

Individual Classification: an Ontological Fuzzy Based Approach

Asma Djellal

Preparatory School of Economics, Business and Management Sciences of Constantine, Algeria

LIRE Laboratory, Constantine 2 - Abdelhamid Mehri -University, Constantine, Algeria

E-mail: asmadjellal@gmail.com

Zizette Boufaida

LIRE Laboratory, Constantine 2 - Abdelhamid Mehri -University, Constantine, Algeria

E-mail: zizette.boufaida@univ-constantine2.dz

Keywords: fuzzy logic, fuzzy ontology, classification reasoning, individual classification, fuzzy ontologies realization

Received: July 25, 2016

Recently, several reasoners for very expressive fuzzy Description Logics have been implemented. However, in some cases, applications do not require all the reasoner services and would benefit from the efficiency of just certain reasoning tasks. To this scope, we are interested in the individual fuzzy classification issue. In fact, decision-making applications for real world domain is often based on classifying new situations into fuzzy categories. Therefore, we propose Fuzzy Realizer to offer an effective classification even with imprecise/vague or incomplete knowledge so that appropriate decision can be made. Fuzzy Realizer is a Java prototype implementation for realizing fuzzy ontologies. It supports the well-known fuzzy description logic $\mathcal{ZSHOIN}(\mathcal{D})$. It allows (i) fuzzy concrete domains, (ii) modified and (iii) weighted concepts. It is able to (i) classify new individuals, even with incomplete descriptions, (ii) provide a more human-oriented classification by hiding the crisp boundaries between different fuzzy categories and (iii) to populate fuzzy ontologies which address an aspect of fuzzy ontologies evolution, a topic which is rarely discussed.

Povzetek: Razvit je postopek za individualno klasifikacijo s pomočjo mehke logike.

1 Introduction

Crisp ontologies, based on first-order logic formalisms, are not suitable for handling imperfect knowledge. Knowledge imperfection, manifested by incomplete, vague or imprecise notions, is inherent to several real-world domains, and this problem has therefore attracted the attention of many research communities [21, 22, 26, 28, 29]. Several approaches have incorporated fuzzy logic into ontology languages and description logics (DLs) to build so-called fuzzy ontologies. Indeed, a number of reasoners for very expressive fuzzy DLs have been implemented [31], including FiRE [25], FuzzyDL [3, 6] and DeLorean [2]. Moreover, a number of optimization techniques have been proposed recently for improving reasoning efficiency for very expressive fuzzy DLs [5, 24]. However, in some cases, applications do not require all the reasoner services and would benefit from the efficiency of just certain reasoning tasks. To this scope, we have been interested in the fuzzy ontologies realization issue.

Realizing fuzzy ontologies with new individuals is a very important reasoning task. Using this reasoning task, several real world domains can benefit from affective decision-making applications. Indeed, in a domain like e-health, doctors always classify their patients into fuzzy categories. When referring to a patient's fever, for example, if we have a body temperature of 38.5° , it will be stated that the patient has a "high" fever. However, a temperature of 38° will present a "high" fever, but also it can be stated that it is an "average" fever. A similar

classification can be used in industry where Industrial Process Control Systems collect data, such as temperature and pressure of gas and oil pipes, for example, to be classified as safe situations or not. Based on this classification appropriate decisions can be made.

Classification is the main reasoning mechanism for systems based on class/instance models. It is one of the most powerful and fundamental human inference mechanisms. It maintains the stability of the knowledge base in the presence of new knowledge, by connecting each knowledge to its class. However, since we are handling imperfect knowledge, giving exact definitions of class boundaries seems to be a very difficult, perhaps even impossible, task. Therefore, we have integrated fuzzy logic with classification to enable the attachment of an individual to several fuzzy classes. Such attachment makes the sharp borders between classes disappear, which better reflects reality and allows a more human-oriented modelling process.

Having these ideas in mind, we propose a fuzzy-based approach for realizing fuzzy ontologies by classifying new individuals and connecting them to their most specialized concepts. Based on this classification operators may take the appropriate decisions. With our approach, two features of knowledge imperfection can be handled: vagueness/imprecision and incompleteness. Indeed, based on a fuzzy classification algorithm, the proposed reasoning service can classify new individuals, even with incomplete description. To validate our ideas, we have

implemented this algorithm in what we call *Fuzzy Realizer*. It is a Java prototype implementation supporting the fuzzy DL *SHOIN(D)* under Zadeh semantics (*Z.SHOIN(D)*). It allows (i) fuzzy concrete domains, (ii) modified and (iii) weighted concepts.

The underlying key of *Fuzzy Realizer* is that (i) it can classify new individuals, even though we may lack information about them, (ii) it provides a more human-oriented classification process by assigning an individual to several fuzzy concepts with different membership degrees. Finally, (iii) it can populate fuzzy ontologies which address an aspect of fuzzy ontologies evolution, a topic which is rarely discussed. Indeed, ever since the development of ontologies, especially from large text corpora, became a well-understood problem [23], reconstruction is always preferred to an evolutionary process. In fact, the evolution problem is challenging [33] and need to be analysed from different point of views, thus, the present paper addresses the individual classification issue by providing a realization service for fuzzy ontologies.

The remainder of this paper is organized as follows. Section 2 presents some preliminaries that will be used in the rest of the paper, namely, fuzzy logic and classification reasoning mechanism. Section 3 reviews some related works and situates our work in that context. Section 4 discusses the proposed fuzzy realization algorithm then, an extension of this approach, namely a fuzzy relocation process will be presented in Section 5. To validate our ideas, we present in Section 6 *Fuzzy Realizer*. Finally, Section 7 concludes the paper with ideas for future research.

2 Preliminaries

This section describes some background material regarding (i) fuzzy logic and its use for representing imperfect knowledge, and (ii) the classification reasoning mechanism which enables their classification.

2.1 Fuzzy logic and fuzzy ontology

Fuzzy logic was designed to solve the problem of vague/fuzzy and imprecise knowledge representation. It was introduced by L. A. Zadeh in the mid-1960s as an extension of Boolean logic [34]. In classical set theory, there are two possibilities: elements either belong to a set or they do not. This theory does not consider many situations that are frequently encountered in everyday life, where imprecision is manifested by terms like *high*, *young*, *hot* and the like. Fuzzy logic, based on fuzzy set theory, is designed to consider this kind of situation. It is based on the notion of *partial membership*, where each element belongs partially or gradually to defined fuzzy subsets.

Definition. Let X be a set of elements. A fuzzy subset A of X is defined by a function called the membership function and is denoted as $\mu_A(x)$. $\mu_A(x)$ is a mapping which takes any value from the real interval $[0, 1]$:

$$\mu_A(x): x \rightarrow [0, 1], x \in A$$

The crisp set operators negation, intersection and union are extended to fuzzy subsets and performed by fuzzy negation, t-norm and s-norm functions, respectively, so that one can form different fuzzy logics. The most widely used one is Zadeh fuzzy logic, known as Zadeh Semantics [4]. It is a combination of Gödel conjunction (t_G) and disjunction (S_G) ($t_G = \min(a, b)$ and $S_G = \max(a, b)$) and Łukasiewicz negation (N_L) ($N_L = 1 - a$).

Fuzzy calculus is a vast and very flexible research field; indeed, it is used in many domains, one of them is fuzzy ontologies development [5, 1, 14]. Fuzzy ontologies extend crisp ones by interpreting concepts and roles as fuzzy sets of individuals and binary relations respectively. Unlike crisp ontologies which allow an element to be described or not, $\{0, 1\}$, by each concept in the ontology, fuzzy ontologies associate an element to each concept using a membership degree in the interval $[0, 1]$. Such association allows the attachment of each element to different concepts with different membership degrees. Consequently, fuzzy ontologies have a more flexible representation capability than crisp ones. In fact, vague notions, manifested by fuzzy terms like *high_temperature*, *very_close_to* and the like, are quite common in human language, and they can be represented by means of fuzzy ontologies elements using different constructs [29]; the most important of these are:

Explicit fuzzy concepts. Represented by means of fuzzy membership functions using fuzzy concrete domains such *High_temperature* which is a fuzzy concept defined with the fuzzy concrete domain *High* with its Right-Shoulder membership function, *High* (37, 38.5) as:

$$\text{High_temperature} \equiv \text{temperature} \sqcap \exists \text{Degree.High}$$

Modified concepts. Fuzzy modifiers, such as *very* or *slightly*, are defined by functions $f_m: [0, 1] \rightarrow [0, 1]$, applied to change membership functions. For instance, *Very_high_temperature* is a fuzzy modified concept defined with *Very* as a fuzzy modifier having the function $f_{\text{Very}}(x) = x^2$ as:

$$\text{Very_high_temperature} \equiv \text{temperature} \sqcap \exists \text{Degree.Very (High)}$$

Weighted concepts. Sometimes we want to express the importance of concepts representing preferences or priorities, such as 0.8 (C). These concepts, called fuzzy weighted concepts, are defined as follows:

$$D \equiv w(C) / w \in [0, 1]$$

For the rest of the paper, m and f_m are used to represent fuzzy modifiers and their membership functions, while w ($w \in [0, 1]$) is used to express weights of concepts.

In this section, we have provided some preliminaries regarding fuzzy ontologies by introducing the basic concepts which are involved. For a more in-depth presentation, we refer the reader to [30].

2.2 Classification reasoning mechanism

Classification is the fundamental inference mechanism for object-based representations. Indeed, structuring knowledge into classes, subclasses and instances promotes the use of classification to retrieve implicit knowledge. To this end, classification can be used to (i) categorize a set of objects into category graphs, (ii) add a new category to

an already created graph or (iii) to add a new object to its most specialized categories in the created graph [18]. This process, also called individual classification, refers to ontology realization. It is used to retain the stability of an already created knowledge base in the presence of a new individual by connecting it to the most specialized concepts it belongs to (see, Figure 1).

Classification of individuals consists of precisely selecting their belonging classes. Therefore, different classes have to be well separated. However, giving exact definitions of class boundaries is a very difficult, perhaps even impossible, task. The difficulty comes from the vagueness of the modelled knowledge. To address this problem, we have integrated fuzzy logic with classification to enable the use of non-numerical values which allow non-sharp definitions of class boundaries. Fuzzy classification [16, 32] is the process of grouping elements into fuzzy sets. The membership of these elements to each fuzzy set is not full but partial to some degree. The main difference between crisp and fuzzy classification is that in fuzzy classification, an element can belong to several fuzzy classes with different membership degrees. Such membership makes the sharp borders between classes disappear, which better reflects reality and allows a more human-oriented modelling process.

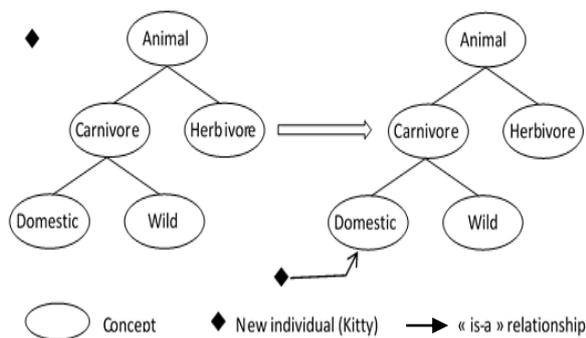


Figure 1: An individual classification example.

3 Related work

Work related to our research context explores two research fields: (i) handling imperfect knowledge and (ii) classification reasoning mechanisms.

3.1 Handling Imperfect knowledge

It has been widely pointed out that crisp ontologies are not suitable to handle imperfect knowledge. Thus, many fuzzy approaches have been proposed to cover this limitation [1, 3, 4, 7, 13, 27]. As a result, a few methodology for developing fuzzy ontologies have been proposed and a number of fuzzy extensions of DL have been used. However, like crisp ontologies, the success of fuzzy ones depends on the availability of effective software allowing their exploitation. Consequently, the reasoning task has been a very interesting topic for many researchers. DeLorean (DEscription LOGic REasoner with vAgueNess) [2] was the first reasoner that supported a fuzzy extension of the DL *SRQIQ*. As far as we know,

DeLorean is the only reasoner that supports fuzzy OWL2. Based on Zadeh Semantics, it represents fuzzy operators and reduces the resulting fuzzy *SRQIQ* knowledge base to a crisp one by creating new crisp concepts and roles representing α -cuts [20] of original fuzzy ones. Other quite similar studies have proposed reasoners for expressive fuzzy DLs. For instance, Fire implements a tableau algorithm for fuzzy *SHIN* restricted to Zadeh Semantics [25]. YADLR is a Prolog implementation based on linear programming [17]. It supports a fuzzy extension of *ALCOQ* under Łukasiewicz and Zadeh fuzzy logics and allows variables as degrees of truth. In order to benefit from the full expressivity of a less expressive language and then guarantee the reasoning efficiency, LiFR was proposed [31]. It is a lightweight fuzzy reasoner oriented to mobile devices and the supported language is f-DLP. It allows fuzzy concept assertions and weighted concepts. *FuzzyDL* [3, 6] is an important fuzzy reasoner supporting fuzzy extensions of *SHIF(D)* under Zadeh, Łukasiewicz and classical semantics. It was successfully used in some practical applications. Its interesting features are aggregation of fuzzy concepts, explicit fuzzy set membership functions and fuzzy modifiers.

Like all these cited works, we were interested in reasoning with imperfect knowledge using fuzzy logic. However, unlike them, we have been interested in just one reasoning task to propose a fuzzy ontologies realization service as much complete and efficient as possible. As far as we know, no other work exploits the fuzzy classification mechanism with fuzzy ontologies especially with incomplete individuals. On the other hand, there have been some previous attempts to combine this reasoning mechanism with fuzzy logic in other research fields, such as pattern recognition and data mining.

3.2 Classification reasoning mechanism

fCQL (Fuzzy Classification Query Language) is a toolkit for classification, analysis and decision support applied in the marketing domain of a telecom company [19, 32]. Meier et al. claimed that ‘Using linguistic terms and variables hides the complexity of the domain and permits a more intuitive and human-oriented querying process in different application domains’ [19, pp. 586-587]. Therefore, they exploited the advantages of fuzzy logic to reduce the business data complexity and extracts valuable hidden information through fuzzy classification. *fCQL* allows formulating fuzzy queries which are then transformed into SQL statements. This approach benefits from fuzzy logic in classification and querying. However, its main disadvantage is that it is a data oriented approach, thus semantic retrieval of resources is not supported.

A closer approach to ours is [12], which defines a semi-automated musical genre classification mechanism using an ontological representation. Fuzzy classification was used to allow the classification of music resources into musical genres based on a score provided by the resource composer expressing its viewpoint. Indeed, in music classification, different users are not required to agree about the classification of a specific music resource in the same musical genre. In this approach, fuzzy classification

is flexible regarding the different interpretations of music genres. However, the consideration of vagueness is quite limited because (i) music resources are represented by crisp ontologies, (ii) fuzzy logic is restricted to express users' viewpoints and (iii) the membership degree is not calculated based on membership functions but is instead given by the user. Finally, in this approach, (iv) knowledge imperfection was considered without reference to knowledge incompleteness, which is an important feature of fuzzy knowledge.

4 A fuzzy realization algorithm

Using an illustrative example, we will study and improve upon a fuzzy realization algorithm proposed in a previous work [9]. The following algorithm has been extended and improved in order to accelerate the classification process.

Algorithm 1. Fuzzy realization algorithm

Input: H: Fuzzy concepts hierarchy (Fuzzy Ontology)

A: New individual

Output: Evolved fuzzy ontology

1. Initialization ();
2. $C^* := \text{TOP}(H)$;
3. **While** (not empty (C^*)) **do**
4. Matching (C^* , A);
5. Marks-Propagation (C^* , label, degree) ;
6. $C^* := \text{Next-Concept}(C^*)$;
7. **End while**

Table 1: Fuzzy realization algorithm

The proposed algorithm allows the realization of fuzzy ontologies and results to evolved ones in which the new individual A will be attached to its most specialized concepts. First, the user provides the necessary knowledge (line 1) to start the classification loop. This loop consists of exploring the hierarchy and matching the current concept C^* with A (line 4), starting at the hierarchy root TOP (line 2). The Matching procedure verifies A's membership in C^* and, if A belongs to C^* , the concept will be marked with a label and a membership degree. To accelerate the classification, the Marks_Propagation procedure (line 5) propagates marks to different concepts related to C^* based on some logical rules. The next concept to be matched with A is chosen by the Next_Concept function (line 6). If there are no more unmarked concepts, Next_Concept returns *null* which terminate the classification.

Illustrative example. In the following sections, we will study the proposed algorithm for the following illustrative example; it is an excerpt from a simple fuzzy knowledge base about persons:

TBox

- [Ax 1] $\text{Person} \sqsubseteq \top$ [Ax 2] $\text{Male} \sqsubseteq \top$
 [Ax 3] $\text{Female} \sqsubseteq \top$ [Ax 4] $\text{Male} \equiv \neg \text{Female}$
 [Ax 5] $\text{Man} \equiv \text{Person} \sqcap \text{Male}$
 [Ax 6] $\text{Woman} \equiv \text{Person} \sqcap \text{Female}$
 [Ax 7] $\text{Young} \equiv \text{Person} \sqcap \exists \text{HasAge. YoungAge}$
 [Ax 8] $\text{Adult} \equiv \text{Person} \sqcap \exists \text{HasAge. AdultAge}$

- [Ax 9] $\text{Teacher} \equiv \text{Adult} \sqcap \exists \text{HasFunction. Teacher}$
 [Ax 10] $\text{Very Young} \equiv \text{Person} \sqcap \exists \text{HasAge. very (YoungAge)}$
 [FCP 1] $\text{YoungAge}(x) = \text{Left-shoulder}(10, 30)$
 [FCP 2] $\text{AdultAge}(x) = \text{Trapezoidal}(30, 35, 50, 60)$
 [FCP 3] $\text{Very}(x) = x^2$

ABox

- [FCA 1] $\langle \text{Tom: Person} = 1 \rangle$ [FCA 2] $\langle \text{Tom: Male} = 1 \rangle$
 [FCA 3] $\langle \text{Lina: Person} = 1 \rangle$ [FCA 4] $\langle \text{Lina: Female} = 1 \rangle$

Person, Male and Female are defined as atomic concepts. Axioms [Ax 5] and [Ax 6] define crisp concepts, while [Ax 7]–[Ax 10] describe some fuzzy ones. [FCP 1] and [FCP 2] concern fuzzy concrete predicates YoungAge and AdultAge; they indicate the degree to which a person is young or adult, respectively, using left-shoulder and trapezoidal membership functions. [FCP 3] defines the fuzzy modifier Very. Finally, the ABox contains some fuzzy concept assertions to define two individuals: Tom and Lina.

4.1 Initialization () procedure

To start the classification loop, we need to collect some information about the new individual in the form of (attribute, value) pairs. The user must provide as much knowledge as possible so that the algorithm can classify the individual as precisely as possible in the hierarchy. If the user do not have enough information, the Initialization procedure accepts the value 'Unknown'. Consequently, the proposed algorithm can classify incomplete individuals.

Definition. Let A be an individual defined by its description in the form of a set of (attribute, value) pairs. If we are missing information about an attribute of A, then it is *incomplete*. Formally:

A is incomplete $\Rightarrow A = \{(\text{Att}_1, \text{Val}_1) \dots (\text{Att}_n, \text{Val}_n)\}$ and $\exists i / (\text{Att}_i, \text{Unknown}) \in A$.

Example 1. Consider our ABox, having the individual Tom with its description: $\text{Tom} = \{(\text{Name}, \text{Tom}), (\text{Age}, 33), (\text{Size}, 1.7), (\text{Function}, \text{Unknown}) \dots\}$. Since we are missing information about the attribute "Function", Tom is incomplete.

4.2 Matching (C^* , A) procedure

Matching is the algorithm key procedure. It has the role of checking an individual's membership in the current concept based on a membership function. Classical, two-valued, membership function has been successfully applied to consider complete and precise knowledge, for which we can exactly define their belonging classes. However, it seems to be inappropriate to be used for managing fuzzy knowledge bases, in which we handle imprecise and incomplete knowledge. To cover this limitation, we have chosen the membership function with three values. The scope of this function is extended to accept the value *possible*, if we do not have sufficient information for affirming or denying an instance's membership in a given class. This function can be described as follows, given that x is an instance and C is a class with the membership function $C(x)$:

$$C(x) = \begin{cases} \text{sure} & \text{if } x \in C \\ \text{impossible} & \text{if } x \notin C \\ \text{possible} & \text{otherwise} \end{cases}$$

Based on this function, Matching procedure marks the current concept C^* with a label, indicating whether it is sure, possible or impossible for the new individual (see Figure 2). Since there is no full membership in fuzzy ontologies, C^* will be marked with another mark expressing the degree of this membership. In sum, Matching procedure generates the following output: $\langle C^*, \text{label}, \text{degree} \rangle$ where $\text{label} \in \{S, P, I\}$ and $\text{degree} \in [0, 1]$, if the new individual belongs to C^* (that is, $\text{label} = \text{sure}$), or null if there is no membership. For the rest of the paper, S, P and I will be used to represent, respectively, the marks *Sure, Possible and Impossible*:

- $\langle C^*, S, d \rangle$ (A is C^* with a truth-value of d): if A 's value for each attribute satisfies the constraints of C^* . This membership can be determined only if A is complete.
- $\langle C^*, I, \text{null} \rangle$ (A is not C^*): if A 's value for at least one attribute does not satisfy the constraints in C^* . In this case, we do not consider whether A is incomplete.
- $\langle C^*, P, \text{null} \rangle$ (A may be C^*): if A is incomplete and its values do not stand in contradiction with C^* .

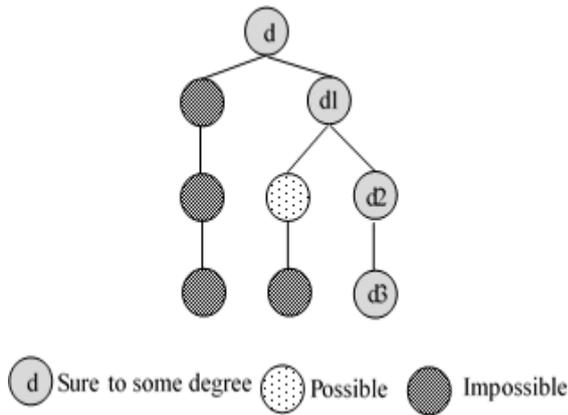


Figure 2: Fuzzy classification of an individual based on concept marking.

Table 2: Matching $\langle C^*, A \rangle$ procedure.

If C^* includes some attributes that are not defined in the description of A , then Matching asks the user for values for these attributes (lines 2–4). Using the function Get_degree , the membership of A in C^* is computed (line 5). If there is no membership ($\text{Degree} = 0$), the matching stops and C^* will be marked Impossible (lines 12–14). If all constraints are satisfied (that is, $\text{Degree} > 0$), two cases are considered:

- A is incomplete: the matching stops and C^* will be marked Possible (lines 6–8).
- A is complete: C^* will be marked Sure to some 'Degree' (lines 9–11).

In order to mark C^* , the function Get_degree calculates A 's degree of membership in C^* . C^* can be described based on several logical expressions: concept conjunction,

Procedure 1. Matching $\langle C^, A \rangle$*

Input: C^* : Current concept
 $A = \{(\text{Att}_1, \text{Val}_1), \dots, (\text{Att}_n, \text{Val}_n)\}$: new individual
Output: $\langle C^*, \text{label}, \text{degree} \rangle$

```

Degree : real;
1  Begin
2  If (  $\exists \text{Val}_i = ""$  ) then
3    Request the user;
4  End if
5  Degree := Get_degree (  $C^*$ ,  $A$  );
6  If ( Degree > 0 ) then
7    If (  $\exists \text{Val}_i = " \text{Unknown} "$  ) then
8      Mark (  $C^*$ ,  $P$ , null );
9    Else
10     Mark (  $C^*$ ,  $S$ , Degree );
11   End if
12 Else
13   Mark (  $C^*$ ,  $I$ , null );
14 End if
15 End.
    
```

modified concept, explicit fuzzy concept etc. Based on the description of C^* and under Zadeh Semantics, the Get_degree function proceeds according to the following cases:

- *Concept conjunction* $C^* \equiv C_1 \sqcap \dots \sqcap C_n$:
 $\text{Degree} (C^*, A) = \min (\text{Degree} (C_i, A)) / i=1..n$.
- *Concept disjunction* $C^* \equiv C_1 \sqcup \dots \sqcup C_n$:
 $\text{Degree} (C^*, A) = \max (\text{Degree} (C_i, A)) / i=1..n$.
- *Concept negation* $C^* \equiv \neg C$:
 $\text{Degree} (C^*, A) = 1 - \text{Degree} (C, A)$
- *Fuzzy modified concept* $C^* \equiv m(C)$:
 $\text{Degree} (C^*, A) = f_m (\text{Degree}(C, A))$.
- *Fuzzy weighted concept* $C^* \equiv w (C)$:
 $\text{Degree} (C^*, A) = w * \text{Degree}(C, A)$
- *Explicit fuzzy concept* $C^* \equiv \exists \text{Attribute.Range}$, where Range is a fuzzy predicate:
 $\text{Degree} (C^*, A) = f_{\text{Range}}(A.\text{Attribute})$.
 eg. $\exists \text{Age.YoungAge}$, results to $f_{\text{Young}}(A.\text{Age})$
- *Limited existential quantification*: $C^* \equiv \exists R.C$. In this case, the function returns the maximum degree of membership of all individuals (A_i) related by the role R in the concept C .
 $\text{Degree}(C^*, A) = \max (\text{Degree}(C, A_i))$.
- *Value restriction*: $C^* \equiv \forall R.C$. The function returns the minimum degree of membership of all individuals (A_i) related by the role R in the concept C :
 $\text{Degree} (C^*, A) = \min (\text{Degree}(C, A_i))$.
- *Max cardinality* $C^* \equiv \geq n R.C$:
 If $|\text{Degree} (C, A_i) > 0| \geq n$ then $\text{Degree} (C^*, A) = 1$
 else 0.
- *Min cardinality* $C^* \equiv \leq n R.C$:
 If $|\text{Degree} (C, A_i) > 0| \leq n$ then $\text{Degree} (C^*, A) = 1$
 else 0.

Example 2. Assuming that the individual Lina in our earlier illustrative example is a 12 years old girl. Since *Person* is already marked *sure* ($\langle \text{Person}, S, 1 \rangle$). Based on $[\text{Ax } 7]$ and $[\text{CPF } 1]$, $\text{Get_degree} (\text{Young}, \text{Lina})$ returns 0.9,

and thus, the fuzzy concept *Young* will be marked $\langle \text{Young}, S, 0.9 \rangle$.

Example 3. Consider the individual Tom in Example 1. Based on its description, [Ax 8] and [FCP 2] and $\text{Get_degree}(\text{Adult}, \text{Tom}) = 0.6$, Matching will mark this concept as $\langle \text{Adult}, S, 0.6 \rangle$. Moreover, based on [Ax 9], *Teacher* will be marked as $\langle \text{Teacher}, P, \text{null} \rangle$.

4.3 Marks-propagation (C^* , label, degree) procedure

In order to accelerate the classification process, Marks-propagation minimizes the number of concepts to be verified by Matching procedure. It is a recursive procedure that propagates marks to concepts related to C^* , and to their related concepts. This procedure propagates marks based on the mark of C^* according to certain rules and under Zadeh Semantics. For instance, according to [R.1], all synonymous of C^* will be marked sure to some degree (d). Then, each of these synonymous (D) will be the new input of Marks-propagation, and then this procedure starts to propagate marks to all concepts related to it (D), and so on until there will be no rule to be applied or no unmarked related concept to be marked.

[R. 1] If $\langle C^*, S, d \rangle$, then $\forall D, C^* \equiv D \langle D, S, d \rangle$.

[R. 2] If $\langle C^*, I, \text{null} \rangle$, then $\forall D, C^* \equiv D \langle D, I, \text{null} \rangle$.

[R. 3] If $\langle C^*, S, d \rangle$, $\forall D, C^* \equiv \bigcap D \langle D, I, \text{null} \rangle$.

[R. 4] If $\langle C^*, I, \text{null} \rangle$, then $\forall D, C^* \equiv \bigcap D \langle D, S, d \rangle / d = \text{Get_degree}(D, A)$.

In this case, we can confirm the membership of A in D. However, the degree of this membership must be computed by $\text{Get_degree}(D, A)$.

[R. 5] If $\langle C^*, S, d \rangle$, then $\forall D, C^* \sqsubseteq D, \langle D, S, \geq d \rangle$.

[R. 6] If $\langle C^*, I, \text{null} \rangle$, $\forall D, D \sqsubseteq C^*, \langle D, I, \text{null} \rangle$.

[R. 7] If $\langle C^*, P, \text{null} \rangle$, then $\forall D, D \sqsubseteq C^*, \langle D, \text{label}, \text{null} \rangle / \text{label} \notin \{S\}$.

This rule can be used to check some consistency problems. Indeed, if C^* is possible for A, then A is incomplete for C^* and for all of its specific concepts.

[R. 8] If $\langle C^*, S, d \rangle$, then $\forall D, D \equiv m(C^*), \langle D, S, f_m(d) \rangle$.

[R. 9] If $\langle C^*, S, d \rangle$, $\forall D, D \equiv w(C^*)$, then $\langle D, S, w^*d \rangle$.

Supposition1. D is defined by a concept conjunction including C^* as $D \equiv C^* \sqcap C_1 \sqcap \dots \sqcap C_n$.

[R. 10] If $\langle C^*, I, \text{null} \rangle$, then $\langle D, I, \text{null} \rangle$.

[R. 11] If $\langle C_i, S, d_i \rangle$, and $\langle C^*, S, d \rangle$, then $\langle D, S, \text{deg} \rangle / \text{deg} = \min(d, d_i) / i=1 \dots n$.

[R. 12] If $\langle D, I, \text{null} \rangle$, $\langle C^*, S, d_i \rangle$ and $\exists j \in \{1 \dots n\} \langle C_j, \text{""}, \text{""} \rangle$ (which means that C_j is unmarked), $\forall i \in \{1 \dots n\} / i \neq j \langle C_i, S, d_i \rangle$, then $\langle C_j, I, \text{null} \rangle$.

Supposition2. D is defined as a concept disjunction including C^* as $D \equiv C^* \sqcup C_1 \sqcup \dots \sqcup C_n$.

[R. 13] If $\langle C^*, S, d \rangle$, then $\langle D, S, \text{deg} \rangle / \text{deg} = \max(d, d_i) / i = 1 \dots n$.

[R. 14] If $\langle C_i, I, \text{null} \rangle / i = 1 \dots n$ and $\langle C^*, I, \text{null} \rangle$, then $\langle D, I, \text{null} \rangle$.

[R. 15] If $\langle D, S, d \rangle$, $\langle C^*, I, \text{null} \rangle$ and $\exists j \in \{1 \dots n\} \langle C_j, \text{""}, \text{""} \rangle, \forall i \in \{1 \dots n\} / i \neq j \langle C_i, I, \text{null} \rangle$, then $\langle C_j, S, d \rangle$.

Example 4. Recall the individual Tom from our illustrative example. Since *Person* and *Male* are already marked *Sure* ([FCA 1] and [FCA 2]), based on [Ax 5] and applying [R. 11], Marks-Propagation can propagate the mark *Sure* to the concept *Man* as: $\langle \text{Man}, S, 1 \rangle$. It can also propagate the mark *Impossible* to *Female* based on [Ax 4] and applying [R. 3]. Moreover, *Woman* will be *impossible* for Tom based on [Ax 6] and applying [R. 10].

Example 5. During the classification of Lina in Example 2, we have generated the result $\langle \text{Young}, S, 0.9 \rangle$. Thus, based on [Ax 11] and using [R. 8], Marks-Propagation can mark the modified concept *VeryYoung* as $\langle \text{VeryYoung}, S, 0.81 \rangle$. Consider same concepts and the individual Tom. If Matching (*Young*, Tom) results to $\langle \text{Young}, I, \text{null} \rangle$, then the same mark will be propagated to *VeryYoung* as $\langle \text{VeryYoung}, I, \text{null} \rangle$.

4.4 Next-Concept (C^*) function

The aim of this function is to select a new unmarked concept to be the next current concept, by traversing the hierarchy of fuzzy concepts. We use the breadth-first search traversal, which is one of the important graph traversal techniques, to explore the hierarchy graph. Using this technique, Next-Concept selects the next unmarked neighbouring concept of C^* . After testing all the unmarked neighbours, the function moves to the next level of the hierarchy and goes from left to right to select a new target concept. If there are no more unmarked concepts, Next_Concept returns null.

In our work, we were inspired by the multi-viewpoints classification algorithm proposed in [18], in which classification was used in an object-oriented multi-viewpoints representation system named TROPES. This algorithm provides multi-viewpoints instance classifications in which an instance can be classified in one or more viewpoints. This work was extended to consider individuals reclassification in multi-viewpoints ontologies [11]. These multi-viewpoints classification algorithms [11, 18] are both based on the hypothesis of the exclusiveness of sister classes, which assumes that classes at the same hierarchy level (called sister classes) represent mutually exclusive sets. Therefore, an individual which belongs to a class cannot belong to any of its sister classes. Unlike the cited works, our algorithm is not based on this hypothesis. Indeed, in our fuzzy ontology conceptualization, fuzzy concepts are modelled as fuzzy subsets [8]. The strength of fuzzy logic in knowledge representation lies in the intersections between fuzzy subsets, as an element can belong to several fuzzy subsets with different membership degrees. Consequently, the main advantage of fuzzy classification compared to classical one is that an element is not limited to a single class but can be assigned to several sister classes which better reflect reality.

Example 6. Consider the two fuzzy concepts *Child* and *Teenager*, defined by their trapezoidal membership functions (See Figure 3). Having the little girl Lina of Example 2, we can calculate these memberships: $Child(Lina) = 0.66$ and $Teenager(Lina) = 0.33$. These results dedicate that Lina is considered a Child but also a Teenager, with different membership degrees.

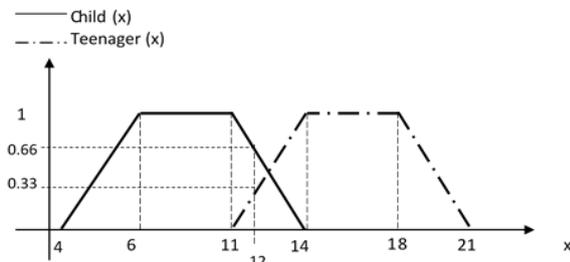


Figure 3: Assignment of an individual to different fuzzy concepts.

5 Individual relocation: an extension of the fuzzy realization approach

The proposed algorithm provides a complete and efficient realization service for fuzzy ontologies. Indeed, it can efficiently classify individuals, even incomplete ones, in their appropriate belonging concepts with their membership degrees. With this, we can ensure an evolutionary aspect of fuzzy ontologies by realizing them with new individuals.

After their classification, individuals may evolve and update their knowledge. Indeed, a person changes age, address or professions. Therefore, a relocation process is necessary to evolve fuzzy ontologies. To this scope, our proposed algorithm is extensible. In fact, an extension process of the fuzzy realization algorithm may consider another aspect of fuzzy ontologies evolution in which already classified, but updated, individuals can be relocated. This process allows an individual to migrate from its current belonging concepts to new ones that satisfy its updated description [10]. Changes of an individual description may be the result of an:

- *Enrichment* of an incomplete individual by replacing its unknown value by a concrete one,
- *Modification* of a concrete value by a new one, or
- *Impoverishment* and removal of a concrete value and replacing it by an unknown one.

In the first two cases, we have to handle a new data. This data can satisfy the fuzzy ontology constraints, and then, results to a consistent fuzzy ontology. It can also be in contradiction with some constraints and then generates an inconsistency:

Fuzzy Ontology in a consistent state. In this case, the individual belonging concepts must keep their marks as $\langle Ci^*, S, di \rangle$. However, the individual new description may allow it to migrate to concepts that are more specific. Thus, for this first case, a simple realization process is

revived to descent the evolved individual in the hierarchy starting at its belonging concepts.

Fuzzy Ontology in inconsistent state. To deal with this inconsistency, a fuzzy relocation process is invoked to migrate the updated individual to its new belonging concepts. To this end, the individual is raised up in the fuzzy hierarchy, by following the path of its sure super-concepts until the first super-concept for which the new data satisfies its constraints. It should be noted that all super-concepts along the individual path (excepting the last one) must change their marks from $\langle Ci^*, S, di \rangle$ to $\langle Ci^*, I, null \rangle$. To complete the individual relocation, the updated individual must descent in fuzzy hierarchy until it reaches its new belonging concepts. Indeed, the individual updated description can satisfy other concepts that are more specific than the first consistent super-concept. Thus, starting at this concept the fuzzy realization process is evoked.

The individual knowledge can evolve to an unknown value. This impoverishment will not affect the ontology consistency. However, the evolved individual becomes incomplete since the concrete value of the updated attribute has been changed with an unknown one. To handle this change, the fuzzy ontology must evolve and the updated individual must raise up, by following the path of its super-concepts until the first sure super-concept in which there is no specification for the impoverished attribute. All these super-concepts (excepting the last one) must change their marks from $\langle Ci^*, S, di \rangle$ to $\langle Ci^*, P, null \rangle$. Unlike the enrichment/modification, in the case of an individual impoverishment, once the ascent to the first sure super-concept is done no further descent is possible. Indeed, there is no concrete new data to be matched with more specific concepts.

6 Validation of the proposed algorithm

In order to validate our ideas, we have implemented¹ the proposed fuzzy realization algorithm as *Fuzzy Realizer*. It is a Java prototype implementation that supports a fuzzy extension of the well-known DL \mathcal{Z} *SHOIN* (\mathcal{D}). *Fuzzy Realizer* has a graphical interface for displaying the fuzzy ontology in the form of a coloured directed acyclic graph (DAG), in order to improve the results presentation and thereby facilitate the decision-making process (see Figure 4). *Fuzzy Realizer* has a modular architecture and is divided into three modules: the Parser, Visualization and Classification modules. The Parser translates the fuzzy ontology into an internal format, so that any fuzzy ontology encoded in any language (OWL, Fuzzy OWL, OWL 2, ...) can be used. The Visualization module displays the loaded ontology hierarchy in the form of a DAG. Finally, Classification, the proposed system's key module, calculates the new individual's membership in different (fuzzy) concepts. Once it is attached to its belonging concepts, the Visualization module displays the concept's marks on the created DAG. Each mark is represented by a colour, which produces a coloured DAG.

¹ As part of a masters' project [15].

The colours red, orange and green are used to represent, respectively, impossible, possible and sure concepts.

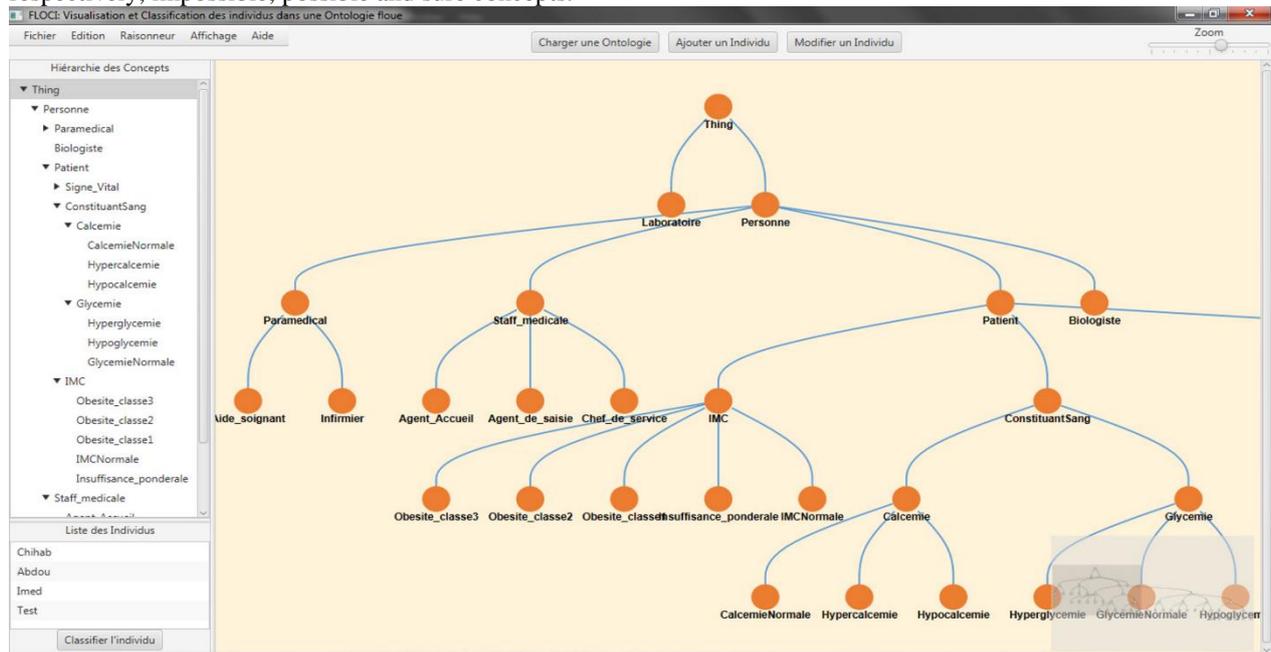


Figure 4: Fuzzy Realizer interface.

In order to better facilitate the decision-making process, membership degrees are represented by numerical values on the coloured graph nodes and also by gradations of the colour green, ranging from light green, which represents a low membership, to dark green, which represents full membership (see Figure 5).

In order to evaluate the proposed system’s performance, we carried out a range of experiments with different fuzzy ontologies, beginning with a simple Medical Checkup Fuzzy Ontology (MCFO) and then using more highly expressive and voluminous fuzzy ontologies (Fuzzy Wine², Matchmaking³, Multi-criteria decision making⁴). We present the results for two of these ontologies in the following subsections. We also compared our *Fuzzy Realizer* and the well-known fuzzy reasoner FuzzyDL [6, 9]; this was done by replacing the Classification module with the fuzzy reasoner FuzzyDL.

6.1 Medical Checkup Fuzzy Ontology (MCFO)

Uncertainty is the central critical fact about reasoning in e-health domain. Usually, doctors cannot give exact diagnoses and laboratories cannot report exact analysis results. Despite this uncertainty, doctors have to make decisions. In order to implement a decision-making

process using medical check-up fuzzy knowledge, we developed MCFO using the Fuzzy DL *SIQ(D)* and Fuzzy OWL. Then, using Fuzzy Realizer, we realized it with new individuals. Table 3 represents the description of the new (incomplete) individual Tim.

Attribute	Value
Body Temperature	37.45°
Blood Sugar	1.0 g/l
Body Mass Index	26.0 kg/m ²
Heart Pulse	Unknown
Respiratory Rate	Unknown
Diastolic Blood Pressure	70.0 mmHg
Systolic Blood Pressure	100.0 mmHg
Calcium Level	2.3 mmol/l

Table 3: description of the new individual Tim.

Although it is an incomplete individual, Fuzzy Realizer was able to classify it as low as possible in the hierarchy (see Figure 5) by providing the set of its sure (to some degree) and possible fuzzy concepts. This classification cannot be done using FuzzyDL since it does not offer a service for classifying incomplete individuals.

²<http://users.abo.fi/rowikstr/FuzzyWineOntology/FuzzyWineOntology.owl>

³<http://www.umbertostraccia.it/cs/software/FuzzyOWL/ontologies/matchmaking.owl>

⁴<http://www.umbertostraccia.it/cs/software/FuzzyOWL/ontologies/multiCriteria.owl>

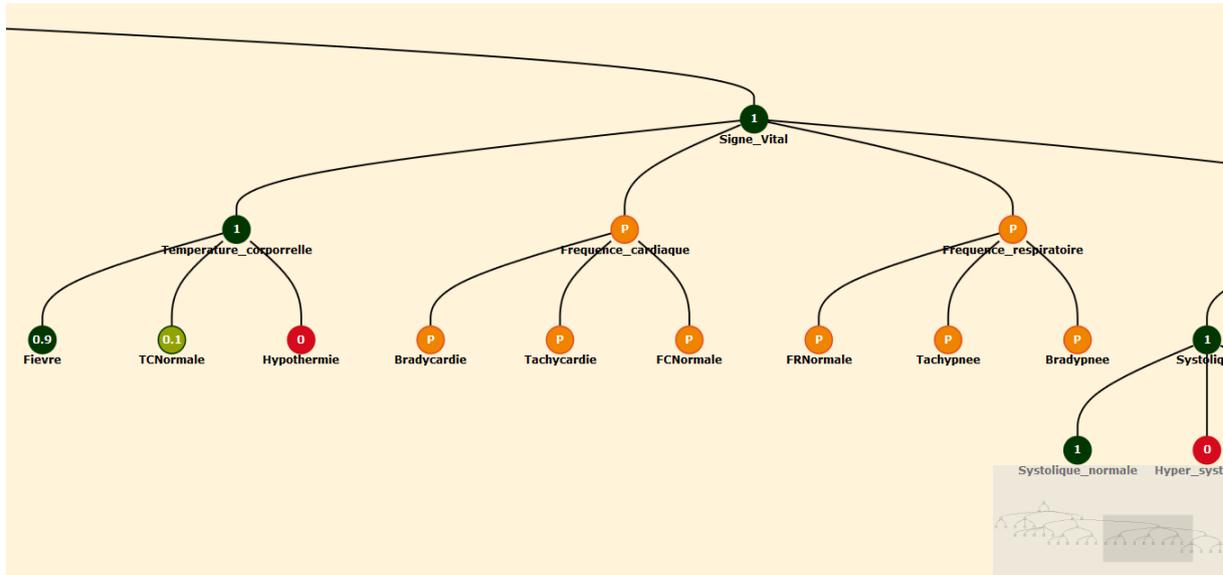


Figure 5: Zoom of the Classification of the incomplete individual Tim

6.2 Fuzzy Wine ontology

The fuzzy extension of the well-known and highly expressive Wine ontology supporting the DL *SHOIN(D)* is the most voluminous open source fuzzy ontology. Thus, we used this ontology in order to test our proposed system’s performance. Despite its large size, we have been able to realize it with new individuals using our prototype. Figure 6 shows the classification of the new individual *ChateauDeMeursauCru2007*, described in Table 4, which is considered to be a *HighUWSWine* to degree 0.1 and a fully (degree = 1) *HighPriceWine* and *TableWine*.

Attribute	Value
Price	38.6
PH	3.42
Acidity	5.8
Sugar	1.7
UWSScore	89.0
Flavor	ModerateWineFlavor
Maker	ChateauDeMeursaultWinery

Table 4: description of the new individual *ChateauDeMeursauCru2007*.

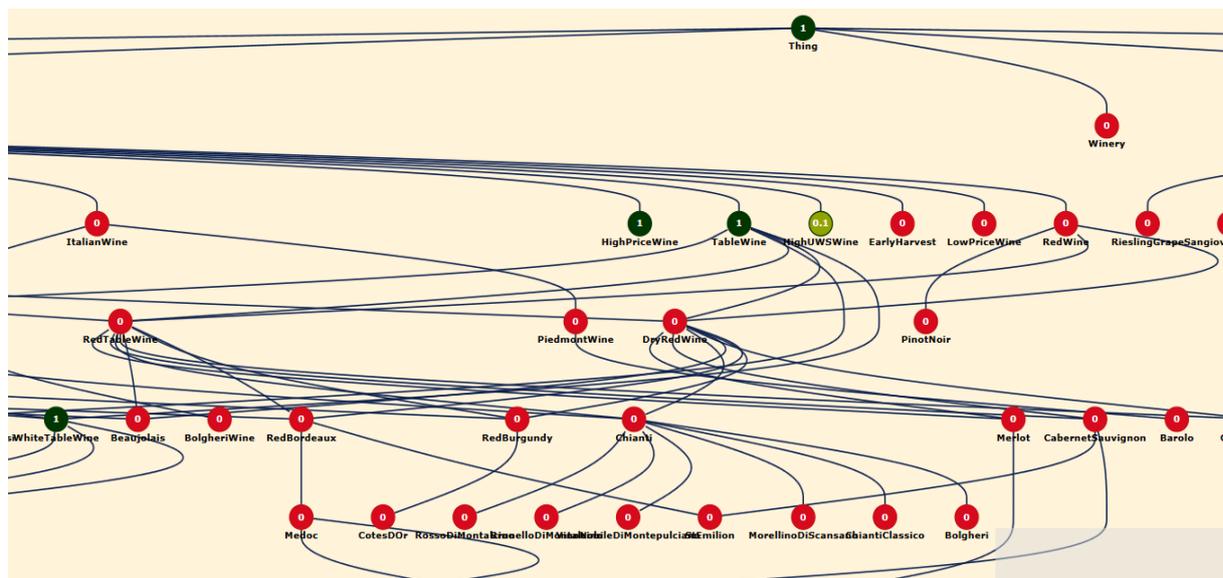


Figure 6: Realizing Fuzzy Wine.

6.3 Discussion

Although *Fuzzy Realizer* is considered to be a simple prototype providing a realization service for fuzzy ontologies, several series of tests show that it offers an efficient realization service since it results to correct classifications (all results are verified by domain experts). Moreover, it is capable of realizing any fuzzy ontology without any constraint on the represented knowledge's imperfection. It is also able to realize highly expressive fuzzy ontologies even with incomplete individuals. Indeed, it was used to realize the most voluminous open source fuzzy ontology (Fuzzy Wine).

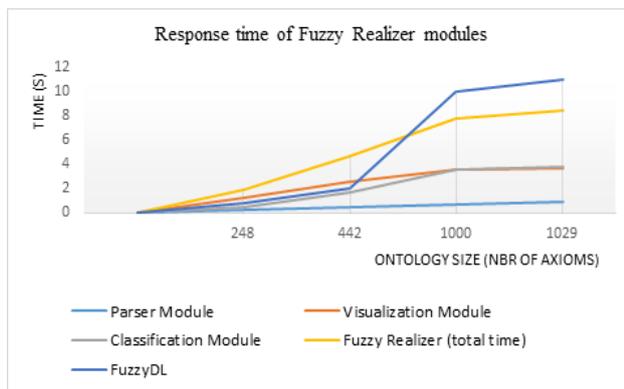


Figure 7: Response time of Fuzzy Realizer modules.

More importantly, its response time is within the limits of acceptability compared to the well-known fuzzy reasoner FuzzyDL as shown in Figure 7. All of these characteristics allow the proposed prototype to be tested in a real application and to handle real world knowledge. In sum, despite its simplicity, *Fuzzy Realizer* can be considered as an optimal solution for realizing fuzzy ontologies. In contrast, FuzzyDL is one of the most expressive and important fuzzy reasoners. However, its long runtime compared with *Fuzzy Realizer* and its inability to classify individuals in case we may lack information, which is a quite common problem, are weaknesses which cannot be ignored.

7 Conclusion

In this paper, we have proposed a fuzzy-based approach for reasoning with imperfect ontological knowledge. As a reasoning mechanism, we have integrated fuzzy logic with the most powerful human reasoning activity, known as classification. Using fuzzy classification, we have proposed *Fuzzy Realizer*, a java prototype for classifying new individuals into fuzzy ontologies. It allows (i) fuzzy concrete domains, (ii) modified and (iii) weighted concepts. We have been interested in just one reasoning task to address an aspect of fuzzy ontologies evolution namely, the realization issue. The proposed prototype can realize fuzzy ontologies even with incomplete individuals. In addition, it offers a more human-oriented classification by assigning an individual to several fuzzy sister classes which hides the sharp boundaries between them.

As future work, we would like to extend the proposed prototype so that it will not be limited to Zadeh semantics, but to be more flexible by supporting more fuzzy logics, for instance Łukasiewicz, Gödel or Product logics. We are also intended to implement the relocation process extension so that we can test and evaluate the proposed idea. Finally, in order to improve *Fuzzy Realizer*'s performance, we would like to minimize the use of the mark 'possible'. To that end, we intend to propose a new conceptualization of concepts by dividing each concept's set of attributes into two groups: key attributes and auxiliary ones. During the classification of an incomplete individual and if the 'unknown' attribute is an auxiliary one, then the current concept can be marked 'sure'. For example, if Tom has obtained a medical diploma, then even though we may lack information about his age, his address or even his last name, we can be sure that he is a doctor. Therefore, in order to mark the concept Doctor as 'sure', it is not necessary to have known values for all attributes.

Acknowledgements

We would like to thank the anonymous referees for their valuable comments on an earlier version of this paper.

References

- [1] Alexopoulos, P., Wallace, M., Kafentzis, K. & Askounis, D. (2012) 'IKARUS-Onto: a methodology to develop fuzzy ontologies from crisp ones', Knowledge and Information Systems, Vol. 32 No. 3, pp. 667-695.
- [2] Bobillo, F., Delgado, M. & Gómez-Romero, J. (2013) 'Reasoning in Fuzzy OWL 2 with DeLorean'. In Bobillo, F. et al. (Eds.), Uncertainty Reasoning for the Semantic Web II, Vol. 7123 of Lecture Notes in Computer Science, Springer-Verlag, pp. 119-138.
- [3] Bobillo, F. & Straccia, U. (2008) 'fuzzyDL: an expressive fuzzy description logic reasoner'. In Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008), IEEE Computer Society Press, pp. 923-930.
- [4] Bobillo, F. & Straccia, U. (2011) Fuzzy ontology representation using OWL 2. International Journal of Approximate Reasoning, Vol. 52 No. 7, pp. 1073-1094.
- [5] Bobillo, F. & Straccia, U. (2013) General concept inclusion absorptions for fuzzy description logics: A First step. In Description Logics, pp. 513-525.
- [6] Bobillo, F., & Straccia, U. (2016). The fuzzy ontology reasoner fuzzyDL. Knowledge-Based Systems, 95, 12-34.
- [7] Calegari, S. & Ciucci, D. (2007) 'Fuzzy ontology, fuzzy description logics and fuzzy-owl'. In Proceedings of the 7th International Workshop on Fuzzy Logic and Applications (WILF 2007), volume 4578 of Lecture Notes in Computer Science, Springer Verlag, pp. 118-126.
- [8] Djellal, A., & Boufaïda, Z. (2012) 'Conceptualisation d'une Ontologie Floue'. In Proceedings of 9eme

- Colloque sur l'Optimisation et les Systèmes d'Information, Tlemcen, Algeria, pp. 62-73.
- [9] Djellal, A., & Boufaïda, Z. (2014) 'Fuzzy ontology evolution: classification of a new individual', *Journal of Emerging Technologies in Web Intelligence*, Vol. 6 No.1, pp. 9-14.
- [10] Djellal, A., & Boufaïda, Z. (2016) Individual Relocation: A Fuzzy Classification Based Approach. In *Model and Data Engineering: 6th International Conference, MEDI 2016, Almería, Spain, September 21-23, 2016, Proceedings (Vol. 9893, p. 209)*. Springer.
- [11] Djezzar, M., Hemam, M. & Boufaïda, Z. (2012) 'Ontological re-classification of individuals: a multi-viewpoints approach'. In *Proceedings of the 2nd international conference on Model and Data Engineering. LNCS 7602 Springer*, pp. 91-102, Poitiers, France.
- [12] Ferrara, A., Ludovico, L.A., Montanelli, S., Castano, S. & Haus, G. (2006) 'A semantic web ontology for context-based classification and retrieval of music resources', *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, Vol. 2 No. 3, pp. 177-198.
- [13] Gao, M. & Liu, C. (2005) 'Extending OWL by fuzzy description logic'. In *Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence: (ICTAI 05)*, IEEE Computer Society, pp. 562-567.
- [14] Ghorbel, H., Bahri, A. & Bouaziz, R. (2009) 'Fuzzy Protégé for fuzzy ontology models'. In *Proceedings of 11th International Protégé Conference (IPC'2009)*, Academic Medical Center, University of Amsterdam, Amsterdam, Netherlands.
- [15] Hecham, A. & Iaiche, I. E. (2015) 'Système de classification et de visualisation d'instances dans une ontologie floue'. Masters thesis, Constantine 2-Abdelhamid Mehri University, Constantine, Algeria.
- [16] Kaufmann, M., & Meier, A. (2009, June). An inductive fuzzy classification approach applied to individual marketing. In *Fuzzy Information Processing Society. NAFIPS 2009. Annual Meeting of the North American (pp. 1-6)*. IEEE.
- [17] Konstantopoulos, S., & Charalambidis, A. (2010) Formulating description logic learning as an Inductive Logic Programming task. In *proceedings of the 19th IEEE International Conference on Fuzzy Systems*, IEEE Press.
- [18] Mariño, O. (1993) 'Raisonnement classificatoire dans une représentation à objets multi-points de vue'. PhD thesis, Joseph-Fourier-Grenoble I University, France.
- [19] Meier, A., Schindler, G. & Werro, N. (2008) 'Fuzzy classification on relational databases'. In J. Galindo (Ed.), *Handbook of Research on Fuzzy Information Processing in Databases*, Vol. 2, Idea Group Publishing, Hershey, PA, pp. 586-614.
- [20] Palash, D., Hrishikesh, B. & Tazid, A. (2011). 'Fuzzy arithmetic with and without using α -cut method: a comparative study'. In *International Journal of Latest Trends in Computing*, Vol. 2 No. 1, pp. 99-107.
- [21] Pérez, I. J., Wikström, R., Mezei, J., Carlsson, C., & Herrera-Viedma, E. (2013). A new consensus model for group decision making using fuzzy ontology. *Soft Computing*, 17(9), 1617-1627.
- [22] Rodríguez, N. D., Cuéllar, M. P., Lilius, J., & Calvo-Flores, M. D. (2014). A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems*, 66, 46-60.
- [23] Scharrenbach, T. & Bernstein, A. (2009) 'On the evolution of ontologies using probabilistic description logics'. In *Proceedings of the First ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web*.
- [24] Simou, N., Mailis, T.P., Stoilos, G., Stamou, G.B. (2010) Optimization techniques for fuzzy description logics. In *Description Logics*
- [25] Stoilos, G., Simou, N., Stamou, G. & Kollias, S. (2006) 'Uncertainty and the semantic web: intelligent systems', *IEEE Intelligent Systems*, Vol. 21 No. 5, pp. 84-87.
- [26] Stoilos, G., Stamou, G. & Pan, J.Z. (2010). 'Fuzzy extensions of OWL: logical properties and reduction to fuzzy description logics', *International Journal of Approximate Reasoning*, Vol. 51 No. 6, pp. 656-679.
- [27] Straccia, U. (2001) 'Reasoning within fuzzy description logics', *Journal of Artificial Intelligent Research (JAIR)*, Vol. 14, pp. 137-166.
- [28] Straccia, U. (2012) 'Description Logics with Fuzzy Concrete Domains'. arXiv preprint arXiv:1207.1410.
- [29] Straccia, U. (2013). *Foundations of Fuzzy Logic and Semantic Web Languages*. CRC Press.
- [30] Straccia, U. (2015). 'All About Fuzzy Description Logics and Applications'. In *Reasoning Web. Web Logic Rules*. Springer International Publishing pp. 1-31.
- [31] Tsatsou, D., Dasiopoulou, S., Kompatsiaris, I., & Mezaris, V. (2014). LiFR: A Lightweight Fuzzy DL Reasoner. In *The Semantic Web: ESWC 2014 Satellite Events (pp. 263-267)*. Springer International Publishing.
- [32] Werro, N. (2015). *Fuzzy classification of online customers*. Springer. ISBN : 3319159690 9783319159690
- [33] Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., ... & Sabou, M. (2015). Ontology evolution: a process-centric survey. *The Knowledge Engineering Review*, 30(01), 45-75.
- [34] Zadeh, L.A. (1965) 'Fuzzy sets', *Information and Control*, Vol. 8 No. 3, pp. 338-353. doi:10.1016/S0019-9958(65)90241-X.

