J. of Mult.-Valued Logic & Soft Computing, Vol. 14, pp. 435–455
Reprints available directly from the publisher
Photocopying permitted by license only

Fuzzy Extended Dependencies to Support Decision-Making in Project Management

Francisco Araque[1], Ramón Carrasco[1], Alberto G. Salguero[1], Amparo Vila[2] and Luis Martínez[3]

[1]*Department of Software Engineering, University of Granada (Andalucía) Spain*
E-mail: faraque,racg,agsh@ugr.es
[2]*Department of Artificial Intelligence, University of Granada (Andalucía) Spain*
E-mail: vila@decsai.ugr.es
[3]*Department of Computer Science, University of Jaén (Andalucía) Spain*
E-mail: martin@ujaen.es

Received: November 28, 2006. Revised: February 19, 2007. Accepted: April 20, 2007.

Project management is becoming an important key process in industrial engineering in order to choose suitable and profitable projects for the companies. We have focused our interest from the manager point of view that needs quality data to make decisions about which kind of projects are more suitable for the company. But nowadays the suitability of the projects don't depend only on quantitative and monetary profits other profits are more and more relevant in the decision for choosing a project such as subjective ones (group satisfaction, cohesiveness of the group, etc.).

The managers have a database with data referred to past projects but usually this database has a huge amount of data that overload the manager in order to study and detect the information he/she needs. Therefore, in this paper we propose a Data Mining process able to deal with quantitative and qualitative features using fuzzy logic (Fuzzy SQL language) to discover the knowledge that the manager needs to make decisions about the more suitable projects for the company. The Data Mining process proposed will obtain Fuzzy Functional Dependencies and Fuzzy Gradual Dependencies by using a flexible query language as the Fuzzy SQL (FSQL), which will provide the information that will support project managers decisions about which type of projects are more suitable for the company based on the projects already done and on objective and subjective features.

Keywords: Fuzzy Functional Dependencies, Gradual Functional Dependencies, Flexible Queries, Fuzzy SQL, Data Mining, Fuzzy Databases, Project Management, Enterprise Strategy, Engineering System, Decision-making.

1 INTRODUCTION

Project management is the discipline of organizing and managing resources in such a way that these resources deliver all the work required to complete a project within defined scope, time, and cost constraints. The project, therefore, is a carefully selected set of activities chosen to use resources (time, money, people, materials, energy, space, provisions, communication, quality, risk, etc.) to meet the pre-defined objectives. Strategic projects are considered to represent the core of corporate growth, change and wealth creation. They are major investments, often involving a high degree of uncertainty, offering intangible benefits (benefits that come from such issues as flexibility, learning, synergies, innovative routines, etc.) and promising attractive long-term financial outcomes [1,7,30].

Therefore our interest is focused on developing tools that support the manager's decisions about the project that should be chosen by the company based on relevant, precise and quality information. It's obvious that managers try to choose those projects that provide greatest profits to the company. However the managers not only evaluate money profits but also other type of profits based on subjective features such as employees satisfaction, employees resign rate, group cohesiveness, etc., that are relevant for the companies in their decisions.

Usually the companies have a database (DB) that contains information about many features that should be analyzed in order to discover the knowledge that supports the manager's decisions. So the use of data mining (DM) processes will help to find out the patterns, features and in general the knowledge that the managers are looking for. We can define DM as the process of extraction of interesting information from the data in databases. According to [34] a discovered knowledge is interesting when it is novel, potentially useful and non-trivial to compute. A series of new functionalities exist in DM, which reaffirms that it is an independent area [34]: high-level language on the discovered knowledge and for showing the results of the user's requests for information (e.g. queries); efficiency on large amounts of data; handling of different types of data; etc.

The DM process involves several steps that take the user along the path from data to information to knowledge. For a successful DM project, the first step is to determine the objective. In our case, we are trying to find useful information about projects performed by an enterprise. To do that, we are going to develop a complete DM process. Next steps include data selection, cleaning, and transformation; model building and pattern discovery; and outcome interpretation and evaluation. Knowledge discovery is a process that can be very iterative and each of these activities may be revisited multiple times. To develop all of the steps of the DM process we use a *Data Warehouse* (DW) approach, where the data selection, cleaning and transformation are performed by the *Extraction, Transformation and Loading* (ETL) module common to any DW architecture and the model building phase corresponds to the multidimensional schema design of the *Data Mart*, i. e. a high focused subset of the DW.

In fact to find out the features, patterns, etc. we have used Functional Dependencies (FD) [23] and Gradual Dependencies (GD) [19] because they reflect immutable properties in a DB hence to discover the knowledge we want to.

Functional Dependencies correspond to correlations among data items, they are expressed in rule form showing attribute-value conditions that commonly occur at the same time in some set of data. In the regular case, a functional dependency, denoted by $X \rightarrow Y$, expresses that a function exists between the two sets of attributes X and Y, and it can be stated as follows: for any pair of tuples t1 and t2, if t1 and t2 have an equal value on X, they also have the same value on Y. Another way of considering the connections between data in databases is to specify a relationship between objects in a dataset and reflect monotonicity in the data by means of that we have called as gradual dependencies (GDs). GD is a concept closely related to the idea of gradual rules introduced by Dubois and Prade [2,3]. In this paper we propose the Extended Dependencies (EDs) as a common framework to integrate FD and GD.

We shall analyze quantitative and qualitative features of the projects, due to this fact, the information involved in the DM process can be uncertain and vague because the subjective features usually are qualitative in nature and it is difficult to provide precise information about this type of information [13]. The modelling and managing of the uncertainty is a key decision in order to obtain good results from the DM process. The probability theory can be a powerful tool. Indeed, traditional risk analysis is conducted primarily using probabilistic tools and techniques. However, it is not difficult to see that many aspects of uncertainties clearly have a non-probabilistic character since they are related to imprecision and vagueness of meanings. Often the type of uncertainties encountered in engineering projects does not fit the axiomatic basis of probability theory, simply because uncertainties in the projects are usually caused due to the inherent incompleteness and fuzziness of features rather than randomness. Therefore the use of the fuzzy logic [35] and linguistic descriptors [36] may be used to describe a subjective feature due to the fact that they are often used by members developing the projects. The linguistic terms are fuzzy judgments and not probabilistic ones. The Fuzzy Linguistic Approach [36] provides a systematic way to represent linguistic variables in a natural decision-making procedure. It does not require providing a precise point at which a subjective feature exists. So it can be used as a powerful tool complementary to traditional methods to deal with imprecise information, especially linguistic information which is commonly used to represent qualitative information [5,22].

Therefore in this paper we propose to develop a DM process based on the fuzzy logic in order to make it more flexible and on the fuzzy linguistic approach to model uncertain information. To do so, we relax the concept of FD and GD by means of Fuzzy FD (FFD) and Fuzzy GD (FGD) that are quite suitable to model non immutable properties existing in the current manifestation of the data.

The concept of FFD given by Cubero and Vila in [19] is a smoothed version of the classical FD. The basic idea consists in replacing the equality used in the FD definition by fuzzy resemblance relations. We can obtain a fuzzy version of GD (FGD) in a similar way. We call Fuzzy Extended Dependencies (FED) to the integration of both FFD and FGD.

The main advantage of FEDs is that they allow us to infer more knowledge from data. Using regular dependencies, only rules that are fulfilled by all of the instances are valid. Using FEDs we can discover dependencies although there are instances that do not fulfil them completely. Furthermore, we can obtain the fulfilment degree for each FED stated.

The DM process proposed will obtain FEDs by using a flexible query language as the Fuzzy SQL (FSQL) [29], which will provide the information that will support project managers decisions about which type of projects are more suitable for the company based on the projects already done and on objective and subjective features. In addition, we have a FSQL Server available to obtain the answers to FSQL queries for Oracle© DBMS [16].

This paper is organized as follows: Section 2 introduces different preliminaries that are necessary to understand the proposal and it is revised in short the FSQL language and the FSQL server. In Section 3 is introduced the definition of Fuzzy Extended Dependencies (FEDs) based on the FSQL operators. In Section 4 FSQL is applied to obtain fuzzy extended dependencies. In Section 5 some experimental results are presented and the paper is concluded in Section 6.

2 PRELIMINARIES

2.1 Related Work

In the last decade many decision-making systems which have to deal with multi-criteria decision problems and qualitative information have shown the capability of Fuzzy Decision Analysis (FDA). Liang and Wang [20] proposed the FDA, which uses fuzzy sets representations and utilizes linguistic variables for rating qualitative factors to aggregate decision-makers' assessments, and applied it on facility site selection and personnel selection. Ghotb and Warren [9] employed FDA to evaluate the necessity of adopting a new hospital information system.

On the other hand, the problem of FD inference has been treated many times in literature. Mannila and Räihä [10] proposed a heuristic algorithm for finding functional dependencies. Kivinen and Mannila [17], and Akutsu and Takasu [32] studied inference of functional dependencies from data with small noise, and gave PAC-type analyses. Investigated for long years, this issue has been recently addressed in a novel and more efficient way by applying principles of data mining algorithms. In this case, the inference of FD is carried out analyzing the data stored in a data base. This method is useful when we have large sets of materialized data (e. g. Data Warehouse environments...). The algorithms fitting in such a trend are TANE, Dep-Miner and others. Recently, due to the high amount of data the organizations store, some researchers has focused in improving the performance of the algorithms [33]. In [26] a strategic planning problem is addressed for a three-stage production–distribution network. Their objective is to minimize the costs associated with production, transportation, and inventory as well as capacity expansion costs over a given time horizon. It proposed three simple linear programming (LP)-based heuristics to obtain good solutions in a reasonable amount of time.

The use of fuzzy theory has been previously used to solve problems related to project management. In [31] a method for measuring functional dependency and sequencing of coupled tasks in engineering design. They provide algorithms for finding the best processing sequence of the coupled tasks in terms of the measured coupling strengths. In [24], it is proposed the use of a new fuzzy linear programming based methodology using a modified S-Curve membership function to solve a fuzzy mix product selection problem. They try to identify the decision for high level of profit with high degree of satisfaction. Quality Function Deployment is a customer-oriented design tool presented in [6] that aims to meet customer needs in a better way and enhance organizational capabilities, while maximizing company goals.

2.2 FSQL: A language for flexible queries

We have developed a language (FSQL) to manage uncertainties and imprecise information [16]. We have extended the SQL language to allow flexible queries. Thus, the language can manage fuzzy attributes, from different nature that is necessary in our problem, which are classified by the system in 3 types:

- Type 1: These attributes are totally crisp, but they have some linguistic trapezoidal labels defined on them.
- Type 2: These attributes admit crisp data as well as possibility distributions over an ordered underlying domain.
- Type 3: On these attributes, some labels are defined and on these labels, a similarity relation has yet to be defined. These attributes have no relation of order.

The Fuzzy Meta-knowledge Base (FMB) stores information for the fuzzy treatment of the fuzzy attributes [8]. It is a set of relational tables which allows the definition of:

- Representation Functions: these functions are used to show the fuzzy attributes in a comprehensible way for the user and not in the internally used format.

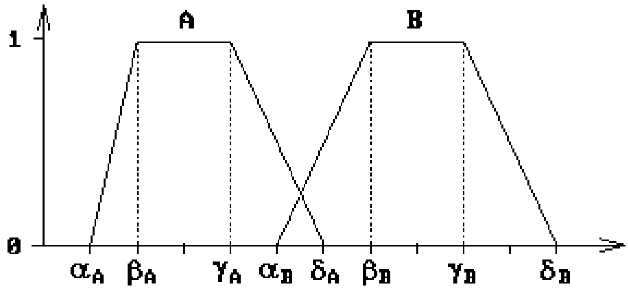

FIGURE 1
Trapezoidal possibility distributions: A, B.

– Fuzzy Comparison Functions: they are utilized to compare the fuzzy values and to calculate the compatibility degrees (CDEG function)

We have extended the SELECT command to express flexible queries and, due to its complex format, we only show an abstract with the main extensions added to this command:

– Fuzzy Comparators: In addition to the common comparators ($=$, $>$, etc), FSQL includes fuzzy comparators of two trapezoidal possibility distributions A, B (see Figure 1) with A $=$ \$[αA, βA, γA, δA] B $=$ \$[αB, βB, γB, δB] in Table 1. In the same way as in SQL, fuzzy comparators can compare one column with one constant or two columns of the same type. Necessity comparators are more restrictive than possibility comparators, i.e. their fulfillment degree is always lower than the fulfillment degree of their corresponding possibility comparator. More information can be found in [14,15].

– Fulfillment Thresholds γ. For each simple condition a Fulfillment threshold may be established with the format $<$ condition $>$ THOLD$_\gamma$, indicating that the condition must be satisfied with a minimum degree γ in [0, 1] fulfilled.

– CDEG($<$ attribute $>$) function: This function shows a column with the fulfillment degree of the condition of the query for a specific attribute, which is expressed in brackets as the argument.

– Fuzzy Constants: In FSQL we can use all of the fuzzy constants which appear in Table 2.

– Fuzzy Quantifiers: They can either be relative or absolute with the formats *\$Quantifier* [*FUZZY*] (*<condition>*) *THOLD* χ *or \$Quantifier* [*FUZZY*] (*<condition_1>*) *ARE* (*<condition_2>*) *THOLD* χ,

F_Comp	Significance	CDEG(A F_Comp B) Possibility operator	CDEG(A F_Comp B) Necessity operator
FEQ NFEQ	Possibly/Necessarily Fuzzy Equal	where U is the domain of A, B. A(d) is the degree of the possibility for d U in the distribution A	where U is the domain of A, B. A(d) is the degree of the possibility for d U in the distribution A
FGT NFGT	Possibly/Necessarily Fuzzy Greater Than	$= 1$ if $\gamma_A \geq \delta_B$ $= \frac{\delta_A - \gamma_B}{(\delta_B - \gamma_B) - (\gamma_A - \delta_A)}$ if $\gamma_A < \delta_B \& \delta_A > \gamma_B$ $= 0$ otherwise	$= 1$ if $\alpha_A \geq \delta_B$ $= \frac{\delta_A - \gamma_B}{(\delta_B - \gamma_B) - (\gamma_A - \delta_A)}$ if $\alpha_A < \delta_B \& \beta_A > \gamma_B$ $= 0$ otherwise
FGEQ NFGEQ	Possibly/Necessarily Fuzzy Greater or EQual	$= 1$ if $\gamma_A \geq \beta_B$ $= \frac{\delta_A - \alpha_B}{(\beta_B - \alpha_B) - (\gamma_A - \delta_A)}$ if $\gamma_A < \beta_B \& \delta_A > \alpha_B$ $= 0$ otherwise	$= 1$ if $\alpha_A \geq \beta_B$ $= \frac{\beta_A - \alpha_B}{(\beta_B - \alpha_B) - (\alpha_A - \beta_A)}$ if $\alpha_A < \beta_B \& \beta_A > \alpha_B$ $= 0$ otherwise
FLT NFLT	Possibly/Necessarily Fuzzy Less Than	$= 1$ if $\beta_A \leq \alpha_B$ $= \frac{\alpha_A - \beta_B}{(\alpha_B - \beta_B) - (\beta_A - \alpha_A)}$ if $\beta_A > \alpha_B \& \alpha_A > \beta_B$ $= 0$ otherwise	$= 1$ if $\delta_A \leq \alpha_B$ $= \frac{\gamma_A - \beta_B}{(\alpha_B - \beta_B) - (\delta_A - \gamma_A)}$ if $\delta_A < \alpha_B \& \gamma_A < \beta_B$ $= 0$ otherwise
FLEQ NFLEQ	Possibly/Necessarily Fuzzy Less or EQual	$= 1$ if $\beta_A \leq \gamma_B$ $= \frac{\delta_B - \alpha_A}{(\beta_A - \alpha_A) - (\gamma_B - \delta_B)}$ if $\beta_A > \gamma_B \& \alpha_A < \delta_B$ $= 0$ otherwise	$= 1$ if $\alpha_A \leq \gamma_B$ $= \frac{\gamma_A - \delta_B}{(\gamma_B - \delta_B) - (\delta_A - \gamma_A)}$ if $\delta_A < \gamma_B \& \gamma_A < \delta_B$ $= 0$ otherwise

TABLE 1
Fuzzy Comparators for FSQL. Operator NOT can precede to F_Comp

F_Comp	Significance	CDEG(A F_Comp B) Possibility operator	CDEG(A F_Comp B) Necessity operator
MGT NMGT	Possibly/Necessarily Fuzzy Much Greater Than	$= 1$ if $\gamma_A \geq \delta_B + M$ $= \frac{\gamma_B + M - \delta_A}{(\beta_A - \alpha_A) - (\gamma_B - \delta_B)}$ if $\gamma_A < \delta_B + M \& \delta_A > \gamma_B + M$ $= 0$ otherwise M is the minimum distance to consider two attributes as very separate. M is defined in FMB for each attribute	$= 1$ if $\alpha_A \geq \delta_B + M$ $= \frac{\gamma_B + M - \beta_A}{(\alpha_A - \beta_A) - (\delta_B - \gamma_B)}$ if $\alpha_A < \delta_B + M \& \beta_A > \gamma_B + M$ $= 0$ otherwise M is the minimum distance to consider two attributes as very separate
MLT NMLT	Possibly/Necessarily Fuzzy Much Less Than	$= 1$ if $\beta_A \leq \alpha_B - M$ $= \frac{\beta_B - M - \alpha_A}{(\beta_A - \alpha_A) - (\alpha_B - \beta_B)}$ if $\beta_A > \alpha_B - M \& \alpha_A < \beta_B - M$ $= 0$ otherwise M is the minimum distance to consider two attributes as very separate	$= 1$ if $\delta_A \leq \alpha_B - M$ $= \frac{\beta_B - M - \gamma_A}{(\delta_A - \gamma_A) - (\alpha_B - \beta_B)}$ if $\delta_A > \alpha_B - M \& \gamma_A < \beta_B - M$ $= 0$ otherwise M is the minimum distance to consider two attributes as very separate

TABLE 1
(Continued)

F. Constant	Significance
UNKOWN **UNDEFINED NULL**	Unknown value but the attribute is applicable The attribute is not applicable or it is meaningless Total ignorance: We know nothing about it
$\mathbf{A} = \$[\alpha_\mathbf{A}, \beta_\mathbf{A}, \gamma_\mathbf{A}, \delta_\mathbf{A}]$ **$label [n, m] #n**	Fuzzy trapezoid ($\alpha_A \leq \beta_A \leq \gamma_A \leq \delta_A$): See Figure 1 Linguistic Label: It may be a trapezoid or a scalar (defined in FMB) Interval "Between n and m" ($\alpha_\mathrm{A} = \beta_\mathrm{A} =$ n and $\gamma_\mathrm{A} = \delta_\mathrm{A} =$ m) Fuzzy value "Approximately n" ($\beta_\mathrm{A} = \gamma_\mathrm{A} =$ n and n-$\alpha_\mathrm{A} = \delta_\mathrm{A} =$ margin)

TABLE 2
Fuzzy constants of FSQL

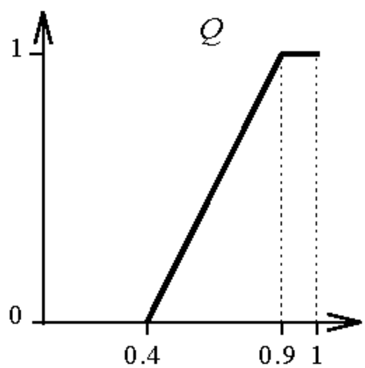

FIGURE 2
Fuzzy quantifier most.

indicating that the quantifier must be satisfied with a minimum degree χ in [0,1] fulfilled. In the Figure 2 we can see an example of the quantifier *most* defined in the FMB.

We have a FSQL Server available to obtain the answers to FSQL queries for Oracle© DBMS.

3 FUZZY FUNCTIONAL DEPENDENCIES AND GRADUAL FUNCTIONAL DEPENDENCIES

There have been several approaches to the problem of defining the concept of FFD but unlike classical FDs one single approach has not dominated. We begin by briefly describing the concept of classical FD, later we give a general definition of FFD and GFD based on fuzzy functions and then, we shall introduce a more relaxed definition of FFD and GFD in order to manage exceptions.

The relation *R* with attribute sets $X = (x_1, \ldots, x_n)$, and $Y = (y_1, \ldots, y_m)$ in its scheme verifies the **FD X → Y** if and only if, for every instance *r* of *R* it is verified:

$$\forall t_1, t_2 r, t_1[X] = t_2[X] t_1[Y] = t_2[Y] \tag{1}$$

The concept of FFD given by Cubero and Vila in [19] is a smoothed version of the classical FD. The basic idea consists in replacing the equality used in the FD definition by fuzzy resemblance relations, in such a way that: The relation *R* verifies an $\alpha - \beta$ FFD $X \rightarrow_{FT} Y$ if and only if, for every instance *r* of *R* it is verified:

$$\forall t_1, t_2 r, F(t_1[X], t_2[X]) \geq \alpha \Rightarrow T(t_1[Y], t_2[Y]) \geq \beta \tag{2}$$

where F and T are fuzzy resemblance relations.

The flexibility provided by the use of the parameters α and β and the different kinds of resemblance relations should be noted. If F is a weak resemblance measure and T is a strong one, we get interesting properties for database design (decomposition of relations). A more detailed description of these concepts can be found in [18,19].

Often just a few tuples in a database can prevent the FFD from being completed. To avoid this, we can relax the FFD definition in such a way that all the tuples of the relationship are not forced to fulfill the above condition, therefore we define:

Definition 1. (confidence of a FFD). The relation *R* verifies an $\alpha - \beta$ FFD $X \rightarrow_{FT} Y$ with confidence *c*, where *c* is defined as:

$$\begin{aligned} &c = 0 \text{ if } Card\{(t_1, t_2)t_1, t_2 \in r/F(t_1[X], t_2[X]) \geq \alpha\} = 0 \\ &c = \frac{Card\{(t_1, t_2)t_1, t_2 \in r/F(t_1[X], t_2[X]) \geq \alpha \wedge T(t_1[Y], t_2[Y]) \geq \beta\}}{Card\{(t_1, t_2)t_1, t_2 \in r/F(t_1[X], t_2[X]) \geq \alpha\}} \\ &\quad \times \text{ Otherwise} \end{aligned} \tag{3}$$

Where $\wedge$ is the logical operator *and*. The basic idea consists of computing the percentage of tuples which fulfill the antecedent and consequent together with respect to those which only fulfill the consequent.

Definition 2. The relation *R* verifies an $\alpha - \beta$ FFD $X \rightarrow_{FT} Y$ with support *s*, where $s \in [0, 1]$, is defined as:

$$\begin{aligned} &S = 0 \text{if } n = 0 \\ &S = \frac{Card\{(t_1, t_2)t_1, t_2 \in r/F(t_1[X], t_2[X]) \geq \alpha \wedge T(t_1[Y], t_2[Y]) \geq \beta\}}{n} \\ &\quad \times \text{ otherwise} \end{aligned} \tag{4}$$

where *n* is the number of tuples of the *r* instance of the relation *R*.

The idea is to find the percentage of tuples which fulfill the antecedent and consequent together with respect to the total rows of the relation.

Another way of considering the connections between data in databases is to specify a relationship between objects in a dataset and reflect monotonicity in the data by means of gradual fuzzy dependencies (GFDs). It is closely related to the idea of gradual rules introduced by Dubois and Prade [3,2]. An intuitive example of a GFD is "the bigger business is the higher earnings they have" and we assume that the concept of GFD can be considered, in this way, as similar to the FFD one. Therefore we define:

Definition 3. ($\alpha - \beta$ gradual functional dependency) The relation R verifies an $\alpha - \beta$ GFD X $\int_{\mathrm{FT}}$ Y if and only if, for every instance r of R it is verified:

$$\forall t_1, t_2 r,\ F'(t_1[X], t_2[X]) \geq \alpha \Rightarrow T'(t_1[Y], t_2[Y]) \geq \beta \tag{5}$$

where F' and T' are fuzzy relations of the type: *fuzzy greater than, fuzzy greater than or equal to, fuzzy less than, fuzzy less than or equal to, fuzzy not equal*, etc. We can define an $\alpha - -\beta$ GFD X $\int_{\mathrm{F'T'}}$ Y with confidence c in the same way that we have made it for FFD (see Definition 1).

4 APPLYING FSQL TO OBTAIN FUZZY EXTENDED DEPENDENCIES

Now, it is necessary to relate the FSQL environment to our definitions. To do so, we first introduce a general definition of Fuzzy Extended Dependencies based on FSQL operators and FSQL CDEG function, later we will show how FED can be calculated with FSQL.

4.1 Fuzzy Extended Dependencies with FSQL operators

Definition 4. The relation R with attribute sets $\mathrm{X} = (\mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n})$, and $Y = (y_1 \ldots y_m)$ whose attributes are trapezoidal possibility distributions, verifies an $\alpha - \beta$ FED X $\blacktriangleright_{\mathrm{F*T*}}$ Y with $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_\mathrm{n})/\alpha_\mathrm{i} \in [0, 1]\ \forall \mathrm{i} = 1, \ldots, \mathrm{n}$ and $\beta = (\beta_1, \beta_2, \ldots, \beta_\mathrm{m})/\beta_\mathrm{j} \in [0, 1] \forall \mathrm{j} = 1, \ldots, \mathrm{m}$, if and only if, for every instance r of R it is verified:

$$\begin{aligned}
&\forall \mathrm{t}_1, \mathrm{t}_2 \in \mathrm{r}, \wedge_{\mathrm{i}=1,2,\ldots,\mathrm{n}}[\mathrm{F}^*_\mathrm{i}(\mathrm{t}_1[\mathrm{x}_\mathrm{i}], \mathrm{t}_2[\mathrm{x}_\mathrm{i}]) \geq \alpha_\mathrm{i}]\\
&\quad \Rightarrow \wedge_{j=1,2,\ldots,m}[T^*_j(t_1[y_j], t_2[y_j]) \geq \beta_j] \text{ where}\\
&\qquad \mathrm{F}^*_\mathrm{i} : \mathrm{UxU} \rightarrow [0, 1]/\mathrm{F}^*_\mathrm{i}(\mathrm{A}, \mathrm{B}) = \mathrm{CDEG}(\mathrm{A}F_Comp_ant_i\mathrm{B})\\
&\qquad \mathrm{T}^*_\mathrm{j} : \mathrm{UxU} \rightarrow [0, 1]/\mathrm{T}^*_\mathrm{j}(\mathrm{A}, \mathrm{B}) = \mathrm{CDEG}(\mathrm{A}F_Comp_con_j\mathrm{B})\\
&\quad \forall \mathrm{A} = \$[\alpha_\mathrm{A}, \beta_\mathrm{A}, \gamma_\mathrm{A}, \delta_\mathrm{A}], \mathrm{B} = \$[\alpha_\mathrm{B}, \beta_\mathrm{B}, \gamma_\mathrm{B}, \delta_\mathrm{B}]\mathrm{U} (\text{see Figure 1})
\end{aligned} \tag{6}$$

$F\_Comp\_ant_i$, $F\_Comp\_con_j$ defined as any fuzzy comparator in FSQL (any *F_Comp* in Table 1, even when preceded by a NOT operator) $\forall \mathrm{i} = 1, \ldots, \mathrm{n}, \forall \mathrm{j} = 1, \ldots, \mathrm{m}$

Definition 5. The relation R with attribute sets $\mathrm{X} = (\mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n})$, and $\mathrm{Y} = (\mathrm{y}_1 \ldots \mathrm{y}_\mathrm{m})$ whose attributes are trapezoidal possibility distributions, verifies an $\alpha - \beta$ FED $\mathrm{X} \blacktriangleright_{\mathrm{F*T*}} \mathrm{Y}$ with $\alpha \in [0, 1]$ and $\beta \in [0, 1]$, if and only if, for every instance r of R it is verified:

$$\begin{aligned} &\forall t_1, t_2 \in r, \wedge_{i=1,2,\ldots,n}[F_i^*(t_1[x_i], t_2[x_i]) \geq \alpha] \\ &\Rightarrow \wedge_{j=1,2,\ldots,m}[T_j^*(t_1[y_j], t_2[y_j]) \geq \beta] \\ &\forall i = 1, \ldots, n,\ j = 1, \ldots, m \end{aligned} \tag{7}$$

Now, we can make a new definition of FFDs and GFDs as a particular case of FEDs.

Definition 6. If *F_Comp_ant*$_i$, *F_Comp_con*$_j$ $\in$ {FEQ, NFEQ} then we say that R verifies an $\alpha_i - \beta_i$ FFD $\mathrm{X} \rightarrow_{\mathrm{F*T*}} \mathrm{Y}$.

Definition 7. If *F_Comp_ant*$_i$, *F_Comp_con*$_j$ $\in$ {FEQ, NFEQ} then we say that R verifies an $\alpha - \beta$ FFD $\mathrm{X} \rightarrow_{\mathrm{F*T*}} \mathrm{Y}$.

Definition 8. If *F_Comp_ant, F_Comp_con* are any *F_Comp* of FSQL such that there exists at least a k from 1 to n which fulfils *F_Comp_ant*$_k$ $\notin$ {FEQ, NFEQ} and at least a s from 1 to m which fulfils *F_Comp_con*$_s$ $\notin$ {FEQ, NFEQ} then we say that R verifies an $\alpha - \beta$ GFD $\mathrm{X} \int_{\mathrm{F*T*}} \mathrm{Y}$.

Definition 9. If *F_Comp_ant*$_i$, *F_Comp_con*$_j$ are any *F_Comp* of FSQL such that there exists at least a k from 1 to n which fulfils *F_Comp_ant*$_k$ $\in$ {FEQ, NFEQ} and at least a s from 1 to m which fulfils *F_Comp_con*$_s$ $\notin$ {FEQ, NFEQ} then we say that R verifies an $\alpha_i - \beta_i$ GFD $\mathrm{X} \int_{\mathrm{F*T*}} \mathrm{Y}$.

Of course we can define an $\alpha - \beta$ FED $\mathrm{X} \blacktriangleright_{\mathrm{F*T*}} \mathrm{Y}$ with confidence c in the same sense that we have made it for FFD (see Definition 1). To simplify notation, in $\mathrm{X} \blacktriangleright_{\mathrm{F*T*}} \mathrm{Y}$ we will denote F* as (*F_Comp_ant*$_i$)* $\forall i = 1, \ldots, n$, and similar notation for T*.

4.2 Obtaining Fuzzy Extended Dependencies from a Database by using FSQL

Let R be a relation with attribute sets $\mathrm{X} = (\mathrm{x}_1 \ldots \mathrm{x}_\mathrm{n})$, $\mathrm{Y} = (\mathrm{y}_1 \ldots \mathrm{y}_\mathrm{m})$ and $\mathrm{PK} = (\mathrm{pk}_1 \ldots \mathrm{pk}_\mathrm{S})$ included in its scheme, where PK is the primary key of R. To determine if R verifies an $\alpha - \beta$ FED $\mathrm{X} \blacktriangleright_{\mathrm{F*T*}} \mathrm{Y}$ for an instance r, we create a FSQL query with the following general format:

```
SELECT count(*) FROM  r A1, r A2
WHERE (A1.PK <> A2.PK)
 AND          (A1.x1 F_Comp_ant1 A2.x1 THOLD α1
     AND...AND A1.xn F_Comp_antn A2.xn THOLD αn)
AND NOT       (A1.y1 F_Comp_con1 A2.y1 THOLD β1
     AND...AND A1.ym F_Comp_conm A2.ym THOLD βm)
```

The basic idea consists of computing the tuples which fulfill the antecedent and do not fulfill the consequent. Therefore, if the if the result of the query is 0, we can say that R verifies FED for the instance r.

If the previous result is not 0, we can determine if R verifies an $\alpha - \beta$ FED X $\blacktriangleright_{F*T*}$ Y with confidence c by means of a simple procedure as follows (algorithm 1):

Step 1.1: To obtain the value a as the number of tuples which fulfil the antecedent and consequent together:

```
SELECT count(*) FROM   r A1, r A2
WHERE  (A1.PK <> A2.PK)
AND          (A1.x1 F_Comp_ant1 A2.x1 THOLD α1
  AND ... AND A1.xn F_Comp_antn A2.xn THOLD αn)
AND          (A1.y1 F_Comp_con1 A2.y1 THOLD β1
  AND ... AND A1.ym F_Comp_conm A2.ym THOLD βm)
```

Step 1.2: To obtain the value b as the number of tuples which fulfil the antecedent:

```
SELECT count(*) FROM r A1, r A2
WHERE (A1.PK <> A2.PK)
 AND          (A1.x1 F_Comp_ant1 A2.x1 THOLD α1
  AND ... AND A1.xn F_Comp_antn A2.xn THOLD αn)
```

Step 2: To obtain the degree of confidence c as $c = a/b$ and support s as $s = a/n$, where n is the number of rows in r.

Step 3: To determine if the computed degree indicates that the FED is good enough, we can compare the value c with some fuzzy quantifier defined in the FMB (by example *most*).

Notice that FSQL also allows us to compare (with fuzzy comparators) crisp attributes. In order to do this, FSQL makes a fuzzyfication of the crisp value before the comparison, transforming it into a triangular possibility distribution (according to values stored in the FMB for the attribute). This fuzzyfication can either be implicit or explicit (with the fuzzy constant #). Also, FSQL can work with scalar attributes but with them we can use only use the comparator FEQ (because an order relationship in their domains is not defined).

If the purpose is to search for FFDs in order to discover intentional properties (constraints that exist in every possible manifestation of the database frame) it seems more appropriate to use a weak resemblance measure in the antecedent (FEQ, based on possibility) as a fuzzy comparator and a strong one in the consequent (NFEQ, based on necessity). In this way, we get interesting properties which can help us with the decomposition of relations [18]. Searching for FFDs or GFDs to discover extensional properties (those existing in the current manifestation of the data) is a task for DM. In this case, the choice of the fuzzy comparators and the parameters α, β we will be made according to the specific problem in question.

5 SUPPORTING DECISION MAKING IN PROJECT MANAGEMENT

In this section we show how the above process is performed. We are going to apply it to support the decision making process on project management.

Projects are the vehicles by which organizations achieve their strategic goals. Effective delivery of strategic plans requires organizations to make better choices about their priorities and how they choose to invest their scarcest resources: time, people and money. Let suppose a project manager who wants to measure the truthfulness of the following statement:

"Bigger projects, with equal or more fitness of the team to those projects and with an equal dedication to those projects implies higher benefits and equal degree of satisfaction for both employees and clients"

The FSQL language is the natural way to obtain such FEDs. We have a FSQL server available for Oracle© Databases, programmed in PL/SQL. This server allows us to query a Fuzzy or Classical Database with the FSQL language (Fuzzy SQL).

Let PROJETS be a relation with the statistics of the projects of some Spanish companies during '2006 with the data shown in Table 3. This table has been obtained from a Data Warehouse (DW) system of a project management company. It corresponds to the facts table of the multidimensional scheme of a DW so the data to analyze are collected directly from a table. Due to the fact that the FSQL is an extended version of the SQL language, it is possible to retrieve the data from many tables by means of multiple joins. In a simple

Project_id	projects_hours	fitting	dedication	Profit	user_satisfation	team_satisfation
1	280	very low	full	10	poor	good
2	532	very low	full	12	regular	good
3	654	low	high	15	regular	excellent
4	828	average	high	8	good	excellent
5	991	average	high	23	good	excellent
6	1555	average	high	26	good	excellent
7	2327	average	high	28	good	poor
8	3498	average	medium	30	good	excellent
9	7896	high	medium	33	good	good
10	8124	high	medium	35	good	good
11	8798	high	low	47	good	good
12	10054	high	low	49	excellent	excellent
13	11000	high	low	50	poor	good
14	12655	high	low	55	excellent	good
15	14567	very high	low	62	excellent	good

TABLE 3
Table PROJECTS

way, we have selected the most representative fields of this table useful to show the process previously detailed. To manage these attributes:

- project_hours: is the duration of the project in hours. This is a crisp attribute but we decide define this as Type 1 in the FMB using the fuzzy constants value #n = 50 (which means approximately n) and margin = 2000 (see Table 2)
- fitting: is the human resources fitting to the project. This is a fuzzy attribute of Type 2 defined in the FMB with the fuzzy labels showed in Figure 3.
- dedication: is the dedication average percentage of human resources to the project. This is a crisp attribute but we decide define this as Type 1 in the FMB. Besides, we decide to define the fuzzy labels showed in the Figure 4 for this attribute to simplify their use.
- profit: is the profit percentage of the project, i.e., the project cost divide by the project profit. This is a crisp attribute but we decide define this

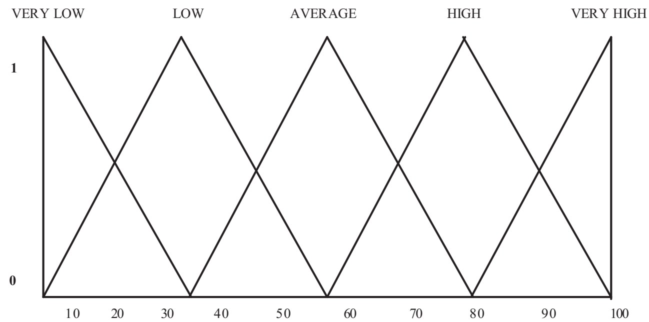

FIGURE 3
Fuzzy distribution for Fitting parameter.

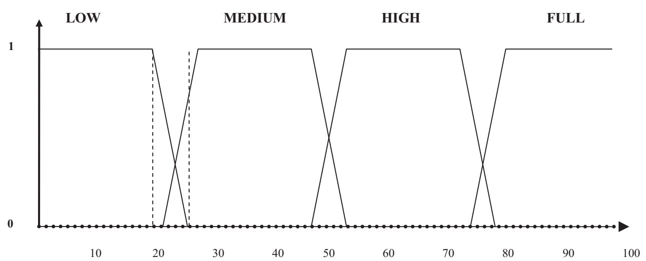

FIGURE 4
Fuzzy distribution for Dedication parameter.

Team and user satisfaction	Poor	Regular	Good	Excellent
Poor	1	0.6	0.2	0
Regular		1	0.6	0.2
Good			1	0.6
Excellent				1

TABLE 4
Similarity relationship defined for user_satisfation and team_satisfation. This attributes are included in a data base table with all clients called PROJECTS

as Type 1 in the FMB using the fuzzy constants value #n = 5 and margin = 60 (approximately n and margin see Table 2)

- user_satisfaction: is the user satisfaction value after the end of the project. This is a scalar attribute (Type 3), therefore we define a similarity relationship for the FEQ comparator in the FMB (see Table 4)
- team_satisfaction: is the human resources satisfaction value after the end of the project. This is a scalar attribute (Type 3), therefore we define a similarity relationship for the FEQ comparator in the FMB (see Table 4)

The selection of the type of each attribute is not a direct task. It is an iterative process where the user refines the knowledge representation of the specific problem in the FMB. Usually, the range of the attribute determines its type, but other parameters like fuzzy distribution of labels or margin similarities of crisp values should be tuned. The more familiar with the problem the user is the lower is the effort needed.

As it has been explained in Section 4.2 and following the scheme showed in Figure 5, after some trials we show the results obtained (Step 1.1 and 1.2 of the previously detailed procedure in 4.2) in Figures 6 and 7. Result of FSQL query Step 1.1 Alg.1 is: 12. Projects 4, 7 and 13 do not fulfil the consequent. Project 4 is supposed to be more profitable than other projects, like project 3 for instance, due to the fact that the *fitting* and *user_satisfaction* parameters contain higher values. In this case, project 4 does not fulfil the functional dependency because we find more profitable projects (project 3, for instance) which have lower values for *fitting* and *user_satisfaction* parameters. Same occur with project 7 which has more profit than project 6, for example, although it has a lower value for the *team_satisfaction* parameter. The same can be applied to projects 12 and 13.

As it has been explained before, it is not a direct task. Usually, the user has to try and refine the threshold parameter values, stored in the FMB, Therefore

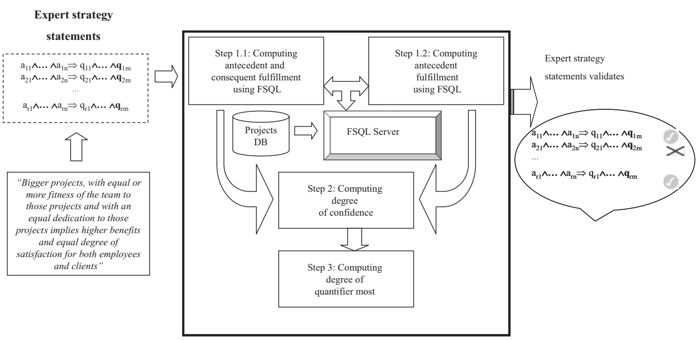

FIGURE 5
Fuzzy Data Mining Decision process.

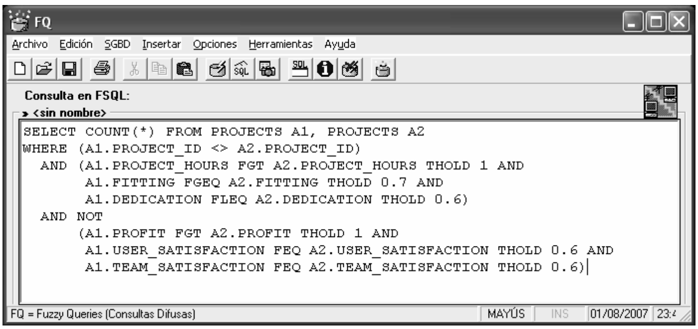

FIGURE 6
Result of FSQL query Step 1.1 Alg.1 is: 12. Projects 4, 7 and 13 do not fulfil the consequent.

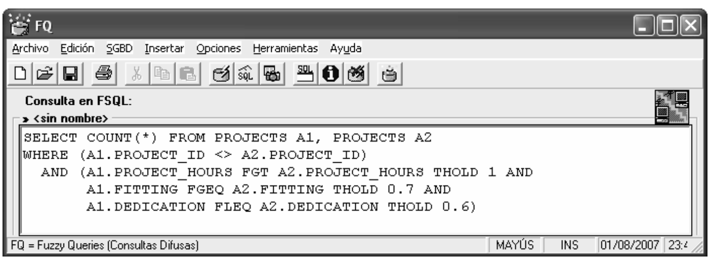

FIGURE 7
Result of FSQL query Step 1.2 Alg.1 is: 15. All the projects fulfil the antecedent.

(Step 2 of procedure in 4.2) we can say that PROJECTS verifies:

$$(1, 0.7, 0.6)\text{–}(1, 0.6, 0.6)GD(\textit{project\_hours}, \textit{fitting}, \textit{dedication})$$
$$\rightarrow_{(\mathrm{FGT},\mathrm{FGEQ},\mathrm{FLEQ})*(\mathrm{FGT},\mathrm{FEQ},\mathrm{FEQ})}$$
$$*\ (\textit{profit}, \textit{user\_satisfaction}, \textit{team\_satisfaction})$$

with confidence c = 0.8. Now (Step 3 of procedure in 4.2) if we compare this value with the fuzzy quantifier *most* (See Figure 2) we can say that the above GD is verified with fulfillment thresholds 0.78 for most of the tuples. We can conclude that the strategy outlined by the project manager, i. e. the strategic plan should be oriented to face bigger projects, is valid given the data in the DW for historical projects.

At this point, we are not interested in knowing which projects fulfill the statement. We need to know if it is a valid hypothesis or not for most of them. Otherwise, instead of using the COUNT aggregating function in the query, it is possible to retrieve the list of projects ranked by their fulfillment degree using the CDEG function explained in Section 2.2.

Figures 6 and 7 show the interface of a fully functional application capable of sending an FSQL query to the server. As explained in point 4.2, the FSQL queries are done in an entirely straight manner. The user only has to select the fields of the table to include in the analysis and the corresponding threshold parameter and then put them in a query according to steps 1.1 and 1.2 of the algorithm explained in section 4.2. In addition, due to the straight manner the queries are built, the development of wizards or friendly applications that help users not familiarized with the SQL language in the building of the query.

We are developing a prototype that, once the user has selected a table in a data base, all the fields of the table can be selected to be part of the precedent, the consequent of neither of them. Then, the user only has to select the fuzzy comparator for each of them. The threshold parameters, i.e. α and β values in the FSQL queries, are calculated automatically by the application given a degree of confidence and/or support value.

6 CONCLUSIONS

In this paper we have proposed a Data Mining process applying to project management in order to support managers decisions about which projects are more suitable for their companies, taking into account objective and subjective features.

This DM process is based on the use of fuzzy extended dependencies (FEDs) as a common framework to integrate fuzzy functional dependencies and gradual functional dependencies. Also, we have relaxed the FED definition for finding FEDs even if exceptional tuples do not verify it. FEDs are defined with the FSQL fuzzy comparators on trapezoidal possibility distribution. Therefore, the FSQL language is the natural way to obtain such FEDs. Using possibility in FEDs as a weak resemblance in the antecedent and necessity as a strong one in the consequent, FSQL could be used to find FFDs which portray constraints that exist in every possible manifestation of the frames in a database (useful for the decomposition of relations). A practical application is to search for FFDs or GFDs in order to discover properties which exist in the current manifestation of the data as a task for DM. The process detailed in this paper helps in the process of finding the fulfillment degree of any statement the user wants to verify given a set of historical data. We have applied it to find more profitable projects for a specific enterprise. Once the more profitable projects have been identified the manager can develop a convenient strategic plan.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Research Program under projects TIN2006-02121, TIN2005-09098-C05-03 and by the Andalucía Research Program under project GR050/06.

REFERENCES

[1] Boris Asrilhant, Robert G. Dyson and Maureen Meadows. On the strategic project management process in the UK upstream oil and gas sector. Omega, Volume 35, Issue 1, February 2007, Pages 89–103.

[2] D. Dubois, H. Prade, "Generalized dependencies in fuzzy databases". In International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU'92, Palma de Mallorca, Spain, pp. 263–266, 1992.

[3] D. Dubois, H. Prade, "Gradual rules in approximate reasoning". Information Sciences, Vol 61, pp. 103–122.

[4] D. Rasmussen, R.R. Yager, "Finding fuzzy and gradual functional dependencies with SummarySQL" Fuzzy Sets and Systems, Vol 106–2, pp. 131–142, 1999.

[5] Delgado, M., J.L. Verdegay, and M.A. Vila, Linguistic Decision Making Models. International Journal of Intelligent Systems, 1992. 7: p. 479–492.

[6] E. Ertugrul Karsak. Fuzzy multiple objective programming framework to prioritize design requirements in quality function deployment. Computers & Industrial Engineering, Volume 47, Issues 2–3, November 2004, Pages 149–163.

[7] E.W. Hans, W. Herroelen, R. Leus and G. Wullink. A hierarchical approach to multi-project planning under uncertainty. Omega, Volume 35, Issue 5, October 2007, Pages 563–577.

[8] Galindo, J., Urrutia, A., Piattini, M. Representation of fuzzy knowledge in relational databases. Proceedings. 15th International Workshop on Database and Expert Systems Applications. 2004 p. 917–921.

[9] Ghotb, F., & Warren, L. (1995). A case study comparison of the analytic hierarchy process and a fuzzy decision methodology. Engineering Economist, 40(3), 233–246.

[10] H. Mannila, K. Räihä, Dependency inference, in: Proc. 13th VLDB Conf., 1987, pp. 155–158.

[11] Hee Beng Kuan Tan and Yuan Zhao. Automated elicitation of functional dependencies from source codes of database transactions. Information and Software Technology, Volume 46, Issue 2, 1 February 2004, Pages 109–117.

[12] Herrera, F. and E. Herrera-Viedma, Linguistic Decision Analysis: Steps for Solving Decision Problems under Linguistic Information. Fuzzy Sets and Systems, 2000. 115: p. 67–82.

[13] Herrera, F., L. Martínez, and P.J. Sánchez, Managing Non-Homogeneous Information in Group Decision Making. European Journal of Operational Research, 2005. 166(1): p. 115–132.

[14] J. Galindo, "Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales". Ph. Doctoral Thesis, University of Granada (Spain), March 1999.

[15] J. Galindo, J.M. Medina, A. Vila, O. Pons, “Fuzzy Comparators for Flexible Queries to Databases”. Iberoamerican Conference on Artificial Intelligence, IBERAMIA’98, pp. 29–41, Lisbon (Portugal), October 1998.

[16] J. Galindo, J.M. Medina, O. Pons, J.C. Cubero, “A Server for Fuzzy SQL Queries”, in “Flexible Query Answering Systems”, eds. T. Andreasen, H. Christiansen and H.L. Larsen, Lecture Notes in Artificial Intelligence (LNAI) 1495, pp. 164–174. Ed. Springer, 1998.

[17] J. Kivinen, H. Mannila, Approximate dependency inference from relations, in: Proc. Fourth Internat. Conf. on Database Theory, 1992, pp. 86–98.

[18] J.C. Cubero, J.M. Medina, O. Pons, M.A. Vila “Fuzzy loss less decompositions in databases” Fuzzy Sets and Systems, Vol 97–2, pp. 145–167, 1998.

[19] J.C. Cubero, M.A. Vila, “A new definition of fuzzy functional dependency in fuzzy relational databases” International J. Intelligent Systems, Vol 9, pp. 441–449, 1994.

[20] Liang, G., & Wang, M. (1991). A fuzzy multi-criteria decision making method for facility site selection. International Journal of Production Research, 29(2), 2313–2330.

[21] M. Chen, J. Han, P.S. Yu ”Data Mining: An overview from a Data Base Perspective”. IEEE Transac. On Knowledge and Data Engineering, Vol 8–6 pp. 866–883, 1996.

[22] Martínez, L., J. Liu, and J.B. Yang, A Fuzzy Model for Design Evaluation Based on Multiple-Criteria Analysis in Engineering Systems. International Journal of Uncertainty , Fuzziness and Knowledge-Based Systems, 2006. 14(3): p. 317–336.

[23] Noël Novelli and Rosine Cicchetti. Functional and embedded dependency inference: a data mining point of view. Information Systems, Volume 26, Issue 7, November 2001, Pages 477–506.

[24] Pandian Vasant. Fuzzy decision making of profit function in production planning using S-curve membership function. Computers & Industrial Engineering, Volume 51, Issue 4, December 2006, Pages 715–725.

[25] Patrick Bosc, Donald Kraft and Fred Petry. Fuzzy sets in database and information systems: Status and opportunities. Fuzzy Sets and Systems, Volume 156, Issue 3, 16 December 2005, Pages 418–426.

[26] Pınar Yılmaz and Bülent Çatay. Strategic level three-stage production distribution planning with capacity expansion. Computers & Industrial Engineering, Volume 51, Issue 4, December 2006, Pages 609–620.

[27] R.A. Carrasco, J. Galindo, M.A. Vila, J.M. Medina, “Clustering and Fuzzy Classification in a Financial Data Mining Environment”. 3rd International ICSC Symposium on Soft Computing, SOCO’99, pp. 713–720, Genova (Italy), June 1999.

[28] R.A. Carrasco, J. Galindo, M.C. Aranda, J.M. Medina, M.A. Vila, “Classification in Databases using a Fuzzy Query Language”. 9th International Conference on Management of Data, COMAD’98, Hyderabad (India), December 1998.

[29] Ramón Alberto Carrasco, María Amparo Vila, Francisco Araque, “dmFSQL: a Language for Data Mining,” *dexa*, pp. 440–444, 17th International Conference on Database and Expert Systems Applications (DEXA’06), 2006.

[30] Seung Heon Han, Kyung Ho Chin and Myung Jin Chae. Evaluation of CITIS as a collaborative virtual organization for construction project management. Automation in Construction, Volume 16, Issue 2, March 2007, Pages 199–211.

[31] Su, C.-Y., Chen, S.-J. and Lin, L., “A Structured Approach to Measuring Functional Dependency and Sequencing of Coupled Tasks in Engineering Design,” Computers and Industrial Engineering, Vol. 45, No. 1, pp. 195–214, 2003.

[32] T. Akutsu, A. Takasu, Inferring approximate functional dependencies from example data, in: Proc. AAAI93 Workshop on Knowledge Discovery in Databases, AAAI, New York, 1993, pp. 138–152.

[33] T. Akutsu, S. Miyano and S. Kuhara, A simple greedy algorithm for finding functional relations: efficient implementation and average case analysis, Theoretical Computer Science, 292, 481–495, 2003.

[34] W.J. Frawley, G. Piatetsky-Shapiro, C.J. Matheus “Knowledge Discovery in Databases: An Overview” in G. Piatetsky-Shapiro, W.J. Frawley eds. “Knowledge Discovery in Databases” pp. 1–31, The AAAI Press, 1991.

[35] Zadeh, L.A., Fuzzy sets. Information and Control, 1965. 8: p. 338–353.

[36] Zadeh, L.A., The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Information Sciences, Part I, II, III, 1975. 8,8,9: p. 199–249, 301–357, 43–80.