FUDEM : A Fuzzy Deductive Model

Ana Isabel Aguilera Faraco*, Leonid José Tineo Rodríguez
Departamento de Computación, Universidad Simón Bolívar
Apartado 89000, Caracas 1080-A, Venezuela
Phone: +58 2 9063269, Fax: +58 2 9063243
email:leonid@ldc.usb.ve
*Departamento de Computación, FACYT, Universidad de Carabobo
Apartado 5053, Naguanagua 2005, Venezuela
Phone: +58 41 244336. Fax: +58 41 677770
email:{aaguilef, asubero}@thor.uc.edu.ve

ABSTRACT: In this paper we present a Fuzzy Deductive Database Model, which makes more flexible the conventional Deductive Database Model. This model aims to solve the problem of flexible handling on precise data by defining a new query language and by extending the extensional and intentional databases. The classic specifications of queries and rules are extended to handle expressions with qualitative properties by means of the use of linguistic labels, hedges, comparators and quantifiers, in a transparent way. Besides, the user can define the precision at which the conditions involved in a query are satisfied, in order to obtain discriminated answers. Others capabilities of flexible deduction are defined through fuzzy rules.

KEYWORDS: Fuzzy Quering, Deductive Databases, Database Model.

INTRODUCTION

The information access is a strategic element in today's society. However, the important thing is not the amount of information you can access, but the quality of the mechanisms we have, in order to get the information required in a given moment. In spite of efforts, the representation of the information and its treatment, are still far from the mechanisms commonly used by the human beings. Some authors have been making big efforts to introduce complex data models capable to correct these lacks. Some of the main aspects that characterize these approaches are the representation and manipulation of information whose semantics is close to the human schemes of reality representation and the introduction of mechanisms that make the system capable to infer new information from the stored one.
The first aspect makes us incorporate capabilities to represent and manipulate fuzzy and uncertain information into data models. From these enriched database models we have obtained numerous contributions, mainly related to the popular relational model. Some authors have extended these models by representing flexible information and expressing flexible queries, Buckles (1982), Medina (1993), Prade (1984), Zemankova (1984); others have extended it by obtaining information in a flexible form from data stored in a precise way, Bosc (1995). All of them use as Possibilistic or/and Similarity formal models without considering the deductive ones. The second aspect implies the integration of a wide range of automatic reasoning mechanisms. Some studies have been focused in designing and implementing deductive databases models, Derr (1993), Naqvi (1989), Ramakrishman (1992), Vaghani (1992). Their main contribution is the possibility of representing the data and their dependencies in a simple way by the definition of relations by comprehension (a set of rules or logical programs), including those relations based on recursion. This has produced a set of tools for the manipulation and deduction of knowledge. There is very little work done about models which offer the possibility of representing and recovering fuzzy information and allowing the inference of new information, Parsons (1994), Pons (1997). Some of these models do not deal in a transparent way with the flexible elements that should be considered in a model sufficiently abstract to allow minor perturbations in an interpretation near of the real world. The model presented in the next section tries to integrate, in the same framework, the advantages of approaches studied so far, by eliminating the inconveniences and constraints of the other models. It is based on fuzzy set theory and uses similarity relations as more natural and expressive representation tool.

COMPONENTS OF FUDEM

FUDEM is composed of two main parts, Aguilera (1998). The first one contains the classical specifications of data definition, data manipulation and deduction capabilities through precise queries and rules. The second part considers flexible extensions on the above.

The flexible extensions to the deductive model are made in three levels:

- On the extensional database (vagueness in facts)
- On the intentional database (vagueness in rules)
- On the query language (expression of vague requests)

The queries defined in this environment will contain flexible elements such as fuzzy predicates (representing gradual properties), linguistic modifiers, comparators and quantifiers, fuzzy operators and levels of satisfaction. The expression of fuzzy rules will offer the prospective deductive capabilities. We will develop these three extensions in the following sections of this paper. Formally, these extensions will be sustained on the definition of a Fuzzy Logical Language, which is presented below.

Definition 1: A **Fuzzy Logical Language** *L* is defined as *L* = *<Lc, Lr, Lt, Lm, Lq>*. Where *Lc* is a set of constant called symbols, *Lr* is a set of symbols called predicates, *Lt* is he a set of symbols called labels, *Lm* is a set of symbols called modifiers and *Lq* is a set of symbols called quantifiers. The set of the predicate symbols *Lr* is partitioned in *Lrp* (the set of predefined predicates), *Lrc* (the set of fuzzy comparators), *Lrd* (the set of fuzzy predicates or linguistic labels), *Lre* (the set of predicates defined by extension) and *Lrr* (the set of predicates defined by rules).

FUZZY EXTENSION TO THE DEFINITION COMPONENT

FUZZY PROCESSING SUSCEPTIBLE ATTRIBUTES

These are predicate attributes that, by their nature, admit some type of flexible manipulation. The metapredicate Fuzzy is defined to indicate this characteristic of the attributes.

Definition 2: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Npred Î Lre È Lrr*, *Natrib Î S(Npred)*, *Nrep Î {fuzset, relsem}*. We define the metapredicate ***fuzzy*** as *fuzzy(Npred, Natrib, Nrep).* Where the argument *Natrib* of *Npred* is susceptible of fuzzy processing, and is processed according to *Nrep* (*fuzset* for fuzzy sets and *relsem* for similarity relations).

Example: Having the following scheme: *employee (name, age, aptitude, pay)* the attributes *age*, *pay* and *aptitude* are susceptible to fuzzy processing, which is expressed as *fuzzy(employee, age, fuzset), fuzzy(employee, aptitude, relsem), fuzzy(employee,* pay, *fuzset).*

TOOLS FOR MANIPULATING THE ATTRIBUTES WITH FUZZY PROCESSING

The tools used for manipulating the fuzzy processing susceptible attributes are the fuzzy sets and the similarity relations, Zadeh (1971). The fuzzy sets are defined by means of the metapredicate *fuzset* and the similarity relations by the metapredicate *relsem*.

Definition 3: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Npred Î Lre È Lrr*, *Natrib Î S(Npred), Netiq Î Lrd, fuzzy(Npred, Natrib, fuzset), A,B,C,D Î Lc.*We define the metapredicate **fuzset** as *fuzset(Npred, Natrib, Netiq, A, B, C, D).* Where *Netiq* represents a fuzzy set expressed by the trapezoid [*A,B,C,D*] that applies on the attribute *Natrib* of the predicate *Npred*. In case that some of the parameters of the trapezoid is infinite, this one is represented by the constant "*inf* ".

Example: We want to define on the attributes *age* and *pay* with the properties represented by the fuzzy set *young*, *middle*, *old* and *low*, *medium* , *high*, respectively.

fuzset(employee, age, young, 0, 16, 30,40).
fuzset(employee, age, middle, 25, 35, 45, 55).
fuzset(employee, age, old, 40, 50, 65, 80).
fuzset(employee, pay, low, 50, 65, 85, 95).
fuzset(employee, pay, medium, 85, 95, 110, 130).
fuzset(employee, pay, high, 110, 130.180, inf).

Definition 4: Let *L = <Lc, Lr, Le, Lm, Lq>* be fuzzy a logical language, *Npred Î Lre È Lrr*, *Natrib Î S(Npred), Netiq Î Lrd, fuzzy(Npred, Natrib, relsem), Esc1,Esc2 Î Lc.*We define the metapredicate ***relsem*** as *relsem(Npred, Natrib, Esc1, Esc2, V), w*hich represents the similarity relations that can be established between the values of the attribute domain. *Esc1, Esc2* are the names of values whose similarity we want to establish. *V* is the grade of similarity between *Esc1* and *Esc2*.

Example: Similarity relation defined on the attribute aptitude.

relsem(employee, aptitude, bad, fair, 0.8).
relsem(employee, aptitude, bad, good, 0.5).
relsem(employee, aptitude, bad, excellent, *0.1).*
relsem(employee, aptitude, fair, good, 0.7).
*relsem(employee, aptitude, fair, e*xcellent*, 0.5).*
relsem(employee, aptitude, good, excellent, *0.8).*
relsem(employee, aptitude, excellent*, bad, 0.1).*

In this work the expression of similarity relations has been considered although some authors indicate that just by using fuzzy sets it is possible to express all the vague elements. We think it turns out to be more expressive in some cases, where the nature of the elements that take part are not easily expressed numerically. It is the case, for example, of the hair color and aptitude, among others.

LINGUISTIC MODIFIERS: HEDGES

These modifiers, defined through functions from [0,1] to [0,1], can be applied in the property functions that define membership of an element to a certain property. They model the effect of the linguistic adverbs like "very ", "rather", "something ", among others.

Definition 5: Let *L = <Lc, Lr, Le, Lm, Lq>* be fuzzy a logical language, *Npred Î Lre È Lrr, Natrib Î S(Npred), Modifier Î Lm, Nrep Î {fuzset, relsem}, fuzzy(Npred, Natrib, Nrep).* We define the metapredicate **hedges** as *hedges(Npred, Natrib, Modifier, Expr),* where *Expr* is the expression of the modification function, whose interpretation is that the modifier defined by the expression *Expr* applies to any defined fuzzy predicate on the attribute *Natrib* of the predicate *Npred*.

Example: *hedges(employee, age, very, mu ^ 2), hedges(employee, age, rather, mu ^ 1/2)*

FUZZY COMPARATORS

Similar to the classic case, where the traditional operators of comparison exist (greater than, less than, equal to), to establish a relation between pairs of attributes or constant, in the fuzzy case there is a mechanism of fuzzy comparison between values, fuzzy comparators represent a fuzzy relation between pairs. Some of these new comparators look like: much greater than, rather smaller than, and approximately equal to, among others. The fuzzy comparators are represented in the model by means of fuzzy set over the difference between the compared elements.

Definition 6: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Comp Î Lrc, A,B,C,D Î Lc*. We define the metapredicate ***fuzset*** as *fuzset(Comp, A, B, C, D),* where the comparator *Comp* is represented by the fuzzy set expressed through the trapezoid [*A,B,C,D*].

Example: We want to define the following flexible comparators represented by the fuzzy set: *much_greater_than, approx_equal_to,* as *fuzset(much_greater_than, 4, 6, 8, inf)* and *fuzset(approx_equal_to, -4, -2, 2, 4).*

FUZZY QUANTIFIERS.

Other critical aspect that must be dealt with, is the definition of fuzzy quantifiers. The idea is to allow through a common frame the representation of both types of quantifiers, Zadeh (1983): absolute (at least three, around 5) and relative (many, at least half, almost all).

Definition 7: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Cuantif Î Lq, A,B,C,D Î Lc*. We define a metapredicate *fuzset* as *fuzset(Cuantif, A, B, C, D).* This indicates that the quantifier *Cuantif* is represented by the trapezoid [*A,B,C,D*]. We define the metapredicate ***quantifier*** as *quantifier(Cuantif, Expr).* That associates to the symbol *Cuantif* the expression of the cardinality given by *Expr*.

Example: For the case of fuzzy absolute quantifier: *quantifier(around_of_5, cardinality_a)* where *cardinality_a* can be defined through the predicate *cardinality_a (Query, Ns, Card)* ¬ *Query Ù Card is sum(Ns)* and *around_of_5* through the fuzzy set represented by *fuzset(around_of_5, 2, 3, 4, inf).*

For the case of fuzzy relative quantifier: *quantifier(majority, cardinality_r)* where *cardinality_r* can is defined through the predicate *cardinality_r(Query1,Ns1,Query2,Ns2,Card)* ¬ *Query1Ù Query2Ù min(Ns1, Ns2, Ns) Ù Card is sum (Ns)* and majority through the fuzzy set represented by *fuzset(majority, n-4, n, n, inf),* where *n* is the total number of tuples considered in the answer.

FUZZY EXTENSION TO QUERIES

FUZZY QUERIES OR FLEXIBLE INTERROGATIONS

A traditional query is just a collection of precise predicates, connected through the extended logical connectors „∨“, „∧“ and „¬“, representing the disjunction, conjunction and negation respectively, where comparators possibly exist. The extension of the traditional queries to fuzzy queries is based on this same philosophy. The fuzzy query contains a collection of connected precise and fuzzy predicates through the extended connectors and where fuzzy comparators possibly exist.

Definition 8: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Q* a formula well founded in L. We say that *Q* is a fuzzy or **flexible query** (or an interrogation) if:

$Q = (pred_1 (t_1, ..., t_{1an}), ... , pred_m (t_m, ..., t_{man})$ Ù $predd_1 (t_i, ns_{d1}), ... , predd_o (t_j, ns_{dn})$ Ù
$predp_1 (t_{p11}, ..., t_{p1n}), ... , predp_r (t_{pr1}, ..., t_{pr1})$ Ù $predc_1 (t_{1i}, t_{1j,}, ns_{c1}), ... , predc_q (t_{qi,} t_{qj,}, ns_{cq}))$.

Where $pred_i$ Î *Lre, 1* £ *i* £ *m,* $predd_j$ Î *Lrd, 0* £ *j* £ *o,* $predp_k$ Î *Lrn, 0* £ *k* £ *r,* $predc_l$ Î *Lrc, 0* £ *l* £ *q*, *t* represents some attribute in a certain predicate, *ns* represents some level of satisfaction that must be satisfied by some fuzzy condition.

Without loss of generality as extended connector was considered only ∧, but the extended connector ∨, can also be used. Another consideration about the connector negation ¬ (that being a logic operator unary) is expressed in the queries as another fuzzy modifier.

The set of tuples that satisfies the conditions imposed in the query is called answer with a satisfaction level, *ns* > 0.

Definition 9: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *Q* an interrogation in *L*, *c = <c1 , ... , cn>* with *ci* Î *Lc,* "*i, 1* £ *i* £ *n, ns* Î *[0,1]*. We say that (*c, ns*) is an **answer** of *Q* if: *Q(c)* is fullfilled or satisfied with degree *ns*.

FUZZY QUANTIFIED QUERIES

There are queries that involve quantifiers in their specification, such as around 5, at least 3, some, few, the majority, among others. The model allows the use of absolute and relative fuzzy quantifiers, it induces tow kinds of fuzzy quantified queries.

Definition 10: Let *L = <Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, *r* Î *Lq* a quantifier symbol, Q_1 and Q_2 fuzzy queries in *L*. We say that $r:Q_1$ is a **quantified fuzzy query**, with *r* an **absolute** quantifier, *r* quantifies the solution set of the query Q_1. We also say that $r:(Q_1 /Q_2)$ is a **quantified fuzzy query**, with *r* a **relative** quantifier, *r* quantifies the proportion between the solution set of Q_1 *and* Q_2

FUZZY EXTENSION TO THE DEDUCTIVE COMPONENT

In this part we show the basic considerations on the rules that define the Intentional Database. In general, when fuzzy rules are mentioned two important elements take part in their definition: the fuzzy predicates and modus ponens generalized that enables the system to have degrees of connection between similar predicates.

The scheme of conditional proposals given by Zadeh (1992), has been interpreted in very different ways by the authors, depending on the application they have been used on. These interpretations consider different types of rules (as conditioning, material implication and triggers that execute a set of actions, among others). Although all of them are valid approaches to model the concrete problems they consider, none of them is suitable in this framework, that is, in solving the problem of obtaining additional information to stored in Database using the defined attributes.

Let $\{A_1, A_2, . . . , A_n\}$ be the set of attributes belonging to the same or to different relations and let's suppose that we wish to obtain some information about a property *B* different to all A_i. In the simplest case, this problem can be modeled with a classical rule as: If *A* then *B* or *A*®*B*, where *A* and *B* are, in general, fuzzy sets on universes *U* and *V*, respectively. In this situation the inference compositional rule, Dubois (1996), offers an ideal frame to derive, from any information related to *A*, noted as *A**, information related to *B*, even allowing to build the fuzzy set *B** associated to *A**, as an answer. However, in the context under study, it is an ideal situation. Let's suppose that we wish to define the concept *good_employee(B)*, on the following features: An employee is good if his aptitude (A_1) is suitable, his yield (A_2) is optimal and his age (A_3) is suitable according to the work. It can be modeled: A_1, A_2 , A_3 ® *B*. If any of the following constructions is applied to the inference compositional rule, they are valid:

A_1^*, A_2 , A_3 ® B_1^* $\qquad$ A_1, A_2, A_3^* ® B_3^*

A_1, A_2^*, A_3 ® B_2^* $\qquad$ A_1^*, A_2^*, A_3 ® B_4^*

A_1*, A_2 , A_3* ® B_5* A_1*, A_2* , A_3* ® B_7*
A_1, A_2* , A_3* ® B_6*

This introduces some complications derived, mainly, from two factors:

- The construction of antecedent is extremely complex and expensive, since all the possible combinations between the fuzzy sets must be considered (thinking that only one set similar to each A exists, which is not necessarily true).
- The consequent construction is also difficult, since when trying to make inference in Database by means of a rule, this one will imply the activation of all the other rules whose consequent are similar to the consulted one.

For this reason, we adopt a model where the inference compositional rule is not used. The problem is reduced to consider the elements that satisfy each condition that appear in the antecedent with certain degree, which implies that the consequent is also verified in certain degree. It is possible to observe that this problem is the same of consulting a Database, with the difference that now the property comes defined by a rule. Therefore, the value of compatibility of the precondition, in relation to the data stored in Database, will be obtained by a matching process, which is used to make the queries. However, the model admits vagueness in the definition of the rule, Baldwin (1987), which includes the virtual relations, constructed by axioms expressed by means of the fuzzy rules or also known as definitory rules, Pons (1997). These rules extend the relations of Database through a range of values of a domain on which attributes will be able to be defined by intension. So, deductions will be able to be carried out. They will allow to obtain additional information from the stored in Database. In addition this eliminates the precise necessity to connect the input and the antecedent, Zadeh (1992).

FUZZY RULES

The formulation of rules that allow in their definition the expression of vague properties (represented by fuzzy sets and/or similarity relations) such as fuzzy comparators, linguistic modifiers, quantifiers fuzzy are grouped under the name of Fuzzy Rules. The expression of a fuzzy rule is equal to the one of a fuzzy query, but considering that now, a property of intensive way can be defined by means of these rules. We admit linear recursion in these rules.

Definition 11: Let *L* = *<Lc, Lr, Le, Lm, Lq>* be a fuzzy logical language, r_i Î *Lr* predicates with *1 £ i £ n*, *C* a rule written in *L*, such that *C* **is of the form** $r_1\ (t_{1a1}, ..., t_{1an})$ ¬ $r_2\ (t_{2a1}, ..., t_{2an})$ Ù . . . Ù $r_n\ (t_{na1}, ..., t_{nan})$**.** We say that *C* is a **fuzzy rule** if $\$\ i$ *, 1 £ i £. n,* r_i Î *Lrd* Ú $\$\ j$ *, 1 £ j £ n,* r_j Î *Lrc*, **where** *Lrd = Lra* È *Lrm*.

CONCLUSIONS

We have formally defined a fuzzy deductive model called FUDEM. This model extends the three basic components of the deductive model: extensional database, intentional database and the query language. This extension allows the use of flexible elements such as fuzzy predicates (representing gradual properties), linguistic modifiers, comparators and quantifiers, fuzzy operators, levels of satisfaction. The proposed model integrates, in an unified frame, fuzzy treatment for precise data in the context of similarity relations and possibility distributions. We have an implementation of a language called ELLENf that is based in FUDEM, Aguilera (1998). The language takes advantage of the capabilities offered by the relational, deductive and fuzzy models, in a transparent and easy way to use. It is implemented as a layer over ORACLE RDBMS and was developed in QUINTUS PROLOG. In practical fields, ELLENf can be used as a development language for intelligent high level applications, which require to express flexible queries and deduction capacities. These applications will have the technology presented by the commercial management system, where these will be implemented.

REFERENCES

Aguilera, Ana, 1998, „Un Lenguaje Flexible para Bases de Datos Deductivas“, Trabajo de Grado, Maestría en Ciencias de la Computación, Universidad Simón Bolívar,Venezuela.

Bald-win, 1987, „FRIL-A Fuzzy Relational Inference Language“, Fuzzy Sets and Systems , Vol. 14, pp. 155-174.

Bosc, Patrick; Pivert, Olivier, 1995, „SQLf : A Relational Database Language for Fuzzy Querying“, IEEE Transactions on Fuzzy Systems, Vol. 3, No. 1, pp 1-17.

Buckles; Petry, 1982, „A Fuzzy Representation of Data for Relational Databases“, Fuzzy Sets and Systems, Vol 7, pp. 213-236.

Derr; Morishita; Phipps, 1993, „Design and Implementation of the Glue-Nail Database System“, ACM-SIGMOD, Vol.

5, pp. 147-156.

Dubois; Prade, 1996, „Fuzzy Sets in Approximate Reasoning. Part I. Inference with Possibility Distributions“, Fuzzy Sets and Systems, Vol. 40, pp. 143-202.

Naqvi; Tsur, 1989, „A Logical Language for Data and Knowledge Bases“, Computer Science Press, New York..

Parsons; Fox, 1994, „A General Approach to Managing Imperfect Information in Deductive Databases“, Proceeding of the Workshop on Uncertainty in Databases and Deductive Systems.

Pons; Medina; Vila; Cubero, 1997, „FREDDI: A Fuzzy Relational Deductive Database Interface“, International Journal of Intelligent Systems, Vol. 17, pp. 597-613.

Prade; Testemale, 1984, „Generalising Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries“, Information Sciences, Vol 34., pp. 115-143.

Ramakrishman, 1992, ‘“CORAL-Control, Relations and Logic“, Proceedings of the 18th International Conference on Very Large Databases“, Canada.

Vaghani, 1992, „An Introduction to the Aditi Deductive Database System“, Technique Report CITRI/TR-92-57, Department of Computer Science, University of Melbourne, Australia.

Zadeh, 1971, „Similarity Relations and Fuzzy Orderings“, Information Sciences, Vol. 8, pp. 177-200.

Zadeh, 1983, „A Computational Approach to Fuzzy Quantifiers in Natural Language“, Computers and Mathematics with Application, Vol. 9, pp. 149-184.

Zadeh, 1992, „The Calculus of Fuzzy If/Then Rules“, AI Expert, pp. 23-24.

Zemankova, Leech; Kandel, 1984, „Fuzzy Relational Databases: A Key for Expert Systems“, Verlag TUV Rheiland.