may not result very appropriate in the user's current context. The problem of efficiency in retrieving information of interest for the user is even more crucial in current information systems, due to the increasing volume of information available.

A way to address this problem is developing context-aware systems that manage more knowledge about the user and his/her context. *Doing the right thing entails that it be right given the user's current context* [14].

The proposal described in this section is intended to be the base for the development of context-aware systems in environments with imperfect data managed using Fuzzy Databases.

Diverse forms of data imperfection involve the notion of graduality [2], moreover the semantics is diverse, i.e. depends on the user's context. Who can set universal limits (even fuzzy) to determine if a person is *High*? This term may be influenced by the location of the user (Japan, Spain, or Venezuela), objectives (looking for a gymnast, a jockey, or a basketball player) or others factors that may be sensitive to the context and affect the semantics of the concepts.

In the broadest sense, a context-aware system must provide mechanisms for acquiring implicit inputs provided by the environment, either through environmental sensors or software agents inferring the context from people activity; furthermore, it must take into account explicit user inputs regarding the context. In Fig. 1 we present a architecture for a context-aware information system that is capable of manage explicit and implicit data regarding the user context. The architecture is founded on two main pillars: Ubiquitous Computing (UbiComp) in order to transparently deal with implicit information about user's context and Computational Theory of Perceptions(CTP) of Zadeh [23] in order to manage data imperfection and a flexible system fit to user's requirements according to the context.

The elements of the figure are:

- User: it is the main beneficiary of this user-oriented architecture. In an ideal situation, the interaction of the user with the system must involve only data and commands related to information system functionalities that must transparently adapt according to user's context inferred thanks to UbiComp technologies.
- Context-aware applications: the interaction of the user with the database is usually done through the use of applications. In our architecture, these

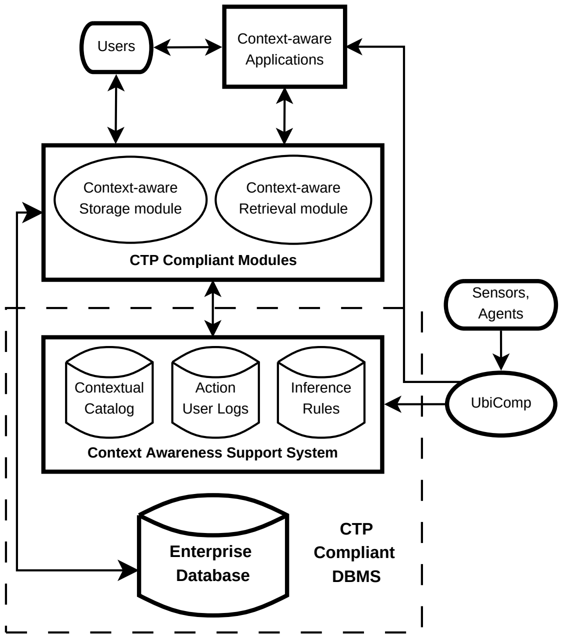


**Fig. 1.** Architecture for a Fuzzy Database system adaptive to user context

applications must be built so that they can easily adapt to user needs according to his/her context. In some cases, these applications can also get context information about the user, when the implicit information obtained by the ubiComp systems is not enough.

– The data flow from user to database must be done through CTP compliant storage (*CAS*) and retrieval (*CAR*) modules. These modules properly translate data from the user language to the system representation (and vice versa) according to the user's context. CTP techniques have a main impact in this part of the system as a suitable tool for data imperfection representation and handling in natural language.
– Below these layers, we can find the context-awareness support system. This layer offers support for context-awareness founded on the use of context information obtained from the UbiComp system, the applications, the user, and a suitable logic to conveniently adapt system functionalities to the context. This part of the architecture is composed by:
  - Contextual catalog: metadata regarding the representation and management of user's context.
  - Inference rules: to express contextual data referred to user's preferences.
  - Action user logs: data regarding user interaction with the system, specially useful in order to infer information about user's context and how to properly adapt to user's requirements according to the context.

– Enterprise database: conventional data storage on a DBMS. It provides data to CTP compliant Modules to manage human nature information.

In order to implement the representation requirements of context-awareness support system and the enterprise database, we can use a CTP compliant DBMS because this kind of enhanced DBMS provides both data management functionalities and data imperfection handling.

We use an Object Relational Database Management System (ORDBMS) like Oracle due to its capability to represent complex objects, to extend the native data types, the variety of methods, and its interoperability with other programming languages besides SQL (Java, C, C + +, PL / SQL). It allows to make intuitive applications in a reasonable time, user-friendly, while preserving the inherent capabilities of a relational database (concurrent access, multiple levels of security, high performance, protecting the consistency and integrity of the data).

In summary, users will benefit as they will interact with customized systems with adaptable semantics based on the context. The system will be able to capture perceptions of the real world as closely as possible, will be used to manage imperfect data, and will allow context-aware queries based on the users preferences with adaptable semantics; moreover, the system will have the capability of inferring and discovering contextual data; thus improving the quality of access to user's information through flexible and intelligent applications.

## 3 Fuzzy Domains with Adaptable Semantics

In this section, we explain how to implement a simplified version of the CTP Compliant Modules through stored procedures that hide to the user the management of an adaptable semantics in the concepts. To achieve this goal, we designed a fuzzy object oriented UML-based model of an example application (see Fig. 2). We implement a fuzzy domain according to the proposal of [8]. We extend the data types of Oracle ORDBMS and we enhance previous work to allow flexible queries, using the standard SQL:99. The interested reader can find a detailed description about how to implement different fuzzy domains in [7].

Fuzzy Domains with adaptable semantics are given by a set of linguistic labels represented by trapezoidal membership functions for a linguistic variable such as height of a person. The system designer creates object types using the ORDBMS Oracle, accordingly to the proposed model. Then, object tables and/or object attributes (i.e. trapezoid) based on the object types are created.

In this naive proof of concept implementation, the user has three attributes: iduser, nickname, and defaultUser. Both terminal users and designer can define linguistic labels that are stored into the Contextual Catalog. Each linguistic label is associated to a given user and a fuzzy domain, and it is described by a *Trapezoid* object. Trapezoid objects can be compared by means of a member function Fuzzy Equal (FEQ).

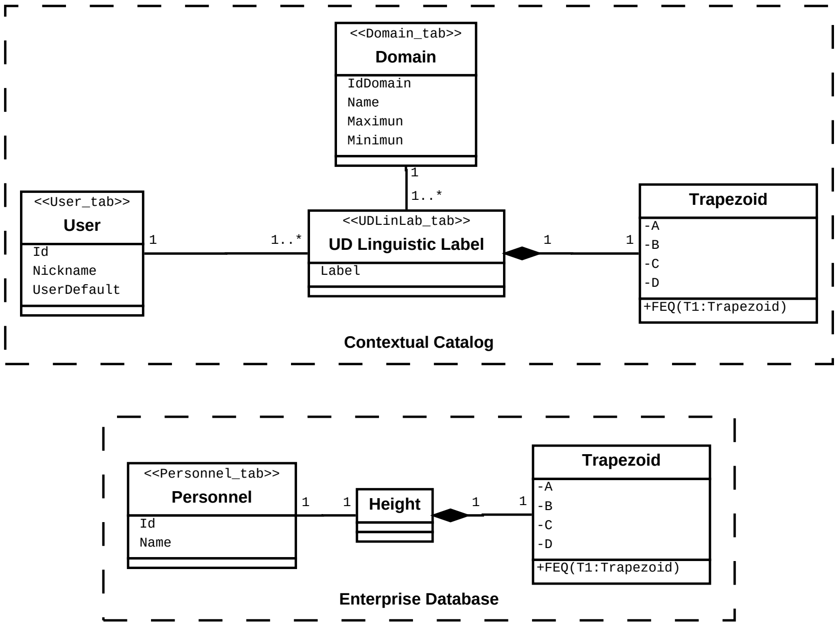


**Fig. 2.** Object-Oriented Fuzzy Data Model of a Context-Aware Application

The Enterprise Database contains a table Personnel with, among other, three attributes: id, name, and height. Each person height is described by a trapezoid of the domain Height.

Users will interact with the system through SQL standard language provided by the ORDBMS Oracle. Some stored procedures and member functions are available for managing contextual data. Below, some representative sentences are showed in relation with this example.

The first step is to input data relative to this simplified version of the *Contextual Catalog*. As we say, each user is able to create his/her own set of preferred linguistic labels. This information makes possible the implementation of adaptability through the *Context-Aware Storage* or *CAS* module. We present a DML statement with the corresponding *CAS* procedure (see Table 1). This procedure is able to identify the user that is interacting with the database when he/she connects to Oracle database, i.e. *USER1*.

Another user may connect to the database, i.e. *USER2*, and runs the following inserts.

```
INSERT INTO PERSONNEL_TAB VALUES(Id_seq.nextval,'Luis Perez',
  CA_New_Height('Average',180,200,210,220));
INSERT INTO PERSONNEL_TAB VALUES (Id_seq.nextval,'Paco Martinez',
  CA_Height('High'));
```

**Table 1.** Context Aware Storage (CAS)

| **CAS Insert** | **Description** |
|---|---|
| INSERT INTO Personnel_tab VALUES(Id_seq.nextval,'Jose Tomas Cadenas', CA_Height('Low')); | Look for Low value on UDLinLab Table for current user. If defined, insert the trapezoid in Personnel_tab; if not, take the default designer's trapezoid. |
| INSERT INTO Personnel_tab VALUES(Id_seq.nextval,'Manuel Fernandez', CA_NEW_Height ('Average',150,175,185,200)); | Create or replace the definition of Average for current user in LinLab table. Insert the trapezoid as value for Height in Personnel_tab |

Then, a new definition for label *Average* of domain *Height* according to the *USER2* is inserted into the Contextual Catalog, and the tuple regarding *Luis Perez* is inserted into personnel with the corresponding trapezoid.

The second insertion, related to *Paco Martinez* is inserted into Personnel table using the trapezoid defined by default for High, because this label has not been previously defined by USER2 and no new semantics is provided.

Below, we show data stored into the Contextual Catalog (see Table 2) after a few number of insertions.

**Table 2.** Contextual Catalog Data

| **User** | **Domain** | **Label** | **Trapezoid** |
|---|---|---|---|
| USER_DEF | Height | Low | (0,0,150,160) |
| USER_DEF | Height | Average | (150,160,170,180) |
| USER_DEF | Height | High | (170,180,300,300) |
| USER1 | Height | Low | (0,0,150,160) |
| USER1 | Height | Average | (150,175,185,200) |
| USER1 | Height | High | (190,210,300,300) |
| USER2 | Height | Low | (0,0,155,175) |
| USER2 | Height | Average | (180,200,210,220) |
| USER2 | Height | High | (170,180,300,300) |

In summary, we can create logical data structures and a stored pre-defined logic to implement compliant CTP modules so that context management can be hidden to the user. The system designer can make appropriate adjustments in accordance with particular applications, e.g. he/she can incorporate diverse user's roles and profiles, generalizing the proposed idea according to figure 1.

## 4 Context-Aware Querying using the ORDBMS Oracle

After populating the database, we show some queries in order to exemplify the implementation of the *Context-Aware Retrieval* (*CAR*) CTP compliant module.

At this moment, the database can be queried with semantics of linguistic labels linked to the current user. Below we show the DML statements that a user, i.e. *USER1*, has to execute in order to find those people that have *average* height according to USER1 semantics.

```
connect USER1
SELECT Name_person, P.Height.FEQ('Average') FEQ
FROM Personnel_tab P
WHERE P.Height.FEQ('Average')>0.5;

NAME_PERSON                 FEQ
-------------------- ----------
Manuel Fernandez              1
Tomas Cadenas               .86
Macringer Montero             1
Maria Rodriguez             .56
Luis Perez                  .57
Paco Martinez               .57
```

Another user, i.e. *USER2*, runs exactly the same statement from his session, obtaining a different output showed below.

```
connect USER2
NAME_PERSON                 FEQ
-------------------- ----------
Macringer Montero             1
Manuel Fernandez            .57
Luis Perez                    1
Paco Martinez                 1
Luis Rivera                   1
```

In summary, any user is able to make queries according to his/her preferences of height with an adaptable semantics.

We implement a method to obtain the trapezoid in a friendly way from Personnel table (*CA_Show*). Below show an example with USER1 connected.

```
connect USER1
SELECT Name_person, P.Height.Show() Height
FROM Personnel_tab P

NAME_PERSON                              Height
-------------------- ---------------------
Jose Tomas Cadenas          (0, 0, 150, 160)
Manuel Fernandez      (150, 175, 185, 200)
Mariamni Cadenas            (0, 0, 150, 160)
```

```
Tomas Cadenas         (150, 160, 170, 180)
Macringer Montero     (170, 180, 300, 300)
Maria Rodriguez           (0, 0, 155, 175)
Luis Perez            (180, 200, 210, 220)
Paco Martinez         (170, 180, 300, 300)
Luis Rivera           (190, 210, 300, 300)
```

The CTP compliant retrieval module also includes a Context Aware method for Height person (*LShow*) allowing to hide the trapezoid defined by an user and retrieving the convex combinations of linguistic labels according to his/her own pre defined semantics.

Below we have two examples with different users

```
connect USER1
SELECT Name_person, P.Height.LShow() Height
FROM Personnel_tab P

NAME_PERSON                                   Height
-------------------- -------------------------------
Jose Tomas Cadenas               .29/Average; 1/Low;
Manuel Fernandez      1/Average; .29/High; .29/Low;
Mariamni Cadenas                 .29/Average; 1/Low;
Tomas Cadenas                   .86/Average; .5/Low;
Macringer Montero                 1/Average; 1/High;
Maria Rodriguez                 .56/Average; .5/Low;
Luis Perez                      .57/Average; 1/High;
Paco Martinez                     1/Average; 1/High;
Luis Rivera                     .29/Average; 1/High;
```

Another user, i.e. *USER2*, can run exactly the same statement from his session, obtaining a different output showed below.

```
connect USER2
NAME_PERSON                                   Height
-------------------- -------------------------------
Jose Tomas Cadenas                           .5/Low;
Manuel Fernandez      .56/Low; .57/Average; 1/High;
Mariamni Cadenas                             .5/Low;
Tomas Cadenas                       .83/Low; .5/High;
Macringer Montero      .17/Low; 1/Average;  1/High;
Maria Rodriguez                     1/Low; .17/High;
Luis Perez                        1/Average; 1/High;
Paco Martinez          .17/Low; 1/Average;  1/High;
Luis Rivera                       1/Average; 1/High;
```

## 5 Conclusions and Future Work

In this paper, we have proposed a fuzzy approach to represent vague concepts with a user-adaptable semantics in an object-relational database as a first step towards a context aware database management system. We have introduced a general architecture for a fuzzy database system based on user context and we have described how the representation of fuzzy domains with adaptable semantics can be faced in the framework of the proposed architecture.

As future work, we have to consider the development of the complete set of CTP compliant storage and retrieval modules of our arquitecture and the extension of this initial capability beyond a more complete context model, based not only on the user, but on other features that describes the execution context, e.g. provided by ubiquitous computing sensors.

## References


1. F. Berzal, J.C. Cubero, N. Marín, M.A. Vila, J. Kacprzyk, and S. Zadrozny. A general framework for computing with words in object-oriented programming. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(Supplement-1):111–131, 2007.
2. F. Berzal, N. Marín, O. Pons, and M. A. Vila. Managing fuzziness on conventional object-oriented platforms. *Int. J. Intell. Syst.*, 22:781–803, 2007.
3. F. Berzal, N. Marín, O. Pons, and M.A. Vila. Development of applications with fuzzy objects in modern programming platforms. *Int. J. Intell. Syst.*, 20(11):1117–1136, 2005.
4. G. Bordogna, G. Pasi, and D. Lucarella. A fuzzy object-oriented data model for managing vague and uncertain information. *International Journal of Intelligent Systems*, 14(7):623–651, 1999.
5. P. Bosc, D.H. Kraft, and F.E. Petry. Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets and Systems*, 156(3):418–426, 2005.
6. P. Bosc. and O. Pivert. *Knowledge Management in Fuzzy Databases*, chapter SQLf query functionality on top of a regular relational DBMS, pages 171–190. Springer-Verlag New York, Inc., 2000.
7. J.T. Cadenas, N. Marín, and M.A. Vila. Gestión de dominios difusos permitiendo semantica adaptable basada en el contexto del usuario utilizando un sistema de base de datos objeto relacional. Technical report, Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, 2010.
8. L. Cuevas. *Modelo Difuso de bases de datos objeto-relacional: propuesta de implementación en software libre.* PhD thesis, Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada, 2001.
9. L. Cuevas, N. Marín, O. Pons, and M.A. Vila. pg4db: A fuzzy object-relational system. *Fuzzy Sets Syst.*, 159:1500–1514, 2008.
10. J. Galindo, editor. *Handbook of Research on Fuzzy Information Processing in Databases.* IGI Global, 2008.
11. J. Galindo, J. Medina, O. Pons, and J. Cubero. A server for fuzzy sql queries. In T. Andreasen, H. Christiansen, and H. Larsen, editors, *Flexible Query Answering Systems*, volume 1495 of *LNCS*, pages 164–174. Springer Berlin / Heidelberg, 1998.

12. A. Hadjali, A. Mokhtari, and O. Pivert. A fuzzy-rule-based approach to contextual preference queries. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Computational Intelligence for Knowledge-Based Systems Design*, volume 6178 of *LNCS*, pages 532–541. Springer Berlin / Heidelberg, 2010.
13. J. Kacprzyk and S. Zadrozny. Computing with words in intelligent database querying: standalone and internet-based applications. *Inf. Sci.*, 134(1-4):71–109, 2001.
14. H. Lieberman and T. Selker. Out of context: Computer systems that adapt to, and learn from, context. *IBM Systems Journal*, 39(3&4):617–, 2000.
15. Z.M. Ma and L. Yan. A literature overview of fuzzy conceptual data modeling. *Journal of Information Science and Engineering*, 26(2):427–441, 2010.
16. N. Marín, J.M. Medina, O. Pons, D. Sánchez, and M.A. Vila. Complex object comparison in a fuzzy context. *Information and Software Technology*, 45(7):431–444, 2003.
17. J.M. Medina, C.D. Barranco, J.R. Campaña, and S. Jaime-Castillo. Generalized fuzzy comparators for complex data in a fuzzy object-relational database management system. In E. Hüllermeier, R. Kruse, and F. Hoffmann, editors, *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, volume 81 of *Communications in Computer and Information Science*, pages 126–136. Springer Berlin Heidelberg, 2010.
18. N.B. Ozgur, M.K., and A. Yazici. An intelligent fuzzy object-oriented database framework for video database applications. *Fuzzy Sets and Systems*, 160(15):2253–2274, 2009.
19. G. De Tré and S. Zadrozny. The application of fuzzy logic and soft computing in information management. *Fuzzy Sets and Systems*, 160(15):2117–2119, 2009.
20. L. Ughetto, W.A. Voglozin, and N. Mouaddib. Database querying with personalized vocabulary using data summaries. *Fuzzy Sets and Systems*, 159(15):2030–2046, 2008.
21. N. Van Gyseghem and R. de Caluwe. *Fuzzy and Uncertain Object-Oriented Databases, concepts and Models*, volume 13, chapter The UFO Model: dealing with Imperfect Information, pages 123–185. World Scientific Publishing, 1997.
22. L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.
23. L.A. Zadeh. A new direction in ai: Toward a computational theory of perceptions. *AI Magazine*, 22(1):73–84, 2001.