Ontology's Fuzzy Relations Modeling and Extracting Hidden Fuzzy Relations in OWL Fuzzy Improvement

Seyed Abas Montaseri
Computer Engineering Department
Islamic Azad University, Beyza Branch
Shiraz, Iran

Paper Reference Number:
Name of the Presenter: Seyed Abas Montaseri

Abstract

Semantic web uses ontology punctuation languages such as Resource Description Framework (RDF) and Web Ontology Language (OWL) for modeling the web content in way that's legible and readable for the machine. By supporting OWL, All users and systems can communicate through exchanging or uniting data on the web. But OWL employs plain and clear concepts to describe information and data. Therefore modeling and representing as method that's supported by the present verbal structure of the OWL isn't sufficient for working with vague information that is found in many applied scopes. Here the discussed problem is how to model and represent the inexplicit phase of data and knowledge in OWL. This essay represents the ontology's fuzzy relationships modeling regard to the theory of the fuzzy sets and the matrix of the fuzzy relations in the fuzzy development of the OWL, by adding the fuzzy membership value to the given ontology's entities and relations. Also some algorithms are studied in order to extract the hidden fuzzy relations between ontology concepts. At the same time saving these new relations with fuzzy OWL syntax in the RDF/XML database is studied too.

Key words: Semantic web, Ontology, Fuzzy ontology, Web ontology language, Fuzzy sets.

1. Introduction

Semantic web [1] uses ontology punctuation languages such as Resource Description Framework (RDF) and Web Ontology Language (OWL) for modeling the web content in way that's legible and readable for the machine.

By supporting OWL, All users and systems can communicate through exchanging or uniting data on the web. But OWL employs plain and clear concepts to describe information and data. Therefore modeling and representing as method that's supported by the present verbal

structure of the OWL isn't sufficient for working with vague information that is found in many applied scopes. The problem that is going to be discussed is how to model and present the inexplicit phase of data or fuzzy data on the OWL. In the recent decades efficient technologies about the inexplicit fuzzy science have been developed in various essays and articles [4, 5, 6, 7].
This essay represents the ontology's fuzzy relationships modeling regard to the theory of the fuzzy sets and the matrix of the fuzzy relations in the fuzzy development of the OWL, by adding the fuzzy membership value to the given ontology's entities and relations. Also some algorithms are studied in order to extract the hidden fuzzy relations between ontology concepts. At the same time saving these new relations with fuzzy OWL syntax in the RDF/XML database is studied too.
In the second section the function of fuzzy membership and symmetrical and asymmetrical fuzzy relations are defined. In the third section the modeling of ontology's fuzzy information based on the graph of the fuzzy ontology and the matrix of fuzzy relation is discussed. Section 4 presents the deduction algorithms for fuzzy relations between ontology concepts. In section 5 description logics and their relation with OWL are discussed. In section 6 the fuzzy OWL with its all rules and constraints are introduced. And, finally in section 7 the implementing of the algorithms in this essay are explained.

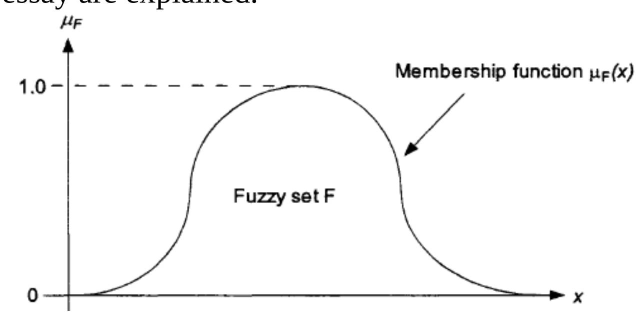

Fig 1: Fuzzy membership function of the F fuzzy set.

The $\mu_F(x)$ function is a bordered function with the label F. This label is a phrase or interpretation like similarity or tallness, which fuzzy sets are about them. In this article the fuzzy set of F is relation between 2 classes and involves the amount of fuzzy relation's membership values with the name of F between the classes of ontology concepts.

1.1 Symmetric Fuzzy Set

The symmetric fuzzy set is a set for show the undirected fuzzy relation between two classes. For example a relation like the amount of similarity is symmetric. So there is no difference between these two phrases "class A is similar to class B" and "class B is similar to class A". It means you can't find any differences between 2 sides of this relationship ($A \to B = B \to A$).

1.2 Asymmetric Fuzzy Set

Asymmetric fuzzy set is a set which is used to show directed fuzzy relation between two classes. For example a relation with priority label is an Asymmetric relation. Therefore there is difference between these two phrases "class A is prior to class B" and "class B is prior to class A". So you can find a difference between two sides of relations ($A \to B \neq B \to A$).

2. Information Modeling in Fuzzy Ontology

In this section first fuzzy ontology is defined to show the fuzzy information in the forms of classes (concepts), Individuals (instances of class) and the fuzzy relationships. Then regard to the proposed fuzzy ontology, the graph of fuzzy ontology is defined in order to prepare a picture of the real fuzzy relations and the hidden ones. Finally the matrix of fuzzy ontology relations will be introduced to save the fuzzy membership values of the real fuzzy relations and calculation and saving fuzzy membership values of hidden fuzzy relations by the algorithms in section 4 will introduce too.

2.1 Fuzzy Ontology

Fuzzy ontology in [9] is defined as $O_F = \{I, C, R, F, A\}$, and the meaning of each item is described as follow.

I : The set of individuals (instances of concepts).

C : The set of concepts that each concept $C \in \mathrm{C}$ is a fuzzy set on individual's domain ($C : \mathrm{I} \mapsto [0,1]$). The set of fuzzy ontology entities is show as E, in other words $\mathrm{E} = \mathrm{C} \cup \mathrm{I}$.

R : The set of relations. Each $R \in \mathrm{R}$ is a fuzzy relation on individual's domain ($R : \mathrm{E}^n \mapsto [0,1]$). The subsumption relation $R : \mathrm{E}^2 \mapsto [0,1]$ defines a specific relation which shows the fuzzy subclasses between entities.

F : The set of fuzzy relations on the set of entities E and a special domain containing $\mathrm{D} = \{integer, string, ...\}$. Fuzzy relations are binary functions so that each element $F \in \mathrm{F}$ is a relation like $F : \mathrm{E}^{(n-1)} \times P \mapsto [0,1]$ and $P \in \mathrm{D}$.

A : The set of rules that stated in a proper logical language. In other words they are interpretations to constrain the meaning of concepts, individuals, relations and functions.

2.2 The Fuzzy Ontology's Graph

The information of fuzzy ontology can be represented by a graph that has got both direction and label. This graph involves a set of vertex (V) and a set of edges (E). Each vertex shows a class or concepts and each edge shows a fuzzy relation between two classes are on ends of that edge ($\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$). In the mentioned graph, each edge has a label or relation title (L) and a fuzzy membership value ($\mu_F : \mathrm{L} \to [0,1]$).

Supposing that $\mathrm{L} : \mathrm{E} \to \mathrm{D}$ is mapping of the edges into set D , which is called labels, then the fuzzy ontology graph is quadruple $\mathrm{O} = (\mathrm{V}, \mathrm{E}, \mathrm{L}, \mu_F)$. Figure 2 shows an example of fuzzy ontology graph for scientific discussions of data mining.

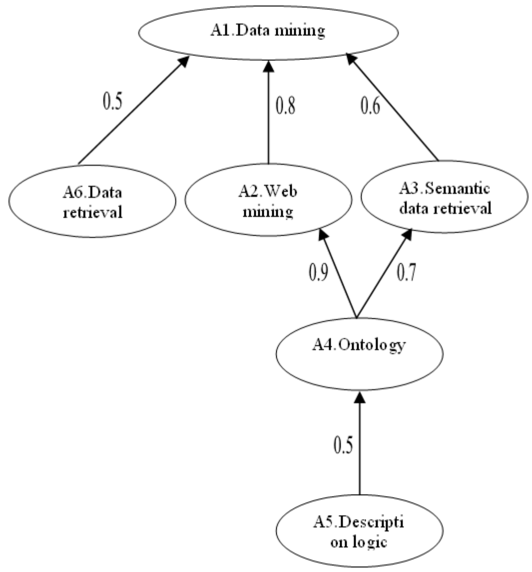

Fig 2: The fuzzy ontology's graph for data mining scientific discussions.

In Figure 2 the fuzzy membership value describes similarity between two concepts. For example web mining (A_2) with the fuzzy membership value of 0.8 is similar to data mining (A_1) and semantic data retrieval (A_3) with the fuzzy membership value of 0.6 is similar to data mining. This means web mining comparing to semantic data retrieval is more similar to data mining. Figure 3 shows the simplified form of this ontology.

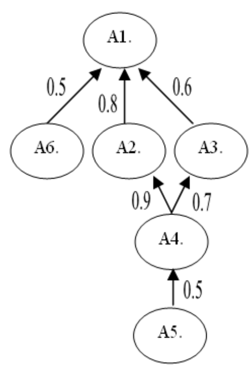

Fig 3: The simplified fuzzy ontology's graph for data mining scientific discussions.

2.3 Fuzzy Ontology's Relationship Matrix

Fuzzy ontology's relationship matrix (M) is used in order to save the fuzzy membership values of relations between concepts (classes). In this matrix the fuzzy membership value of relation between two concepts A_i and A_j $(A_i \rightarrow A_j)$ are saved in row i and column j. in the case the fuzzy relation is symmetric, the M matrix is symmetric too.

To increase the speed of process fuzzy relations between concepts by ontology's reasoner, we can attach the Fuzzy ontology's relationship matrix in the form of a XML [10] document to a fuzzy ontology that's described in OWL. The reason is the XML [11] is W3C's standard and also XML provides the syntax to describe information and this grammar is legible, readable and processable by machine.

Figure 4 shows fuzzy ontology's relationship matrix (M) for data mining scientific discussions regard to figure 3. In M row as i show's the fuzzy membership values of e_{ik} edges that get out from A_i, and column as j show's the fuzzy membership values of e_{kj} edges that get in to A_j. In figure 2 the fuzzy sets is similarity and regard to figure 4, the semantic data retrieval is similar to data mining with fuzzy membership value 0.6 (e_{A_3,A_1}) and ontology is similar to semantic data retrieval with fuzzy membership value 0.7 (e_{A_4,A_3}).

	A1	A2	A3	A4	A5	A6
A1	–	0	0	0	0	0
A2	0.8	–	0	0	0	0
A3	0.6	0	–	0	0	0
A4	0	0.9	0.7	–	0	0
A5	0	0	0	0.5	–	0
A6	0.5	0	0	0	0	–

Fig 4: Fuzzy ontology's relationship matrix of scientific data mining discussions.

2.4 Relations Properties

In fuzzy ontology the considering relations can be symmetric ant transitive. If the fuzzy ontology O_F is given, the binary relation $R : E \times E \mapsto [0,1]$ will be symmetric if

$$\forall i, j \in E \quad R(i, j) = R(j, i) \tag{1}$$

and it is transitive if

$$\forall i, j \in E \quad \sup_{k \in E}\{ t(R(i,k), R(k, j)) \} \leq R(i, j) \tag{2}$$

t is t-norm operator that's one of the defined operators on fuzzy sets ($[0,1]\times[0,1]\rightarrow[0,1]$).

In addition for given binary relation $R : E \times E \mapsto [0,1]$, the inverse relation is defined as $R^{-}(i, j) := R(j, i)$.

Therefore a relation is a symmetric, if and only if $\forall i, j \in E \quad R(i, j) = R^{-}(i, j)$.

2.4.1 Hidden Relations

In case the relation of $R(i, j)$ in the 4'th formula doesn't exist in reality, and the relations of $R(i,k)$ and $R(k, j)$ are exists for at least one k, then in such a situation the relation of $R(i, j)$ can be created in a hidden way and can named $R_V(i, j)$.

In case that relations of $R(i,k)$ and $R(k, j)$ are exists for more than one k, Then there will be the same number of distinct hidden relations exists between entities i and j. On the creation or discretion of these hidden relations, a decision can be make based on how strong and constraint are the fuzzy relations between concepts and individuals in ontology. To compare and choose an appropriate hidden relation $R_V(i, j)$ from the present hidden relations, the criterion of *COMP* can be choose regard to equations 3 and 4.

1. If fuzzy relations between concepts are flexible, means hidden relation between concepts can be created easily, So

$$R_V(i,j) = COMP(x,y) = \max(x,y) = \begin{cases} x & if \quad x > y \\ y & otherwise \end{cases} \tag{3}$$

2. If fuzzy relations between concepts are strong and constraint enough, means hidden relations between concepts can't be created easily, So

$$R_V(i,j) = COMP(x,y) = \min(x,y) = \begin{cases} y & if \quad x > y \\ x & otherwise \end{cases} \tag{4}$$

Supposing $R_{V_1}(i,j) = e_{ij} = x$ and $R_{V_2}(i,j) = e_{ij} = y$. In case the real relations between concepts are symmetric or where the formula 3 is true, then hidden relations are symmetric too. So $\forall i,j \in E \;\; R_V(i,j) = R_V(j,i)$.

3. The Hidden Relations Creation Algorithms

Matrix M involves the fuzzy membership values of real relation between concepts. Based on definition of hidden relations in section 3, the fuzzy membership values of hidden relations between concepts can be achieved by the fuzzy membership values of real relation that stored in matrix M.

If A and B and C are classes in M, and the edges e_{BA} with the membership value μ_{BA} from class B to class A and the edge e_{CB} with the membership value μ_{CB} from class C are connected to class B, The result is class A is achievable from class C is by class B. If there is no direct edge from class C to class A, Then the hidden edge e_{CA} with the membership value μ_{CA}, can be inferred regard to equation 5.

$$\mu_{CA} = \mu_{CB} \times \mu_{BA} \tag{5}$$

Achievable classes mean those classes that can be achieved through other classes with use of real edges. Unachievable classes Mean those ones witch can be achieved by other classes, and real and hidden edges. The algorithm of figure 5 slows how the hidden edges or relations can be created for achievable classes in Matrix M. In this algorithm for select a hidden relation among all existing hidden relations between two classes, compare function $COMP(x,y)$ that introduced in equations 3 and 4, can be used.

$INPUT$: *Fuzzy Ontology Matrix (M)*
$OUTPUT$: *Fuzzy Ontology Matrix with hidden edges (HM)*
C = classes in M;
HM = empty matrix with the same size as M;
HM = M
for each class i of C do
  for each class j of C do
    for each class k of C do
    if $(k \neq j$ *and* $j \neq i)$ *then*
      if $(HM[k,j] > 0)$*and*$(HM[j,i] > 0)$ *then*
        if $HM[k,i] > 0)$ *then*
          $HM[k,i] = COMP(HM[k,i]\ , m_{kj} \times m_{ji})$
        else
          $HM[k,i] = m_{kj} \times m_{ji}$
return HM

Fig 5: The hidden edges creation's algorithm for achievable classes.

The algorithm of figure 6 slows how the hidden edges can be created for all unachievable classes in Matrix M. this algorithm create hidden edges until a new edge isn't add.

$INPUT$: *Fuzzy Ontology Matrix (M)*
$OUTPUT$: *Fuzzy Ontology Matrix with all hidden edges (HM)*
C = classes in M;
HM = empty matrix with the same size as M;
HM = M
Repeat
  step 1 : HM = HM
  step 2 : HM = Create hidden edges for classes (HM)
Until hidden edges dont created in HM
return HM

Fig 6: The hidden edges creation's algorithm for all classes.

Figure 7 shows the fuzzy ontology's graph of data mining scientific discussions with all created hidden edges between classes by making use of equation 3 as comparative function.

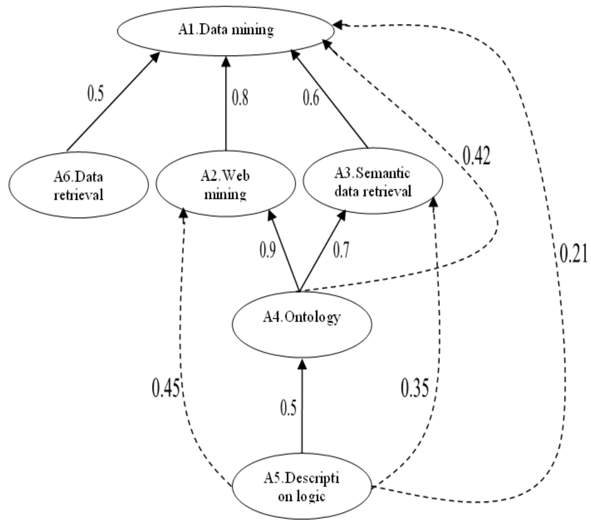

Fig 7: The fuzzy ontology's graph of data mining scientific discussions with all created hidden edges by making use of equation 3.

3.1 The Creation of Hidden Relations for Symmetric Sets

Considering A, B and C as Classes in fuzzy ontology and fuzzy set to be symmetric, and the edge e_{BA} connected from class B to class A with the membership value μ_{BA}, and the edge e_{CA} connected from class C to class A with the membership value μ_{CA}.

If there is no edge between class B and C, then two hidden edges e_{BC} and e_{CB} can be concluded from equation 6 with the membership value μ.

$$\mu = sim(\mu_{BA}, \mu_{CA}) = \frac{\min(\mu_{BA}, \mu_{CA})}{\max(\mu_{BA}, \mu_{CA})} \tag{6}$$

In some cases when fuzzy set is symmetric and two unachievable classes refer to the similar class, then there will be a relation between these two unachievable classes. Equation 6 is stated to calculate the approximate fuzzy membership value between two unachievable classes.

The algorithm of figure 8 shows, how hidden edges are create between achievable and unachievable classes in symmetric fuzzy set.

```
INPUT : F Ontology Matrix with hidden edges(HM)
OUTPUT : F Ontology Matrix with all symetric
            hidden edges (HSM)
C = classes in HM;
HSM = empty matrix with the same size as HM;
HSM = HM
for each class i of C do
   for each class j of C do
      for each class k of C do
      if (i ≠ j and j ≠ k) then
         if (HM[j,i] > 0 and HM[k,i] > 0)then
            if (HSM[j,k] > 0 and HSM[k, j] > 0) then
               HSM[j,k] = COMP(HSM [j,k] , sim(μ_ji , μ_ki ))
               HSM[k, j] = HSM[j,k]
            else
             if (HSM[j,k] = HSM[k, j]) then
                 HSM[j,k] = symetric(HM[j,i],HM[k,i]);
                 HSM[k, j] = HSM[j,k];
              else
                  HSM[j,k] = HM[j,k]
return HSM
```

Fig 8: The hidden edges creation's algorithm for all classes in symmetric fuzzy set.

Figure 9 shows the fuzzy ontology's graph of data mining scientific discussions for symmetric fuzzy set with all created hidden edges between classes by making use of equation 3 as comparative function.

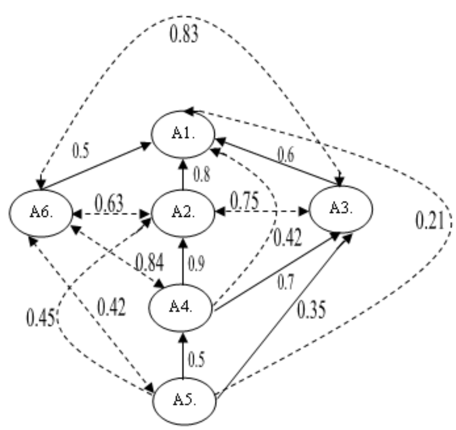

Fig 9: The fuzzy ontology's graph of data mining scientific discussions for symmetric fuzzy set with all created hidden edges by making use of equation 3.

3.2 Reduction in Number of Created Hidden Relations

The presented Algorithms in figures 6 and 8 create lots of hidden edges with different fuzzy membership values for fuzzy ontology's graph, this lead to increase complexity of calculation in ontology's process and deduction. So to decrease the complexity of calculation, threshold

T is used for removing created hidden edges with the membership values smaller than T regard to equation 7.

$$M[i, j] = \begin{cases} M[i, j] & if\ (M[i, j] \geq T) \\ 0 & otherwise \end{cases} \quad (7)$$

The threshold T can be chosen base on equation 8 or any other criteria.

$$T = \frac{MAX(M) - MIN(M)}{2} \quad (8)$$

In equation 8, MIN is the Minimum and MAX is the Maximum of fuzzy membership values that exists in the fuzzy ontology relationship matrix. Threshold T decreases the processing size and speedups it.

4. Description logics

Description logics [12, 13, 14] are a family of knowledge representing methods based on logics that are designed to present knowledge and inference about an applied scope's knowledge in a structural and suitable comprehensible method.

The real semantics and in Description logic are concepts, and roles. The special description logic is basically determined by constructors that make possible the creation of complicated concepts and roles from simple and atomic ones [15]. In other words description logic is a set of class's rules to present information that, explained through the use of various constructors for the establishment of more complicated stages from simplified stages and also for preparing perfect and reliable services. Description logics have a specific impact on OWL language, especially for bringing language's semantics, choosing language's constructors and mixing all data types and data values.

Some example of description logics are $SHOIN(D)$ [16], $SHOIQ$ (that form base of standard OWL.1) [17], $SHIQ$ [18] and $SHIF(D)$ [14].

4.1 Fuzzy Description Logic

Fuzzy description logic can show imprecise concepts and roles, and collaboration between this roles and concepts. As a result the fuzzy Description logic must be defined on the base of syntax and Semantics.

4.1.1 Syntax

Syntax rules of OWL are a graph of RDF type, in other words a set of RDF triples. Web ontology's graph can be written in many various syntax forms.

Although the semantic of OWL is only determined by RDF graph, but the available document uses some RDF/XML documents with specials syntax forms to represent triples RDF. The syntax is prepared for fuzzy ontology in [9].

4.1.2 Semantics

Semantic is one of the key and important properties in description logic. It means that formal languages are defined with appropriate semantics. The standard method to determine the concept of description logic is through semantics theory and the goal is describing the relation between languages grammar and intended models in a domain. The domain is a set of objects and description of class's names, properties and subordinate sets and binary relations.

A model made of domain that's written as Δ^I and interpretation function that's written as I . The fuzzy semantics is presented for fuzzy ontology in [9].

5. Fuzzy OWL Language

Fuzzy ontology was defined in section 3. The next stage is defining a proper new fuzzy language for implementation fuzzy ontology. The chosen language by most ontology editors is OWL DL. Since this language prepare most expressiveness with decision making for inference systems for all users [20]. But OWL DL doesn't allow use of presented data with vague definition. In this section by adding fuzzy membership value to ontology's entities and relations corresponding with fuzzy $SHOIN(D)$, the development of OWL DL that's called fuzzy OWL shown.

Paying attention to fuzzy sets, non fuzzy concepts are concepts with fuzzy membership value of 1. So the class's descriptions in OWL can be developed to class's description of fuzzy concepts. If fowl is the name space of fuzzy knowledge, Then class's descriptions (Intersection, Union, Complement) and class's rules (Subsumption, Equivalence) can apply RDF/XML syntax of OWL as it's shown in table 1.

Fuzzy Logic Syntax	Example for Fuzzy OWL
A, B	<fowl:Class rdf:ID="A"/> <fowl:Class rdf:ID="B"/>
$A \cap B$	<fowl:Class> <fowl:intersectionOfrdf:parseType="Set"> <fowl:Class rdf:about="#A"/> <fowl:Class rdf:about="#B"/> </fowl:intersectionOf> </fowl:Class>
$A \cup B$	<fowl:Class> <fowl:unionOf rdf:parseType="Set"> <fowl:Class rdf:about="#A"/> <fowl:Class rdf:about="#B"/> </fowl:unionOf> </fowl:Class>
-A	<fowl:Class> <fowl:complementOf> <fowl:Class rdf:about="#A"/> </fowl:complementOf> </fowl:Class>

$\geq \alpha\ \exists R.A$	<fowl:Restriction> <fowl:onProperty rdf:resource="#R"/> <fowl:someValuesFrom rdf:resource="#A"/> <fowl: moreOrEquivalent fowl:value= α /> </fowl:Restriction>
$\forall R.A$	<fowl:Restriction> <fowl:onProperty rdf:resource="#R"/> <fowl:allValuesFrom rdf:resource="#A"/> </fowl:Restriction>
(A ô B) å α	<fowl:Class rdf:ID="A"> <fowl:subClassOf rdf:resource="#B"/> <fowl:ineqtype fowl:value= α /> </fowl:Class>
$A \cong B$	<fowl:Class rdf:ID="A"> <fowl:equivalentClass rdf:resource="#B"/> </fowl:Class>

Table1. FUZZY CONSTRUCTORS AND RULES.

Fuzzy constraint can also be considered as domains from individuals that belong to fuzzy concepts. So as shown in table 2, fuzzy constraints can be presented with membership function's value of these concepts. In proposed fuzzy OWL, *fowl:membershipOf* and *fowl:value* shows fuzzy membership value, In other words proportion of individual's dependence to concepts. In table 1, *fowl:ineqtype* replaced by one of the following comparative relations *fowl:moreThan*, *fowl:lessThan*, *fowl:moreOrEquivalent* or *fowl:lessOrEquivalent* that shows quality of fuzzy subsumption relation between two concepts.

Fuzzy Constraints	**Example for Fuzzy OWL**
$A(a) \geq \alpha$	<fowl:individual fowl:name="a"> <fowl:membershipOf rdf:resource="#A"/> <fowl:moreOrEquivalent fowl:value= α /> </fowl:individual>
$A(a) \leq \alpha$	<fowl:individual fowl:name="a"> <fowl:membershipOf rdf:resource="#A"/> <fowl:lessOrEquivalent fowl:value= α /> </fowl:individual>
$R(a,b) \geq \alpha$	<fowl:individual fowl:name="a" fowl:name="b"> <fowl:membershipOf rdf:resource="#R"/> <fowl:moreOrEquivalent fowl:value= α /> </fowl:individual>

$A(a) > \alpha$	<fowl:individual fowl:name=”a”> <fowl:membershipOf rdf:resource=”#A”/> <fowl:moreThan fowl:value= α /> </fowl:individual>
$A(a) < \alpha$	<fowl:individual fowl:name=”a”> <fowl:membershipOf rdf:resource=”#A”/> <fowl:lessThan fowl:value= α /> </fowl:individual>
$R(a, b) \leq \alpha$	<fowl:individual fowl:name=”a” fowl:name=”b”> <fowl:membershipOf rdf:resource=”#R”/> <fowl:lessOrEquivalent fowl:value= α /> </fowl:individual>
$R(a, b) > \alpha$	<fowl:individual fowl:name=”a” fowl:name=”b”> <fowl:membershipOf rdf:resource=”#R”/> <fowl:moreThan fowl:value= α /> </fowl:individual>
$R(a, b) < \alpha$	<fowl:individual fowl:name=”a” fowl:name=”b”> <fowl:membershipOf rdf:resource=”#R”/> <fowl:lessThan fowl:value= α /> </fowl:individual>

Table2. FUZZY CONSTRAITS

6. Fuzzy OWL Editor

For implementing fuzzy ontology’s creation and management, a derivative pattern from OWL protégé software [21] that is a tool for ontology’s creation and management is provided. This software creates and manages ontologies in OWL syntax such as the name of this software (OWL protégé) shows.

Figure 10 represents the environment of designed fuzzy ontology editor that’s used for creation and manage fuzzy ontology. In this implementation we’re create the fuzzy ontology for data mining scientific discussions paying attention to this point that each class possess at least one instance. Since similarity relation between data mining scientific discussions is basically defined among individuals (class instances).

In represented fuzzy ontology editor for implementing the discussions of this essay, the information of classes, subclasses, individuals and real fuzzy relations between individuals is created by user and stored in RDF/XML data base, based on the fuzzy OWL syntax is shown in table 1 and 2.

By running the given algorithms in section 4, hidden relations between concept’s individuals were inference and then added to previous information based on the fuzzy OWL syntax is shown in table 2.

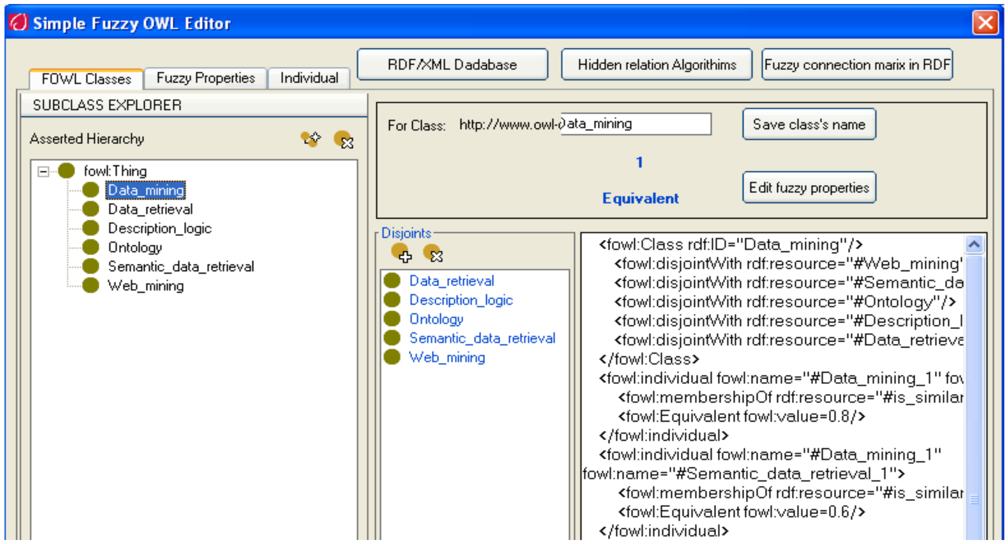

Fig 10: Fuzzy ontology's creation and management software.

Figure 11, 12 showing execution form of represented algorithms in figures 5 and 6 respectively, with resulted fuzzy ontology's relationship matrix for data mining scientific discussions.

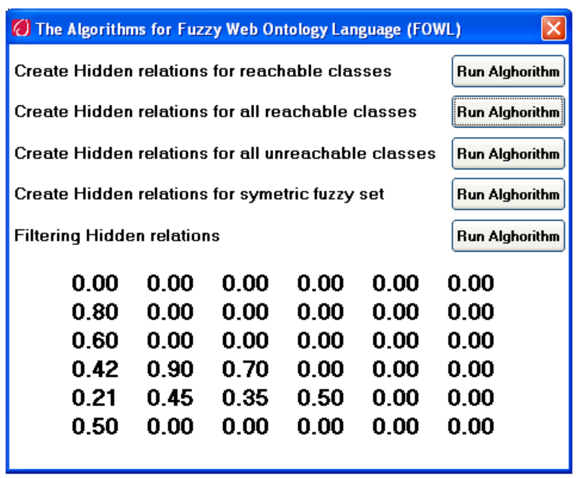

Fig 11: Fuzzy ontology's relationship matrix resulted from algorithm 5.

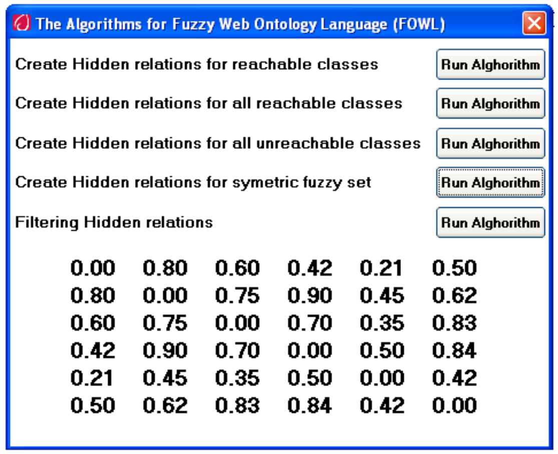

Fig 12: Fuzzy ontology's relationship matrix resulted from algorithm 6.

Figure 13 shows Fuzzy ontology's relationship matrix of data mining scientific discussions in RDF/XML syntax.

```
<?xml:namespace name="http://www.w3.org/TR/WD-rdf-syntax#"as="RDF"?>
<RDF:RDF>
<RDF:Description RDF:HREF="#'Data_mining'">
   <fuzzyrelation>
     <toclass> RDF:HREF="#'Web_mining'" </toclass>
     <RDF:Value>0.8</RDF:Value>
   </fuzzyrelation>
   <fuzzyrelation>
     <toclass> RDF:HREF="#'Semantic_data_retrieval'" </toclass>
     <RDF:Value>0.6</RDF:Value>
   </fuzzyrelation>
   <fuzzyrelation>
     <toclass> RDF:HREF="#'Ontology'" </toclass>
     <RDF:Value>0.42</RDF:Value>
   </fuzzyrelation>
   <fuzzyrelation>
     <toclass> RDF:HREF="#'Description_logic'" </toclass>
     <RDF:Value>0.21</RDF:Value>
   </fuzzyrelation>
   <fuzzyrelation>
     <toclass> RDF:HREF="#'Data_retrieval'" </toclass>
     <RDF:Value>0.5</RDF:Value>
   </fuzzyrelation>
</RDF:Description>
                                   .
                                   .
                                   .
```

Fig 13: Fuzzy ontology's relationship matrix in RDF/XML syntax.

7. CONCLUSION

There are vague and fuzzy information in various applied domains and scopes. The description of these fuzzy data and information and then saving these descriptions in RDF/XML data bases by fuzzy OWL, provides a chance for both people and machines to use and inference approximate and vague information in semantic web that wasn't decidable or ignored in searches and comparisons before this.

The fuzzy membership values of relation's between classes were saved in the fuzzy ontology's relationship matrix and then by use of the algorithms in figures 5, 6 and 8, fuzzy membership values of hidden fuzzy relations created in this matrix and then saved in the RDF format document.

Figure 14 shows the process of fuzzy ontology with hidden relations creation, from fuzzy ontology and the fuzzy relationship matrix (M).

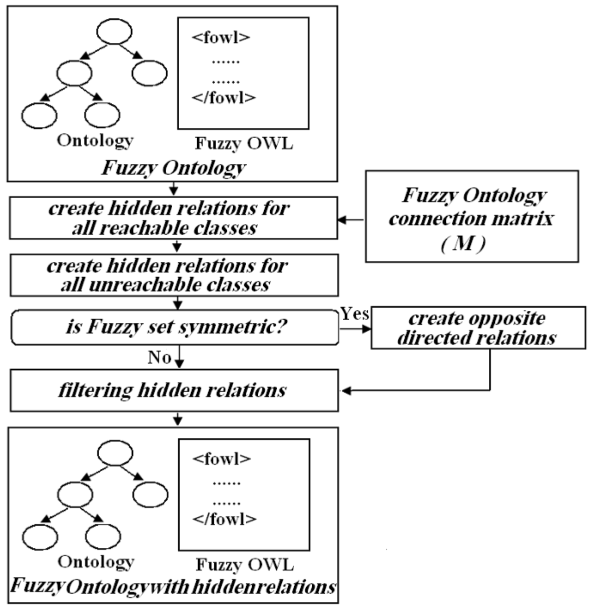

Fig 14: The process of fuzzy ontology with hidden relations creation.

Some editors and reasoner systems have recently been made, to work with ontologies. Some of these ontology editors that can be mentioned include Oiled, Protégé, Swoop, Construct, Ontotrack and some of these reasoner systems contain Cerebra, FaCT++, Kaon2, Pellet and Racer.

Reasoning is an important component of tools and services that help the users and applications to design and maintain the high quality ontology's and answers questions.

The tools and reasoner systems that were mentioned above aren't able to design and maintain and inference bout fuzzy ontologies, therefore the next step is development of these tools and reasoner systems in a way that sore the class concepts, fuzzy relations and constrain between classes and individual in fuzzy OWL syntax and perform inference operations on these fuzzy information.

According to the fuzzy ontology's relations modeling that has been introduces in this essay, these tools and reasoner systems can use both the fuzzy relationship matrix of ontology that stored in the XML document and fuzzy ontology that's described in OWL to decrease the calculation complexity of fuzzy information's process related to a specific applied domain.

References

[1] Berners-Lee, T., Hendler, T., Lassila, J., *"The semantic web"*, Scientific American 284, 34–43(2001).

[2] Giugno, R., Lukasiewicz, T., *"P-SHOQ(D): a probabilistic extension of SHOQ(D) for probabilistic ontologies in the Semantic Web"*, Technical Report INFSYS RR-1843-02-06, Institute of Information system, Technische University at Wien, April 2002.

[3] Straccia, U., *"Reasoning within Fuzzy Description Logics"*, Artif.Intell.Res, 14:137-166, 2001.

[4] Tresp, C. B., Molitor, R., *"A description logic for vague knowledge"*, Proc. Of ECAI-98, pp 361-365, 1998.

[5] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. F., "The Description Logic Handbook: Theory, Implementation, and Applications", Cambridge University Press, 2003.

[6] Resource Description Framework (RDF) Model and Syntax, http://www.w3.org/TR/1998/WD-rdf-syntax-19980216.

[7] Dean, M., eds, G., van Harmelen, S., Hendler, F. J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., Stein, L. A., *"OWL Web Ontology Language Reference"*, http://www.w3.org/TR/owl-ref/, Mar. 2003. *W3C working draft*.

[8] OWL: Ontology Web Language(2004), http://www.w3.org/2004/OWL/.

[9] Zadeh, L.A., *"Fuzzy sets"*, Inform. and Control 8, 338–353 (1965).

[10] Calegari, S., Ciucci, D., *"Fuzzy Ontology, Fuzzy Description Logics and Fuzzy-OWL"*, In: F. Masulli, S. Mitra, and G. Pasi (Eds.):WILF 2007, LNAI 4578, pp. 118–126, 2007. Springer, Heidelberg (2007).

[11] Extensible Markup Language (XML) 1.0, http://www.w3.org/TR/2006/REC-xml-20060816.