A New Approach for Flexible Queries Using Fuzzy Ontologies

Amira Aloui and Amel Grissa

Abstract Motivated by the demand for formalized representation of outcomes of data mining investigations and the successful results of using Formal Concept Analysis (FCA) and Ontology, this chapter addresses the task of constructing an ontology of data mining in order to support flexible query in large Database using FCA and Fuzzy Ontology. A new approach for automatic generation of Fuzzy Ontology of Data Mining (FODM), through the combination of conceptual clustering, fuzzy logic and FCA will be presented. Then, a new algorithm to support database flexible querying using the generated fuzzy ontology will be defined. The approach starts with the organization of the data in homogeneous clusters having common properties which allows to deduce the data's semantic. Then, it models these clusters by an extension of the FCA. This lattice will be used to build a core of ontology. This ontology will be represented, then, as a set of fuzzy rules as an efficient answers to flexible queries. We show that this approach is optimum because the evaluation of the query is not done on the set of starting data which is huge but rather by using the generated fuzzy ontology.

Keywords Data Mining · Clustering · Formal Concept Analysis · Fuzzy Logic · Ontology

1 Introduction

The diversity of the Database (DB) applications showed the limits of the Relational Database Management Systems (RDBMS) in particular in the querying field [11]. The traditional querying of a Relational DB (RDB) is qualified by "Boolean querying"

A. Aloui (✉)
Ecole Nationale d'Ingenieurs de Tunis, LR-SITI, Tunis, Tunisia
e-mail: aloui_amira@yahoo.fr

A. Grissa
Ecole Nationale d'Ingenieurs de Tunis, LIPAH, FST, Tunis, Tunisia
e-mail: amel.touzi@enit.rnu.tn

A.T. Azar and S. Vaidyanathan (eds.), *Computational Intelligence Applications in Modeling and Control*, Studies in Computational Intelligence 575,
DOI 10.1007/978-3-319-11017-2_13

with SQL for example, a query returns a result or nothing at all [47]. This querying surrounds a problem for certain applications. First of all, the user must know all the details concerning the diagram and the data from the database to express his preferences or he should use imprecise linguistic terms as “moderate”, means” to better characterize the sought-after data.

The aim of the database flexible querying is to extend this binary behaviour by introducing preferences into the query criteria [40]. Thus, an element returned over by a query will be “more or less” relevant according to user preferences. Generally, the proposed approaches treat the flexible query in case of the RDB but not in case of the large DB. This work focuses on flexible query in large DB. For this purpose, we suggest the use of ontologies to improve the performance of retrieving information.

In fact, recent research showed that adopting formal ontology to describe heterogeneous data sources has many benefits. It provides not only a uniform and flexible approach to integrate and describe such sources, but it can also support the final user in querying them and improving the usability of the integrated system. Unfortunately, many deficiencies still exist in ontology. On the one hand, it is difficult to determine the granularity of ontology. On the other hand, the depth of concept expression of ontology is still not enough [6]. Thus fuzzy ontology is introduced to solve the above problems. The application of formal concept analysis and concept lattice theory in ontology building and mapping not only makes the building automatic, but also makes the newly generated ontology more formalized. It should be a better way to combine formal concept analysis with ontology to express and process the knowledge. This new proposed method supports the task of formulating a request for a user in a specific domain. In fact, the ontology defines a vocabulary which is often richer than the logical schema of the underlying data and usually closer to the users own vocabulary. The ontology can be effectively exploited by the user in order to formulate a query that best captures their information need. Consequently, the user is constantly guided and assisted in this task because the intelligence is dynamically driven by reasoning over the ontology.

This new approach helps the user in choosing what is more appropriate for him respecting their information need and restricting the possible choices which are more relevant and meaningful in a given context by considering only some parts of the ontology. For those reasons, the user is free to explore the ontology without the worry of making a wrong choice at some point and can thus concentrate on expressing his need. Besides, queries can be specified through a refinement process consisting in the iteration of few basic operations: The user specifies, first, an initial request, then before constructing the ontology, we use the fuzzy logic techniques [3] and Formal Concept Analysis concept [52] to classify the data which will refine or delete some of the not used information, thus the number of concepts constructing the ontology is always less than the number of objects starting on which we apply the classification algorithm [29] because the application of FCA reduces considerably the complexity until the resulting query satisfies the need of the user, changes the level of granularity in the process of the evaluation of the ontology and apply the clustering operation. So, the interrogation will focus necessarily on clusters.

Thus, we start by generating a Meta-DB formed by a set of clusters resulting of a preliminary fuzzy classification on data. This set represents a reduced view of the initial BD and allows to deduct semantics of the initial DB. The data classification aims to divide a data set into subsets, called clusters, so that:

- All data in the same cluster are similar and data from different clusters are dissimilar.
- The number of clusters generated by a classification algorithm is always less than the number of objects starting on which we apply the classification algorithm.
- All objects belonging to the same cluster have the same properties.

In this context, the query is modelled knowing the set of clusters modelling the meta-DB. To generate the meta-DB, we use the concepts of Clustering, Formal Concept Analysis (FCA) and Ontology. Thus, the use of these methods is justified by:

- Fuzzy clustering has been a very successful data analysis technique as demonstrated in diverse areas like signal processing, monitoring, and medical diagnosis [4]. Clustering is a widely used technique in data mining application for discovering patterns in underlying data. Most traditional clustering algorithms are limited in handling datasets that contain categorical attributes. Conventional clustering means classifying the given data objects as exclusive subsets (clusters).That means we can discriminate clearly whether an object belongs to a cluster or not. However such a partition is insufficient to represent many real situations. However, in many real situations, there not exists an exact boundary between different clusters. Therefore a fuzzy clustering object belongs to overlapping clusters with some membership degree. In other words, the essence of fuzzy clustering is to consider not only the belonging status to the clusters, but also to consider to what degree do the object belong to the cluster
- Fuzzy logic is derived from fuzzy set theory dealing with reasoning that is approximate rather than precisely deduced from classical predicate logic. Fuzzy logic is used with one of the web mining technique i.e. Clustering [21]. The advantages of Fuzzy Systems as well as various applications are presented thoroughly in [2].
- FCA is a method for knowledge representation that takes advantage of the features of formal concepts [51];
- Ontology is used for knowledge sharing and reuse. It improves information organization, management and understanding. It was introduced for better description of information in objective world. Ontology has a significant role in the areas dealing with vast amounts of distributed and heterogeneous computer based information [8, 23, 24].

The remainder of the chapter is organised as follows: A brief introduction to some basic definitions of Formal Concept Analysis (FCA), ontology and flexible querying in Sect. 2. Before we present our generic and automatic method for Fuzzy Ontology generation in Sect. 5, an overview of existing problems and contributions in Sects. 3 and 4 are given. Section 6 details the step of extracting flexible Query from resulted ontology. Section 7 evaluates the proposed approach. Section 8 summarizes the chapter, enumerates the advantages and concludes with an outlook on future work.

2 Basic Concepts

In this section, we present the basic concepts of flexible querying, ontologies and Formal Concept Analysis (FCA).

2.1 Flexible Querying

Definition A flexible query is a query in which comprise vague descriptions and/or vague terms.

The traditional systems of interrogation distinguish two categories of data: those which satisfy the search criteria and those which do not satisfy them. The principle of the flexible interrogation aims to extend this bipolar behaviour by introducing the concept of approximate pairing. Thus, an element returned by a request will be at least relevant according to its satisfaction degree to the constraints of interrogation.

Four principal approaches have been proposed to express and evaluate the flexible queries:

- Use of the secondary criteria [33, 52].
- Use of the distance and the similarity [19].
- Expression of the preferences with linguistic terms [38].
- Modelling of the inaccuracy by the fuzzy subsets theory [14, 44]. A comparative study of the systems of flexible interrogation has been achieved in [42, 49].

The problem of the expression of the users preferences in the flexible queries received much attention these last years [13, 20, 32]. In general, it is possible to distinguish two families of approaches for the expression of the preferences: implicit and explicit.

In the implicit approach: Mechanisms of numerical scores, commensurable or not, are used to represent the preferences. In the first case, the values of preferences can be aggregated to deliver a total value and to define a total order on the answers. In the second case, when there is not commensurability, only a partial order of the answers, based on the order of Pareto is possible for the incomparable classes of answers are built. This approach is detailed in Lietard and Rocacher [35] and is illustrated by the Skyline operator [10] or in PreferenceSQL [32].

In the explicit approach: The preferences are specified by binary relations of preferences and in the majority of the cases, a partial order is obtained on the tuples.

In addition, the preferences can be divided into constraints (preferences obligatory) and wishes (optional preferences). This reveals that this bipolar vision [27] of the preferences makes it possible to bring a refinement of the set of the answers to satisfy the constraints, then, if possible, wishes. The preferences of the users can also be expressed by criteria of selection based on fuzzy sets. The predicates are not any more "all or nothing" but can be "more or less" satisfied. Other researchers used charts to model the preferences on a great number of alternatives. As an example, we can quote the Conditional Preferences Networks (CP-Nets) [26], which constitute a chart appraisal for modelling the preferences.

Bosc and Pivert [15] suggest the introduction of the preferences in the form of subsets of n-uplets (stratified divisor). Thus, they used the terms of "stratified divisor" and "stratified division". Consequently, an element x of the dividend will be more acceptable as it will be associated with a large number of subsets (Si) defining the divisor. Three types of requests studied by Bosc et al. are expressed in SQL language, where the dividend can be an intermediate relation and the stratified divisor is given explicitly by the user or is the result from sub queries.

As example of principal systems of interrogation with preferences, we can quote, the systems PreferenceSQL and Preference Queries [12] which are based on a partial order, consequently, they deliver to the user the not dominating tuples. Preference SQL also incorporates a concept of bipolarity in the Preferring clause. The system top-K queries [11, 26], Domshlak uses an ad hoc score function f and delivers the k better answers of the total order obtained by f. However, this score function remains difficult to establish. The SQLf language uses the fuzzy set theory to define the preferences and makes the assumption of commensurability. It offers a framework founded to combine obligatory preferences.

Our work is related to the introduction of certain flexibility into the query writing. In fact, the traditional database querying uses a query to find elements satisfying a Boolean condition. In certain applications, the user can find a difficulty to describe in a precise and clear way the information for which he is seeking. It can also express preferences on the search criterion level with various degrees of importance between these criteria. This is why the concept of flexible query was proposed in the database systems. Let us consider for instance the case of a person who is looking, in an advertisement database, an apartment close to the town center with an approachable cost. In order to express such preferences, this person can formulate a flexible query comprising the terms *"near"* and *"accessible"*. It can also express the fact that the price criterion is more significant than that of the distance.

2.2 Ontologies

Ontologies are content theories about the classes of individuals, properties of individuals, and relations between individuals that are possible in a specified field of knowledge [18]. They define the terms for describing our knowledge about the domain. An ontology of a domain is beneficial in establishing a common (controlled) vocabulary when describing a domain of interest. This is important for unification and sharing of knowledge about a domain and its connection with other domains.

In reality, there is no common formal definition of what an ontology is. All the same, most approaches share a few core items such as: concepts, a hierarchical IS-A-relation, and further relations. For the sake of generality, we do not discuss more specific features like constraints, functions, or axioms in this paper, instead we formalize the core in the following way:

Definition 1 A (*core*) *ontology* is a tuple **O = (C, is_a, R, σ)** where

- C is a set whose elements are called concepts
- is_a is a partial order on C (i.e., a a binary relation is_a $\subseteq$ C × C which is reflexive, transitive, and anti symmetric)
- R is a set whose elements are called relation names (or relations for short)
- σ: R $\rightarrow$ C^{+} is a function which assigns to each relation name its arity

In the last years, several languages have been developed to describe ontologies. As example, we can cite, the Resource Description Framework (RDF) [17, 34], the Ontology Web Language (OWL) [7] and extension of OWL language like OWL 2 [37] or Fuzzy OWL [9]. The Web Ontology Language (OWL) is a family of knowledge representation languages for authoring ontologies and Description Logics (DL) are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. Today description logic has become a cornerstone of the Semantic Web for its use in the design of ontologies. The Web Ontology Language Description Logics (OWL DL) becomeless suitable in domains in which the concepts to be represented do not have precise definitions. In our case, this scenario is, unfortunately, likely the rule rather than an exception. To handle this problem, the use of fuzzy ontology offers a solution. Classical ontology languages are not appropriate to deal with imprecision or vagueness in knowledge. Therefore, DL for the semantic web can be enhanced by various approaches to handle probabilistic or possibilistic uncertainty and vagueness. Although fuzzy logic was introduced already in the 1960's [54], the research on fuzzy ontologies was almost non-existent before 2000, so we can claim that this is a fairly new research field with a great potential. This is even more surprising considering that Pena (1984) reasoned already in the 1980's why the use of fuzzy logic as the basis for ontology building would be beneficial and solve many problems pertaining to classical ontologies. He proposes "to reject the maximality rule, according to which only altogether true sentences are true, and embracing instead the rule of endorsement, which means that whatever is more or less true is true". Among the advantages of fuzzy ontology he mentions:

- Positing fuzzy predicates usually simplifies our theories in most scientific fields.
- Fuzzy predicates are much more plausible, and give us a much more attractive and cohesive worldview, than their crisp counterpart.
- Degree-talk and comparative constructions.

Also, the number of environments and tools for building ontologies have grown exponentially. These tools provide support for the ontology development process and for the subsequent ontology usage. Among these tools, we can mention the most relevant: Ontolinguav [28], WebOnto [25], WebODE [1], Protégé-2000 [39], OntoEdit [48] and OilEd [7].

In this work, we propose to use fuzzy-OWL2 language iteself to generate automatically scripts from fuzzy ontologies. More precise, we use Protégé 4.2 as an OWL2 editor for fuzzy ontology representation.

2.3 Fuzzy Conceptual Scaling and FCA

Conceptual scaling theory is the key part of a Formal Concept Analysis (FCA). It allows the introduction of the given data and embed much more general scales than the usual chains and direct products of chains. In the direct products of the concept lattices of these scales, the given data can be embedded. FCA starts with the notion of a formal context specifying which objects have what attributes and thus a formal context may be viewed as a binary relation between the object set and the attribute set with the values 0 and 1.

In Tran et al. [50], an ordered lattice extension theory has been proposed: Fuzzy Formal Concept Analysis (FFCA), in which uncertainty information is directly represented by a real number of membership values in the range of [0,1]. This number is equal to similarity which is defined as follows:

Definition 2 The similarity of a fuzzy formal concept $C_1 = (\varphi(A_1), B_1)$ and its subconcept $C_2 = (\varphi(A_2), B_2)$ is defined as:

$$S(C_1, C_2) = \frac{|\varphi(A_1) \cap \varphi(A_2)|}{|\varphi(A_1) \cup \varphi(A_2)|}$$

where ∩ and ∪ refer intersection and union operators on fuzzy sets, respectively.

In Sassi et al. [45], we showed that these FFCA are very powerful as well with the interpretation of the results of the Fuzzy Clustering and in the optimization of the flexible query.

3 Related Work

Many researchers in the field of data mining have tried to find the efficient way to respond to the user query. We study in this section the most important approaches that generate information from data.

- **Approaches based on concept lattices for information retrieval.**

A first detailed formalization of how to use lattices for information retrieval appears to date back to Mooers (1958). His approach is contained in Salton's (1968) famous book and originally received some attention [46] but has not been further elaborated in the mainstream information retrieval community. Most of the few, current applications of lattices in information retrieval are based on formal concept analysis [29], which was invented in the early 1980's and relates lattices to object-attribute matrices or document-term matrices in information retrieval. Formal concept analysis applications to information retrieval are similar to Mooers's ideas but have been developed independently.

Quan Thanh et al. [43] proposed to incorporate fuzzy logic into FCA to make FCA deal with uncertainty in data and reasonably interpret the concept of hierarchy. The proposed framework is known as Fuzzy Formal Concept Analysis (FFCA). They use FFCA for automatic generation of ontology for scholarly semantic web. Concept lattices have been also applied in search of information at the onset of formal concept analysis [29]. A restriction of information retrieval by lattice is the theoretical complexity of the number of concepts for a context large number of objects or properties. More solutions to control the size of the lattice corresponding to the major contexts have been proposed in our approach.

- **Approaches based on domain ontology to improve the performance of information retrieval.**

The ontology building is usually performed manually, but researchers try to build an ontology automatically or semi automatically to save the time and the efforts of building the ontology.

Clerkin et al. [22] used concept clustering algorithm (COBWEB) to automatically discover and generate ontology. They argued that such an approach is highly appropriate to domains where no expert knowledge exists, and they propose how they might employ software agents to collaborate, in the place of human beings, on the construction of shared ontologies.

Wuermli and Joller [53] used different ways to build ontologies automatically, based on data mining outputs represented by rule sets or decision trees. They used the semantic web languages, RDF, RDF-S and DAML + OIL for defining ontologies. The problem with those approaches is that they are constructed ontology that do not describe the complete domain of data mining, but are simply made with a specific task in mind. Also, some existing ontology-based information retrieval approaches use RDF [41].

Mena et al. [36], Baer et al. [5], and Kapetanios et al. [31] provided insufficient knowledge for query reformulation. These approaches also lack the details of what needs to be included in the ontology from the data sources along with the domain knowledge to drive the process of query reformulation. The focus of these approaches (for example Kapetanios et al. [31]) remains towards interactive query generation through nondirected graphs supporting multiple natural languages.

- **Approaches based on Query to improve performance.**

Four principal concepts were proposed in the traditional approaches to express and evaluate the flexible queries:

- The use of the secondary criteria,
- The expression of the preferences with linguistic terms,
- The modeling of the inaccuracy by the fuzzy subset theory.

Ounalli and Belhadj [40] proposed a relieving approach within the fuzzy set framework. This approach appears too promising. The first contribution is to take into consideration the semantic dependencies between the query research criteria and to determine its reliability or not. The second contribution relates to its cooperative aspect in the flexible interrogation. For the dependencies extraction, this approach consists on building TAH's and MTAH from relieving attributes. The problem here lies in storage, indexing of such structures and the incremental update of these structures. To fullfill such works, fundamental research was focused on the following problems:

- Flexible queries formulation and evaluation,
- Vague or fuzzy data description and processing,
- Definition and use of fuzzy dependencies,
- Fuzzy Data Mining [30].

4 Motivation and Contributions

We have faced two types of problems:

- **At the level of flexible query:** The majority of the current approaches presented to support flexible queries have several limits, in particular, in The consideration of the dependencies between the search criteria that permit to detect the unreliable requests (having an empty answer) with the user, and the generation of the turned over approximate answers.
- **At the level of the ontology approaches:** several approaches have been proposed, but, generally these authors don't propose any solutions for the evaluation of the queries knowing onthologies generated by their approaches.

Several algorithms for Data Mining try to trace the decision tree or the FCA or one of these extensions to extract the association rules. In this case, researchers always focus on giving an optimum set of rules modeling in a faithful way the starting data unit, after having done a data cleansing step and an elimination of invalid-value elements. Accordingly, the limits of these approaches reside in the extraction of this ontology starting from the data or a data variety, which may be huge. As a result, we note the following limits:

- The rules generated from these data are generally redundant rules.
- These algorithms generated a very big number of rules, almost thousands, that the human brain cannot even assimilate.
- Generally the goal from extracting a set of rules is to help the user to give semantics of data and to optimize the information research. This fundamental constraint is not taken into account by these approaches.

To resolve these problems, we propose:

- A new approach for the ontology generation using conceptual clustering, fuzzy logic, and FFCA. We propose to define rules (Meta-Rules) between classes resulting from a preliminary classification on the data. Indeed while classifying data, we construct homogeneous groups of data having the same properties, so defining rules between clusters implies that all the data elements belonging to those clusters will be necessarily dependent on these same rules. Thus, the number of generated rules is smaller since one processes the extraction of the knowledge on the clusters which number is relatively lower compared to the initial data elements.
- A new algorithm to support database flexible querying using the generated knowledge in the first step. This approach allows the end-user to easily exploit all knowledge generated.

5 Presentation of the Fuzzy Ontology of Data Mining: FODM

5.1 Principe of the FODM

In this section, we present the architecture of the Fuzzy Ontology of Data Mining (FODM) approach and the process of fuzzy ontology construction.

Our FODM approach takes the database records and provides the corresponding fuzzy ontology. Figure 1 shows our proposed FODM approach. We suggest the ontology definition between classes resulting from a preliminary classification of the data. The FODM approach is organized according to two following main steps. Data Organization step and Fuzzy Ontology Generation step.

5.2 Theoretical Foundation of the FODM Model

In this part, we provide the theoretical foundations of the proposed approach, based on the following properties:

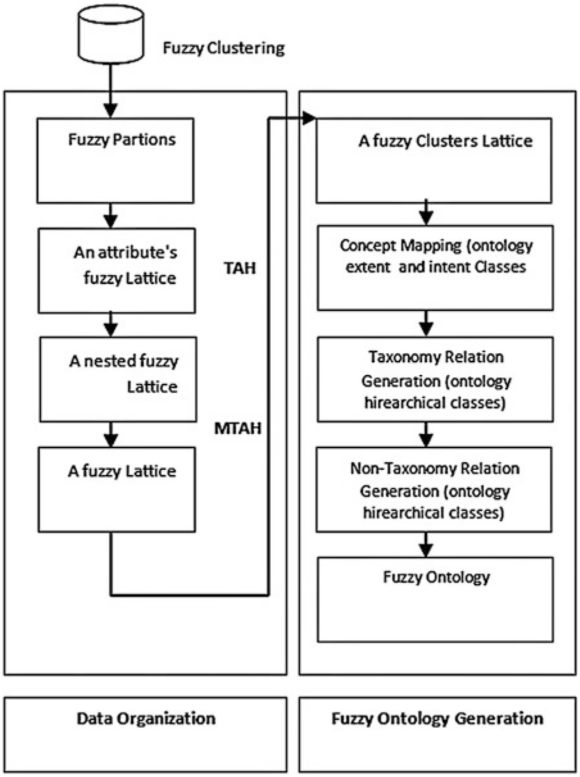

Fuzzy Clustering
Fuzzy Partions
An attribute's fuzzy Lattice
TAH
A nested fuzzy Lattice
MTAH
A fuzzy Lattice
A fuzzy Clusters Lattice
Concept Mapping (ontology extent and intent Classes
Taxonomy Relation Generation (ontology hirearchical classes)
Non-Taxonomy Relation Generation (ontology hirearchical classes)
Fuzzy Ontology
Data Organization
Fuzzy Ontology Generation

Fig. 1 Presentation of the fuzzy ontology of data mining approach

Property 1

- The number of clusters generated by a classification algorithm is always lower than the number of starting objects.
- All objects belonging to one same cluster have the same proprieties. These characteristics can be deduced easily knowing the center and the distance from the cluster.
- The size of the lattice modeling the properties of the clusters is lower than the size of the lattice modeling the properties of the objects.
- The management of the lattice modeling the properties of the clusters is optimum than the management of the lattice modeling the properties of the objects.

Property 2 Let C1, C2 be two clusters, generated by a classification algorithm and verifying respectively the properties p1 and p2. Then the following properties are equivalent:

$$\mathrm{C1} \Rightarrow \mathrm{C2}\,(\mathrm{CR}) \Leftrightarrow$$

- ∀ object O1∈C1 ⟹ O1∈C2 (CR),
- ∀ object O1∈C1, O1 checks the property p1 of C1 and the property p2 of C2 (CR)

Property 3 Let C1, C2 and C3 be three clusters generated by a classification algorithm and verifying respectively the properties p1, p2 and p3 respectively. Then the following properties are equivalent:

$$\mathrm{C1}, \mathrm{C2} \Rightarrow \mathrm{C3}\ (\mathrm{CR}) \Leftrightarrow$$

- ∀ object O1 ∈ C1 ∩ C2 ⟹ O1 object ∈ C3 (CR).
- ∀ object O1 ∈ C1 ∩ C2 then O1 checks the properties p1, p2 and p3 with (CR).

The validation of the two properties come from to the fact that all objects which belong to a same cluster check necessarily the same attribute as their cluster.

5.3 Data Organization Step

This step gives a certain number of clusters for each attribute. Each tuple has values in the interval [0,1] representing these membership degrees. Linguistic labels, which are fuzzy partitions, will be assigned to the attributes. This step consists on TAH's and MTAH generation of relieving attributes. This step is very important in the Fuzzy ontology generation process because it allows to define and interprete the distribution of objects in the various concepts.

Example Let's have a relational database table presented by Table 1 containing the list of AGE and SALARY of Employee

Table 2 presents the results of fuzzy clustering applied to Age and Salary attributes. For Salary attribute, fuzzy clustering generates three clusters (C1, C2 and C3). For the attribute AGE, two clusters have been generated (C4 and C5).

Table 1 Relational database table

	Salary	Age
A_1	800	30
A_2	600	35
A_3	400	26
A_4	900	40
A_5	1,000	27
A_6	500	30

Table 2 This fuzzy conceptual scales for age and salary attributes

	Salary			Age	
	C1	C2	C3	C4	C5
A_1	0.1	0.5	0.4	0.5	0.5
A_2	0.3	0.6	0.1	0.4	0.6
A_3	0.7	0.2	0.1	0.7	0.3
A_4	0.1	0.4	0.5	0.2	0.8
A_5	–	0.5	0.5	0.6	0.4
A_6	0.5	0.5	–	0.5	0.5

Table 3 This fuzzy conceptual scales for age and salary attributes with α-cut

	Salary			Age	
	C1	C2	C3	C4	C5
A_1	–	0.5	0.4	0.5	0.5
A_2	0.3	0.6	–	–	0.6
A_3	0.7	–	–	0.7	–
A_4	–	0.4	0.5	–	0.8
A_5	–	0.5	0.5	0.6	–
A_6	0.5	0.5	–	0.5	0.5

We apply an α-cut to the set of membership degrees, to replace these last by values 1 and 0 and to deduce the binary reduced formal context.

In our example, α-cut (Salary) = 0.3 and, α-cut (Age) = 0.5, so, the Table 2 can be rewritten as shown in Table 3.

The corresponding fuzzy concept lattices of fuzzy context presented in Table 3, noted as TAH's are given by the line diagrams presented in the Figs. 2 and 3.

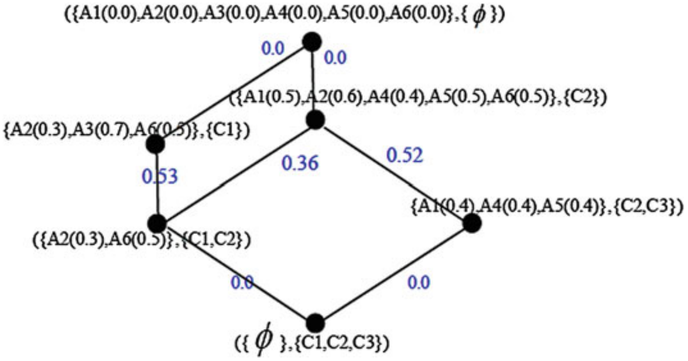

({A1(0.0),A2(0.0),A3(0.0),A4(0.0),A5(0.0),A6(0.0)},{ ϕ })
0.0
0.0
({A1(0.5),A2(0.6),A4(0.4),A5(0.5),A6(0.5)},{C2})
{A2(0.3),A3(0.7),A6(0.5)},{C1})
0.53
0.36
0.52
{A1(0.4),A4(0.4),A5(0.4)},{C2,C3})
({A2(0.3),A6(0.5)},{C1,C2})
0.0
0.0
({ ϕ },{C1,C2,C3})

Fig. 2 Salary TAH

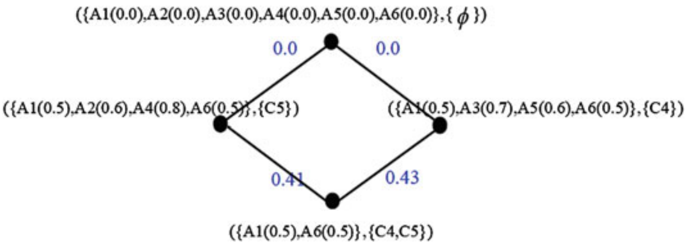

({A1(0.0),A2(0.0),A3(0.0),A4(0.0),A5(0.0),A6(0.0)},{ ϕ })
0.0
0.0
({A1(0.5),A2(0.6),A4(0.8),A6(0.5)},{C5})
({A1(0.5),A3(0.7),A5(0.6),A6(0.5)},{C4})
0.41
0.43
({A1(0.5),A6(0.5)},{C4,C5})

Fig. 3 Age TAH

The minimal (resp. maximal) value of each cluster corresponds to the lower (resp. higher) interval terminal of the values of this last. Each cluster of a partition is labelled with a ***linguistic label*** provided by the user or a domain.

The Table 4 presents the correspondence between the linguistic labels and their designations for the attributes Salary and Age.

The fuzzy concept lattices of fuzzy context presented in Table 5, noted as TAH's are given by the line diagrams presented in Figs. 2 and 3.

This very simple sorting procedure gives us for each many-valued attribute the distribution of the objects in the line diagram of the chosen fuzzy scale. Figure 4 shows the fuzzy nested lattice constructed from Figs. 2 and 3.

Table 4 Correspondence of the linguistic labels and their designations

Attribute	Linguistic labels	Designation
Salary	Low	C1
Salary	Medium	C2
Salary	High	C3
Age	Young	C4
Age	Adult	C5

Table 5 Fuzzy conceptual scales for age and salary attributes with α-cut

	Salary			Age	
	C1	C2	C3	C4	C5
	Low	Med	High	Young	Adult
A_1	–	0.5	0.4	0.5	0.5
A_2	0.3	0.6	–	–	0.6
A_3	0.7	–	–	0.7	–
A_4	–	0.4	0.5	–	0.8
A_5	–	0.5	0.5	0.6	–
A_6	0.5	0.5	–	0.5	0.5

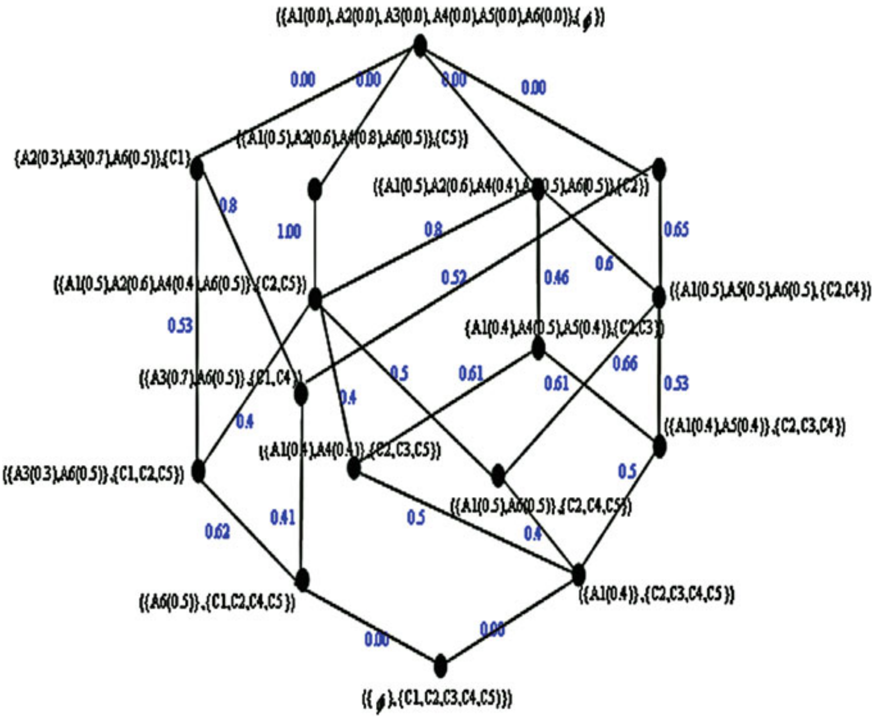

({A1(0.0), A2(0.0), A3(0.0), A4(0.0),A5(0.0),A6(0.0)},{ϕ})
0.00
0.00
0.00
0.00
({A1(0.5),A2(0.6),A4(0.8),A6(0.5)},{C5})
{A2(0.3),A3(0.7),A6(0.5)},{C1}
({A1(0.5),A2(0.6),A4(0.4),A5(0.5),A6(0.5)},{C2})
0.8
1.00
0.8
0.65
({A1(0.5),A2(0.6),A4(0.4),A6(0.5)},{C2,C5})
0.52
0.46
0.6
({A1(0.5),A5(0.5),A6(0.5)},{C2,C4})
0.53
{A1(0.4),A4(0.5),A5(0.4)},{C2,C3}
({A3(0.7),A6(0.5)},{C1,C4})
0.5
0.61
0.61
0.66
0.53
0.4
0.4
({A1(0.4),A5(0.4)},{C2,C3,C4})
({A1(0.4),A4(0.4)},{C2,C3,C5})
({A3(0.3),A6(0.5)},{C1,C2,C5})
0.5
({A1(0.5),A6(0.5)},{C2,C4,C5})
0.62
0.41
0.5
0.4
({A6(0.5)},{C1,C2,C4,C5})
({A1(0.4)},{C2,C3,C4,C5})
0.00
0.00
({ϕ},{C1,C2,C3,C4,C5})

Fig. 4 Fuzzy lattice: MTAH

5.4 *Fuzzy Ontology Generation Step*

This step consists on constructing the Fuzzy Ontology. It aims to deduce the Fuzzy Cluster Lattice corresponding to MTAH lattice generated in the first step, then to generate Ontology Extent and Intent Classes, Ontology hierarchical Classes, Ontology Relational Classes and finally the Fuzzy Ontology.

Definition (*Fuzzy Clusters Lattice*) A *fuzzy Clusters Lattice* (FCL) of a Fuzzy Formal Concept Lattice, consists on a Fuzzy concept lattice for which each equivalence class (i.e., a node of the lattice) contains only the intentional description (intent) of the associated fuzzy formal concept. This lattice will be used to build the core of the ontology.

Definition (*Level of FCL*) A *level i* of FCL is a is the set of nodes of FCL with cardinality equal to i.

Definition (*Concept Hierarchy*) A concept hierarchy is a poset (partially ordered set) (H, <), where H is a finite set of concepts and < is a partial order on H.

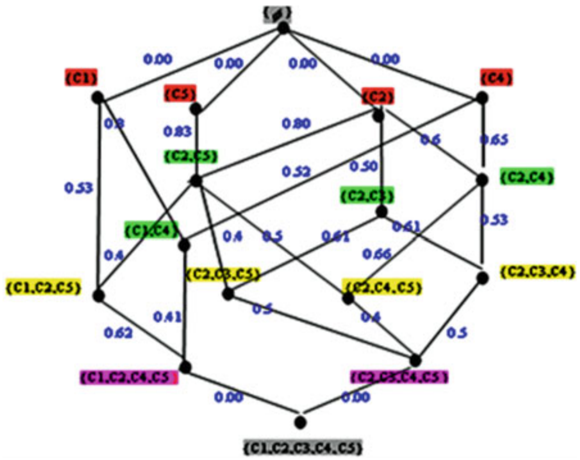


Fig. 5 Fuzzy clusters lattice (*FCL*)

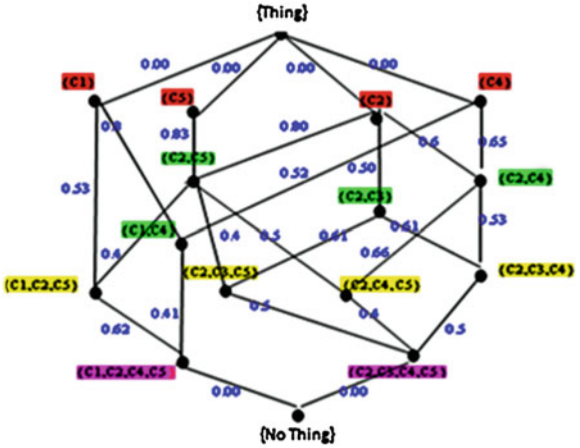


Fig. 6 Fuzzy ontology lattice

We make in this case a certain abstraction on the list of the objects with their degrees of membership in the clusters. The nodes of FCL are clusters ordered by the inclusion relation. As shown in the Fig. 5, we obtain a lattice more reduced, simpler to traverse and to store. Figure 6 illustrates the hierarchical relations constructed from the conceptual clusters given in 5. Each concept in the concept hierarchy is represented by a set of its attributes.

The supremum and infimum of the lattice are considered respectively as “Thing” and “Nothing” concepts.

The next step constructs fuzzy ontology from a fuzzy context using the concept hierarchy created by fuzzy conceptual clustering. This is done because both FCA and ontology support formal definitions of concepts. Thus, we define the fuzzy ontology as follows:

Definition (Fuzzy Ontology) A fuzzy ontology Fo contains of four elements (C, A^C, R, X), where:

- C represents a set of concepts,
- A^C represents a collection of attribute sets, one for each concept,
- R = (R_T; R_N) represents a set of relationships, having two elements: R_N is a set of non taxonomy relationships and R_T is a set of taxonomic relationships.
- X is a set of axioms. Each axiom in X is a constraint on the concept's and relationship's attribute values or a constraint on the relationships between concept objects.

We briefly describe the ontology mapping process from context to an ontology. The principal schema of one to one corresponding relations among the elements of FCA and OWL ontology are shown in Fig. 7.

- **Concept Mapping:** The mapping of concepts is one of the important stages of construction of a Fuzzy Ontology from a lattice of fuzzy concepts. It maps the extent and intent of the fuzzy context into the extent and intent classes of the ontology.
- **Taxonomy Relation Generation:** It expands the intent class of the ontology as a hierarchy of classes using the concept hierarchy. The process can be considered as an isomorphic mapping from the concept hierarchy into taxonomy classes of the ontology.
- **Non-taxonomy Relation Generation:** It generates the relation between the extent class and intent classes. This task is quite straight forward. However, we still need to label the non-taxonomy relation.
- **Instances Generation:** It generates instances of the extent class. Each instance corresponds to an object in the initial fuzzy context. Based on the information available on the fuzzy concept hierarchy, instances, attributes are automatically furnished with appropriate values.
- **Semantic Representation Conversion:** The generated ontology with concept hierarchies in Protégé-2000 [39] is shown in Fig. 8. This schema introduces the transformation rules for the automatic generation of OWL ontology based on the analysis of the concept hierarchy derived from FCA.

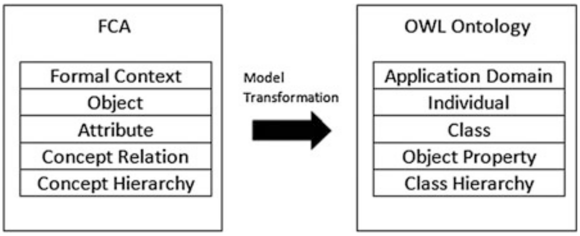


Fig. 7 From FCA to OWL ontology

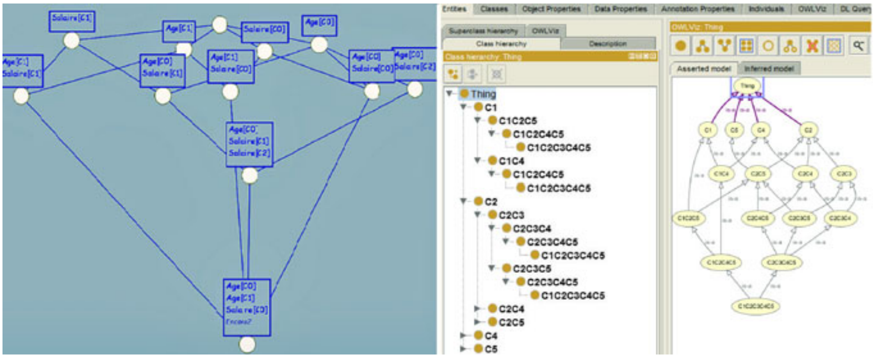


Fig. 8 From cluster FCA to Protégé-2000 to querying

5.5 Algorithm for Automatically Generating a Fuzzy Ontology

This section presents the main steps of our algorithm:

1. Assign the superclass “Thing” to the root of the concept lattice noted TOP.
2. For each concept of the level 1 of the lattice:
   (a) Create a subclass of the root class: Thing.
   (b) Assign the value 0 as a membership degree to the Thing class.
3. From the level 1, traverse the lattice and for each concept:
   (a) Find its successors/sub-concepts.
   (b) Put the given subconcepts in a set noted SetSucc.
4. Browse all successors of the SetSucc: For each successor belonging to SetSucc:
   (a) Create a subclass of the previous class.
   (b) Assign a membership value from each super concept.

The algorithm for generating a Fuzzy Ontology is described bellow:

Algorithm 1: Generation algorithm from FCL to Fuzzy Ontology

```
Input: Fuzzy Concept Lattice
Output: Fuzzy Ontology
begin
    initialization;
    Nf= Number of fuzzy concept;
    Nf1 = Number of fuzzy concept from level 1;
    hasValue = Create DataProperty;
    Thing Concept = 0, TOP;
    for i :=1 to Nf1 do
        Concept (i) = class (i);
        Class (i) = Create subclass From (Thing);
        Class (i) = HasValue (0.0) From (Thing);
    end
    for i:=1 to Nf do
        SetSucc = Succ (concept (i));
        for j:=1 to Number (SetSucc) do
            Under Class (i) = SetSucc (j);
            Under Class (i) = HasValue (SetSucc (j)) From SetSucc (j)
        end
    end
end
```

An example of ontology mapping is illustrated in Fig. 8

In this example, we used the domain “Employe” to illustrate the building process and to evaluate the resulted queries. We deduce the fuzzy lattice from our platform “ClusterFCA”, then we construct the ontology using Protégé-4.2 while taking the FCA as a guideline.

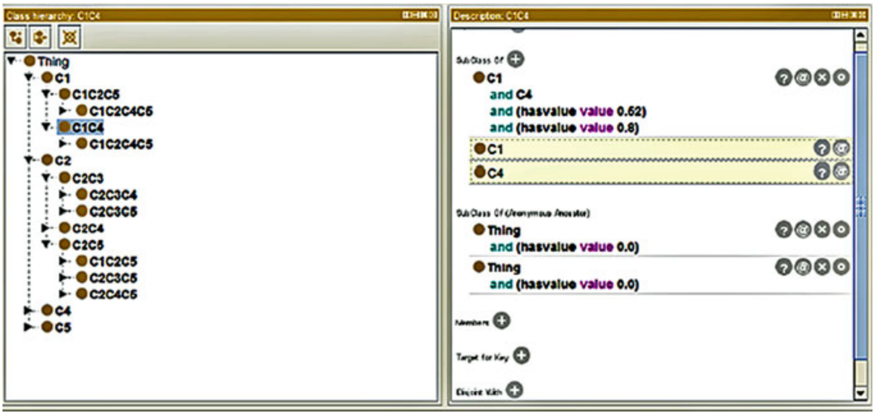
We consider nodes as concepts. The name of the concept as linguistics label. Nevertheless, taxonomic relationships between concepts are presented in the lattice. These classes visualization is done in Asserted class hierarchy of Protégé and the view is offered by OWLViz Plugin.

5.6 Mapping Ontology to Queries

Next step is to provide means for transforming concept lattice based ontology expression to associations rules. This process enables to produce logical expression of ontology lattice and specify intended semantics of the descriptions in first order logic. Once the ontology are defined, thus we can model the resulted rules deduced from our Fuzzy Ontology using Protege 4.2 software as bellow (Fig. 9):

In order to define non-taxonomic relationships the following groups of rules are defined:

- **Properties of concepts:** For properties of concepts definition, the following predicate can be used: has property (Concept name, Property name).
- **Inheritance of properties:** Inheritance of properties can be represented by the following rule: has property (C1, X) ← is a (C1,C2), has property (C2, X).



Class hierarchy: C1C4

```
Thing
  C1
    C1C2C5
      C1C2C4C5
    C1C4
      C1C2C4C5
  C2
    C2C3
      C2C3C4
      C2C3C5
    C2C4
    C2C5
      C1C2C5
      C2C3C5
      C2C4C5
  C4
  C5
```

Description: C1C4

```
SubClass Of
  C1
    and C4
    and (hasvalue value 0.62)
    and (hasvalue value 0.8)
  C1
  C4

SubClass Of (Anonymous Ancestor)
  Thing
    and (hasvalue value 0.0)
  Thing
    and (hasvalue value 0.0)

Members

Target for Key

Disjoint With
```

Fig. 9 Fuzzy association rules

- **Ontological relationships:** Like part-of, related-to, etc., can be easily represented via predicates. The following predicates demonstrate opportunities adding other ontological relationships: part of (C1, C2), related (C1, C2), synonyms (C1, C2), etc.

6 Presentation of FODM-FQ: FODM to Flexible Query

The architecture of the FODM-FQ is detailed in Fig. 10.

In this section, a new method to support database flexible querying using the generated knowledge deduced from the FODM model is defined.

It has three steps: the first and the second step performs the data organization and Fuzzy ontology generation described on the Sect. 5. We focus now on step3.

6.1 Flexible Query Algorithm

A flexible and cooperative database flexible querying approach within the fuzzy ontology framework has been proposed. This approach takes into account the semantic dependencies between the query and the search criteria to determine its realizability or not. Thus the idea is to change the level of granularity and apply the clustering operation, so the interrogation will focus necessarily on clusters.

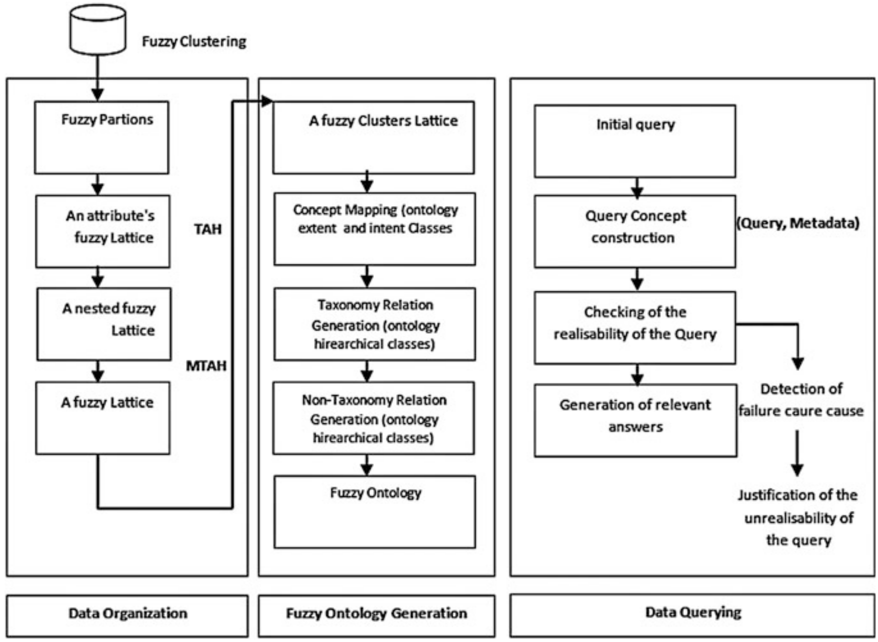

Fuzzy Clustering
Fuzzy Partions
An attribute's fuzzy Lattice
TAH
A nested fuzzy Lattice
MTAH
A fuzzy Lattice
A fuzzy Clusters Lattice
Concept Mapping (ontology extent and intent Classes
Taxonomy Relation Generation (ontology hirearchical classes)
Non-Taxonomy Relation Generation (ontology hirearchical classes)
Fuzzy Ontology
Initial query
Query Concept construction
(Query, Metadata)
Checking of the realisability of the Query
Generation of relevant answers
Detection of failure caure cause
Justification of the unrealisability of the query
Data Organization
Fuzzy Ontology Generation
Data Querying

Fig. 10 Presentation of the new approach: FODM to flexible query

Next step presents our flexible query algorithm using the generated ontology in the second step. Let R the user Query. The pseudo-code for flexible query generation algorithm is given:

```
Algorithm 2: Flexible query generation
Input: The user Query R
Output: List of answers
begin
    Concept_Query(R,$\Phi_B$);
    let i =Cardinality($\Phi_B$);
    Examine the rules in level i;
    if there is a rule which uses all the elements of $\Phi$ then
        R is realizable with CF = 100%;
        Extract(R, i);
    else if there is a rule which uses at least one elements of the $\Phi_B$
    then
        R is realizable with CF< 100%;
        Extract(R, i);
    else
        R is not realizable;
    end
end
```

Note that:

- **Concept_Query (R, Φ_B):** is a procedure that determine the concept Φ_B of R.
- **Extract (R, i):**is a procedure that determines answers of the request while using the Backward chaining. This procedure calls upon all the rules closely related to the request of level ≤ i

Applying this algorithm, the generated knowledge is in the form of rules. We obtain a query concept $\Phi = (\Phi_A, \Phi_B)$.

6.2 Construction of the Query Concept

We define a query concept $\Phi = (\Phi_A, \Phi_B)$ where Φ_A is a name to indicate a required extension and Φ_B is the set of classes describing the data reached by the query.

The set of classes Φ_B is determined by the following procedure:

```
Input: Vector V(A)=vj : j=1,...,C(A) of $\Phi$ cluster centres of relievable
       attribute A and the value of associated to this last.
Output: Query concept $\Phi$=($\Phi_A$,$\Phi_B$)
begin
   Step 1: Calculate the membership degrees of the specified clusters
   for each value of the criterion of $\Phi$ associated to the relievable
   attribute A;
   Step 2: Apply $\alpha$-Cut to generate the fuzzy context;
   Step 3: Form the set $\Phi_B$ of clusters whose membership is higher
    than the $\alpha$-Cut value;
end
```

Example For better explaining this step, we consider a relational database table describing apartment announces. The query is as follows:

$$Q\left\{\begin{array}{llll}
\textbf{Select} & \mathit{ref\,An}, & \mathit{price}, & \mathit{surface} \\
\textbf{From} & \mathit{Announce}, & \mathit{Appartment} & \\
\textbf{Where} & \mathit{price} = & 105 & \\
(A1) & & & \\
\textbf{and} & \mathit{surface} = & 75 & \\
(A2) & & & \\
\textbf{and} & \mathit{city} = & '\mathit{Paris}' &
\end{array}\right. \tag{1}$$

In this query, the user wishes that its preferences be considered according to the descending order: Price and Surface. In other words, returned data must be ordered and presented to the user according to these preferences. Without this flexibility, the user must refine these search keys until obtaining satisfaction if required since he does

Table 6 Query memberships degrees

Price			Surface	
C1	C2	C3	C4	C5
0.1	0.3	0.6	0.2	0.8

Table 7 Query memberships degrees with α-Cut

Price			Surface	
C1	C2	C3	C4	C5
–	0.3	0.6	–	0.8

not have precise knowledge on the data which he consults. According to the criteria of the query ϕ only the **A1** and **A2** criteria correspond to relievable attributes.

Initially, we determine starting from the DB the tuples satisfying the non relievable criteria (A_3, A_4, A_5), result of the following query:

$$Q\begin{cases}\textbf{Select} & refAn, & price, & surface\\ \textbf{From} & Announce, & Appartment & \\ \textbf{Where} & city\ = & 'Paris' & \\ (A3) & & & \\ \textbf{and} & place\ = & '16^{eme}arrondissement' & \\ (A4) & & & \end{cases} \tag{2}$$

These tuples is broken up into clusters according to labels of the relievable attributes *Price* and *Surface*. The query concept are given with part of the fuzzy clustering operation to determine the objects membership's degrees in the various clusters. Table 6 present the membership degrees associated to the query. These degrees are obtained while basing on memberships matrix obtained by a fuzzy clustering algorithm.

Then, we apply the α-Cut for each attribute to minimize the number of concepts. We obtain the reduced context request presented by the Table 7.

According to our example, the query Q seek the data sources having the metadata $Q\{\text{C2, C3, C5}\}$.

6.3 Checking of the Query Realisability

If the query criteria are in contradiction with their dependences extracted from the database, it is known as unrealisable.

Proposition *Let a query Q having the concept $\Phi = (\Phi_A,\Phi_B)$. A query Φ_A is unrealisable if and only if there is no data source in Φ_A which divide any metadata of the set Φ_A.*

This proposition of relevance is at the base of the research process. It is different from the vicinity concept used in [16], which can lead to obtain data that don't share any metadata with the initial query and don't correspond to the end-user needs.

Table 8 Characteristics of datasets

Data Set	Nb of objects	Size of objects	Nb of items
C20d10 K	10,000	20	386
T25i10d10 K	1,000	25	1,000
Mushrooms	8,416	23	128
Car	1,728	7	26
Achat	28	5	5

7 Evaluation of the Proposed Approach

The performance of the proposed algorithm for Discovering Fuzzy queries can be measured in order to evaluate the generated ontology. To do this, we compare two approaches using 4 datasets known on the ECD field (Table 8).

The first approach does not apply the clustering concept and the second uses the formal concepts for structuring and building ontology-based classification with AFC adopted by “ClusterFCA”. ClusterFCA is a java platform developed by our team. It includes a classification module containing algorithms for binary and fuzzy clustering. It also includes an AFC module for the construction of simple and nested lattice.

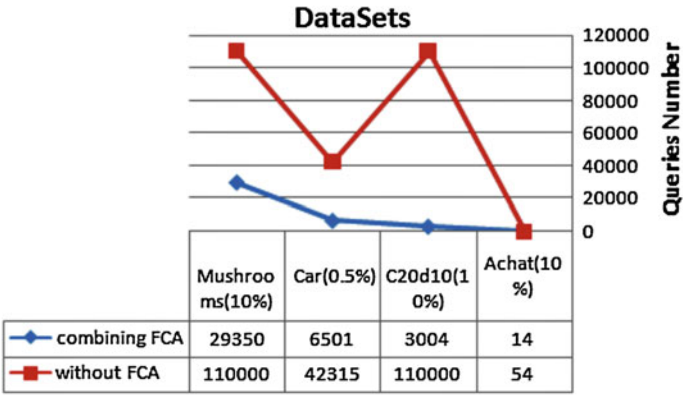
In this chart we show the number of rules resulting from these data sets: Mushrooms (8,416 objects), C20d10 K (10,000 objects), Car (1,728 objects), Achat (28 objects).

The existing algorithms dont take into account any semantics of the data. All the researchers focused themselves on the reduction of the set of rules, by proposing the concept of metadata, or on the method of visualization of this rules. Our main contribution resides in extracting the ontology from datasets by using FCA and transforming it to a rule language in order to model the expression of the user’s preferences and generate the relevant answers. Thus, we prove in Fig. 11 that with FCA, we minimize the space complexity of the resulting lattice. The combination of two concepts: FCA and ontology models a certain abstraction of the data that is fundamental in the case of an enormous number because the defined ontology is deduced from clusters not from the initial objects. The flexible query approach proposed the followings contributions compared to the similar approaches

- The automatic generation of TAH’s and MTAH from relieving attributes.
- The research of relevant data sources for a given query.
- A detection of the query unrealisability.
- The scheduling of the results.

Different advantages are granted by the proposed approach. This approach is:

- More reliable compared to the classic one (without clustering). In the examples, the number of classes generated in the case of the application of our new approach is less than the number of classes of input ontology. The decrease



	Mushrooms(10%)	Car(0.5%)	C20d10(10%)	Achat(10%)
◆ combining FCA	29350	6501	3004	14
■ without FCA	110000	42315	110000	54

Fig. 11 Metrics of the proposed approach

mainly depends on the number of clusters that choice, leads to a considerable reduction in the number classes composed the ontology. It retrieve pertinent informations which still more meaningful to the end-user and allows him to easily exploit all knowledge generated. The decrease of association rules mainly depends on the number of chosen clusters.

- Applicable to any type/amount of data: As part of our support system to build an ontology using the AFC, the experts in each field could reach a mass of acceptable information. Indeed, formal Concept Analysis helped us to structure and then build ontologies because the FCA was able to express ourselves in an fuzzy ontology format. The lattice is easy for people to understand and can be used as a guideline for the construction of ontologies. We gave also an example to show the power of the formal concept analysis.
- Applicable with any fuzzy classification algorithm to classify the initial data.

8 Conclusion

This paper focuses on how the future retrieval information from large dataset might look like and the interaction between the sources of information to yield perfect and real time results with unique power of intelligence is done to interpret the best possible solution for the user query.

The main idea is to collaborate the search to be more informative and provide it an intelligence in order to retrieve user oriented results. The model defined for this approach is FODM-FQ. It consists of four steps: The first organizes the database records in homogeneous clusters having common properties to deduce the data's semantic. This step consists of TAH's and MTAH generation of relieving attributes. The second step called Discovering Knowledge is used to deduce the Fuzzy Cluster Lattice corresponding to MTAH lattice generated in the first step. Then, on the third step, the FCL is mapped to an owl ontology design. From this ontology, the rules modeling the Knowledge (Set of Fuzzy Associations Rules on the attributes SFR) are extracted. We prove that the discovered rules does not contain any redundant rule. The fourth step ensures database flexible querying using the generated ontology.

An example of an ontology was simulated using Protege and results were analyzed. The keywords entered by the user were given priority and bases on that we have discarded certain resulted query using FCA methodology which make the search more compact and effective. The future scope of this method is to first integrate the current into large domains resulting in expansion of knowledge base. Secondly, an intelligent distributed ontology query processing method will be proposed to deal with the growth of the data size and the number of distributed queries which access the common part of the resources and successfully meet the user preferences.

References

1. Arpirez, J., Corcho, O., Fernandez-Lopez, M., Gomez-Perez, M.: WebODE: a workbench for ontological engineering. In: First International Conference on Knowledge Capture (K-CAP01), Victoria, pp. 613 (2001)
2. Azar, A.T.: Fuzzy Systems. IN-TECH, Vienna (2010). ISBN 978-953-7619-92-3
3. Azar, A.T.: Overview of type-2 fuzzy logic systems. Int. J. Fuzzy Syst. Appl. (IJFSA) **2**(4), 1–28 (2012)
4. Azar, A.T.: Adaptive neuro-fuzzy systems. In: Azar, A.T. (ed.) Fuzzy Systems. IN-TECH, Vienna, ISBN 978-953-7619-92-3 (2010a)
5. Baer, P.G.D., Kapetanios, E., Keuser, S.: A Semantics Based Interactive Query Formulation Technique, UserInterfaces to Data Intensive Systems. Second International Workshop on User Interfaces to Data IntensiveSystems, Zurich, 43–49 (2001)
6. Bao-xiang, X., Zhang, Y.: Research on the development of information system modeling theory. J. Intell. **29**(5), 70–74 (2010)
7. Bechhofer, S., Horrocks, I., Goble, C., Stevens, R.: OilEd-a reason-able ontology editor for the semantic web. In: Joint German-Austrian Conference on Artificial Intelligence (KI01), Vienne, pp. 396–408 (2001)
8. Berners-Lee, T.: Weaving the Web. HarperCollins, New York, ISBN 006-251-5861 (1999)
9. Bobillo, F., Straccia, U.: Fuzzy description logics with general t-norms and datatypes. Fuzzy Sets Syst. **160**(23), 3382–3402 (2009)
10. Borzsonyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: International Conference on Data Engineering (ICDE), Heidelberg (2001)
11. Bosc, P., Galibourg, M., Hamon, G.: Fussy quering with SQL: extensions and implementation aspects. Fussy Sets Syst. **28**(3), 333–349 (1988)
12. Bosc, P., Liétard, L.: Aggregates computed over fuzzy sets and their integration into SQLf. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **16**(6), 761–792 (2008)
13. Bosc, P., Lietard, L., Pivert, O.: Databases and flexibility: gradual queries. TSI-Technique et Science Informati ques-RAIRO **17**(3), 355–378 (1998)
14. Bosc, P., Pivert, O.: SQLf: a relational database language for fuzzy querying. Comput. J. IEEE Trans. Fuzzy Syst. **3**(1), 1–17 (1995)

15. Bosc, P., Pivert, O.: A propos de requetes a preferences et diviseur stratifie. 28me Congrs INFORSID, France, pp. 311–326 (2010)
16. Carpineto, C., De Mori, R., Romano, G., Bigi, B.: An information theoretic approach to automatic query expansion. ACM Trans. Inf. Syst. **19**(1), 1–27 (2001)
17. Carroll, J., Klyne, G.: RDF concepts and abstract syntax. Recommendation, W3C (2004), http://w3c.org/TR/rdf-concepts
18. Chandrasekaran, B., Josephson, J., Benjamin, V.: What are ontologies, and why do we need them? IEEE Intell. Syst. **14**(1), 20–26 (1999)
19. Chang, C.: Decision support in an imperfect world. In: Trends and Applications on Automating Intelligent Behavior-Applications and Frontiers, Denmark, p. 25 (1983)
20. Chomicki, J.: Preference formulas in relational queries. ACM Trans. Database Syst. **28**(4), 427–466 (2003)
21. Chu, W., Yang, H., Minock, M., Chow, G., Larson, C.: CoBase-a scalable and extensible cooperative information system. J. Intell. Inf. Syst. **6**(2–3), 223–259 (1996)
22. Clerkin, P., Cunningham, P., Hayes, C.: Ontology discovery for the semantic web using hierarchical clustering. In: European Conference Machine Learning (ECML) and European Conference Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD-2001) (2001)
23. Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: The semantic web-the roles of XML and RDF. IEEE Internet Comput. **5**(4), 63–74 (2000)
24. Ding, Y., Foo, S.: Ontology research and development: Part 1 a review of ontology generation. J. Inf. Sci. **28**(2), 123–136 (2002)
25. Domingue, J., Motta, E.: Knowledge modeling in webonto and ocml. http://kmi.open.ac.uk/projects/ocml (1999)
26. Domshlak, C., Hoos, H., Boutilier, C., Brafman, R., Poole, D.: Cp-nets : a tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artif. Intell. Res. **21**, 135–191 (2004)
27. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Proceedings of the 5th International Conference on Flexible Query Answering Systems (FQAS) 02, London, UK, pp. 174–182 (2002)
28. Farquhar, A., Fikes, R., Rice, J.: The ontolingua server: a tool for collaborative ontology construction. In: The 10th Knowledge Aqcuisition for Knowledge-Based Systems (KAW96), Canada, pp. 174–182 (1996)
29. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical Foundations, vol. 1640. Springer, Heidelberg (1999)
30. Grissa Touzi, A., Sassi, M., Ounelli, H.: An innovative contribution to flexible query through the fusion of conceptual clustering, fuzzy logic, and formal concept analysis. Int. J. Comput. Appl. 16(4), 220–233 (2009)
31. Kapetanios, E., Baer, D., Glaus, B., Groenewoud, P.: MDDQL-Stat: data querying and analysis through integration of intentional and extensional semantics. Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004), Switzerland, pp. 353 (2004)
32. Kiessling, W.: Data querying and analysis through integration of intentional and extensional semantics. In: Foundations of Preferences in Database Systems. Very Large Data Base (VLDB) Endowment Inc, pp. 311–322 (2002)
33. Lacroix, M., Lavency, P.: Preferences-putting more knowledge into queries. In: Proceedings of the 13th International Conference on Very Large Data Bases, University of Vienna, Austria, pp. 217–225 (1987)
34. Lassila, O., Swick, R.: Resource description framework (RDF) model and syntax specification. Recommendation, W3C (1999)
35. Lietard, L., Rocacher, D.: On the definition of extended norms and co-norms to aggregate fuzzy bipolar conditions. In: The European Society of Fuzzy Logic and Technology Conference, pp. 513–518 (2009)

36. Mena, E., Illarramendi, A., Kashyap, V., Sheth, A.: OBSERVER: an approach for query processing in global information systems based on interoperation across pre-existing ontologies. J. Distrib. Parallel Databases **8**(2), 223–271 (2000)
37. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: OWL 2 web ontology language: profiles. Recommendation, W3C. http://www.w3.org/TR/owl2-profiles/ (2009)
38. Motro, A.: VAGUE-A user interface to relational databases that permits vague queries. ACM Trans. Office Inf. Syst. **6**(3), 187–214 (1988)
39. Noy, N., Fergerson, R., Musen, M., IENG, R. D., CORBY, O. The knowledge model of protg2000 : combining interoperability and flexibility. In: 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW00), Juan-les-Pins, France, pp. 17–32 (2000)
40. Ounalli, H., Belhadj, R.: Interrogation flexible et coopérative d'une BD par abstraction conceptuelle hirarchique, pp. 41–56. INFORSID, Biarritz, France (2004)
41. Paton, N.W., Stevens, R., Baker, P., Goble, C.A., Bechhofer, S., Brass, A.: Query Processing in the TAMBIS Bioinformatics Source Integration System. Proceedings of the IEEE International Conference on Scientific and Statistical Databases (SSDBM), 138–147 (1999)
42. Pivert, O.: Contribution a l'interrogation flexible de bases de donnees: expression et evaluation de requetes floues. PhD thesis (1991)
43. Quan Thanh, T., Hui, S.C., Fong, A., Cao, T.H.: Automatic fuzzy ontology generation for semantic web. IEEE Trans. Knowl. Data Eng. **18**(6), 842–856 (2006)
44. Rabitti, F., Savino, P.: Retrieval of multimedia documents by imprecise query specification. In: Advances in Database Technology-EDBT90, pp. 203–218. Springer, Berlin (1990)
45. Sassi, M., Grissa Touzi, A., Ounelli, H.: Clustering quality evaluation based on fuzzy FCA. In: 18th International Conference on Database and Expert Systems Applications, (DEXA07), Regensburg, Germany, pp. 62–72. LNCS (2007)
46. Soergel, D.: Some remarks on information languages, their analysis and comparison. Inf. Storage Retrieval **3**(4), 219–291 (1967)
47. Spoerri, A.: InfoCrystal: a visual tool for information retrieval management. In: Second International Conference on Information and Knowledge Management, Washington, pp. 11–20 (1993)
48. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: collaborative ontology engineering for the semantic web. In: First International Semantic Web Conference (ISWC02) of Lecture Notes in Computer Science, Chia, Sardaigne, Italie, vol. 2342, pp. 221–235 (2002)
49. Tahani, V.: A conceptual framework for fuzzy query processing: a step toward very intelligent database systems. Inf. Process. Manage. **13**(5), 289–303 (1977)
50. Tran, T., Wang, H., Rudolph, S., Cimiano, P.: Top-k exploration of query graph candidates for efficient keyword search on rdf. In: IEEE Computer Society (ed.) Proceedings of the 2009 IEEE International Conference on Data Engineering, pp. 405–416. IEEE Computer Society (2009)
51. Uri, K., Jianjun, Z.: Fuzzy clustering principles, methods and examples, vol. 17(3), p. 13. Technical Report, Technical University of Denmark, IKS, Denmark, (1998)
52. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, vol. 83. Springer, Berlin (1982)
53. Wuermli, O., Wrobel, A.C. a. H. & Joller, J. (2003). Data mining for ontology building: semantic web overview. PhD thesis, Nanyang Technological University
54. Zadeh, L.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)