Journal of Engineering Technology
Volume 3, July. 2015, Pages 114-134

ORIGINAL ARTICLE

Development of Fuzzy Rough Features in Ontology Knowledge Representation

Nilavu Devadoss [1] and Sivakumar Ramakrishnan [2]

[1] Research Scholar,
Department of Computer Science,
A.V.V.M Sri Pushpam College of Bharathidasan University,
Thanjavur, TamilNadu, India.
[2] Associate Professor,
Department of Computer Science,
A.V.V.M Sri Pushpam College of Bharathidasan University,
Thanjavur, TamilNadu, India.

Abstract. Knowledge about a system contains ambiguity and vagueness. Fuzzy ontologies using rough concepts are able to deal with imprecise and vague knowledge. Several important amount of work has been carried out in the field of fuzzy ontologies during the last years and they have been successfully used in several applications, but appropriate methodology to represent fuzzy rough concepts using annotation properties have not yet been considered in fuzzy ontologies. In this article, we extend the current standard language *OWL2*, to deal with uncertainty features and also propose an appropriate methodology to represent fuzzy rough ontologies using *OWL2* annotation properties. We also provide an plug-in, used to edit fuzzy rough ontologies using *OWL2* annotation properties and some parsers that translate the fuzzy rough ontologies into the languages supported by reasoners. Mainly, this paper attempts to represent the fuzzy rough ontologies using *OWL2*. We also conduct a large user study to prove the effectiveness of the Fuzzy rough ontology representation using *OWL2* annotation properties with the help of the *OntoSri* plug-in named by *FuzzyRough*.

Keywords: Fuzzy rough OWL2, Fuzzy rough ontologies, Fuzzy rough description logics, Semantic web, OntoSri plug-in.

1 INTRODUCTION

Ontology is a hierarchical description of significant classes (or concepts) in a specific domain, along with the description of the properties (of the instances) of each concept. In other words, ontology can be defined as a systematic description of *part-of* relationships and *entity dependencies.* 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 easiest way. Now-a-days description logic has become a keystone of the Semantic Web for its use in the design of ontologies. The Web Ontology Language Description Logics (OWL DL) become less suitable in domains in which the concepts to be represented do not have precise definitions. Recently, it has been noted that classical ontologies are not appropriate to deal with vague and imprecise knowledge, which is fundamental to several real world domains [28]. To handle this problem, the use of fuzzy ontology with rough concepts offers a solution. In our work, we extend the current standard language *OWL2*, to deal with uncertainty and also provide a well-defined methodology to represent fuzzy rough ontologies. In addition to that, we implement an *OntoSri plug-in* to edit fuzzy rough ontologies using *OWL2* annotation properties and a general parser that translates the fuzzy rough ontologies into the languages supported by some DL reasoners. This paper is structured as follows: we describe some basic background on knowledge representation using fuzzy in section 2. More precisely, section 3 shows the main contribution of our work. Then, Section 4 illustrates the methodology with an example. Implementation and experimental evaluation of our work is presented in section 5. An empirical study is described in section 6. Related work is shown in section 7. At last, the conclusion and future works are included in section 8.

2 BACKGROUND OF THE WORK

This section presents some basic background on knowledge representation using fuzzy such as fuzzy ontology (Section 2.1), type-2 fuzzy (section 2.2), fuzzy rough ontology representation (section 2.3).

2.1Fuzzy ontology

Surveying the literature, we can find that there is no unique definition of fuzzy ontology. In the simplest case [2], a fuzzy ontology is a pair (C, R), where C is a set of (fuzzy) concepts and R is a set of (fuzzy) binary (n-ary) relations. In various approaches, this pair can be extended in several ways:

- individuals (I), fuzzy axioms (A) [10],
- concept hierarchy (H) and axioms [29],
- attributes of a concept, concept hierarchy, fuzzy events of a concept [33].

Fuzzy ontology can be seen as extended domain ontology [16], which makes use of the exact domain and fuzzy information processing as follows:

(i) the input is unstructured data;
(ii) the definition of related concepts in the particular domain,
e.g. instances,objects, and their relationships;
(iii) the generation of domain ontology;
(iv) the domain ontology extended as fuzzy ontology; and
(v) applying the fuzzy ontology to the specific domain.

Classical ontology languages are not appropriate to deal with *vagueness* or *imprecision* in knowledge. Hence, Description Logics for the semantic web can be enhanced by various approaches to handle probabilistic or possibilistic uncertainty and vagueness. Though fuzzy logic was introduced in the 1960's [33], the research on fuzzy ontologies was approximately non-existent before 2000, so we can assert that this is a quite new research field with a grand potential. This is even more astonishing considering that [24] reasoned already in the 1980's why the use of fuzzy logic as the foundation for ontology building would be useful and solve many problems pertaining to classical ontologies. Pena [24] proposes "to discard the *maximality* rule, according to which only overall true sentences are true, and accepting instead the rule of *endorsement*, which specifies that anything is more or less true, is true".

Following are important *advantages of fuzzy ontology* :

- positing fuzzy predicates habitually simplifies our theories in most scientific fields

- fuzzy predicates are more reasonable, and give us a more attractive and cohesive worldview, than their crisp matching part.

Concepts are rather vague than precise in the context of semantic Web and multimedia applications. There are rising needs to deal with vague knowledge. So it is important to cope with the inexact concepts on the Semantic Web. The goal of the research of fuzzy ontology is to integrate these characteristics. The fuzzy ontology is capable of dealing with fuzzy knowledge [13]. *Abulaish* and Dey [1] propose a fuzzy ontology framework in which a concept descriptor is represented as a fuzzy relation which encodes the degree of a property value using a fuzzy membership function. Calegari and Davide [9] show in their research how a Fuzzy Ontology based approach can improve semantic documents retrieval. In the ontology-based CBR paradigm, Alexopoulos et al., [4] present a novel CBR approach that manages and utilizes imprecise knowledge through the integration of Fuzzy. On the aspect of knowledge representation, Soomi et al., [30] build an Area Profile Ontology and import other related ontology to annotate data from disparate sources, which promote the improvement of knowledge. Mohd Kamir et al., [23] present an ontology and semantic approaches for using data from heterogeneous database. Harshit [14] share the development process of an ontology which is developed for knowledge management in an enterprise.

2.2Type 2 Fuzzy

We know very well, it has been widely pointed out that classical ontology is not sufficient to deal with imprecise and vague knowledge for some real-world applications like personal diabetic-diet recommendation. Apart from that, fuzzy ontology can powerfully help to handle and process imprecise data and knowledge. A recent development in the field is the appearance of type-2 fuzzy ontologies (T2FO). Lee et al., [19] introduced a T2FO, which is composed of (i) a type-2 fuzzy personal profile ontology (ii) a type-2 fuzzy food ontology and (iii) a type-2 fuzzy-personal food ontology. Li et al., [39] newly represent a type-2 fuzzy version of *ALC* and express its syntax, semantics and reasoning algorithms. In addition to that Li et al., [20] specify an implementation of the logic with type-2 fuzzy OWL. Lee et al., [19] represent the computer Go knowledge with the extension of a Fuzzy Markup Language (FML)-based type-2 fuzzy ontology. It consists of an FML transformation mechanism, a type-2 fuzzy set inference mechanism and a type-2 fuzzy set construction. Mezei and Wikstrom [22] present new definitions of ordered weighted averaging distance (OWAD) operators for interval-valued fuzzy numbers (IVFN) and show that the new definitions satisfy some important properties. Wang et al., [32] present the technologies of the type-2 fuzzy sets and fuzzy inference approach based on FML. Bukhari and Kim [8] employ an integrated secure type-2 fuzzy ontology multi-agent platform to completely automate the process of manual air ticket booking. Recently, T2FS characterized by Membership Functions (MFs) that are themselves fuzzy, have attracted interest. Interval Type-2 Fuzzy Set (IT2FS), a special case of T2FS, is currently the most widely used because of reduced computational cost [21]. The proposed T2FO stores T2FSs and is an extended version of the fuzzy ontology [15]. A T2FO is a knowledge representation model to describe the domain knowledge with uncertainty.

2.3 Fuzzy Rough Ontology Representation

Ontology is broadly used in the areas of web based data mining, knowledge engineering, etc. During the last years, several domains in the Semantic Web engage different sorts of imprecision, ontologies must be expressive enough to compact with imperfect information. Most of approaches to handle imprecision in ontologies deal either with probability or use fuzzy set and fuzzy logic methods for representing ontologies for intrinsically vague domains. Klinov et al., [18] show how they can be balanced by rough set techniques to capture another type of vagueness caused by approximation spaces. Ishizu et al., [17] formulate a concept of rough ontology, define upper and lower approximation, approximation accuracy of preference, granularity concept of preference. Bobillo [7] represent generalized fuzzy rough description logics to handle uncertainty.

Several approaches have been proposed to represent various forms of uncertainty in ontologies. By surveying the literature, we can find that they offer syntactic modifications in some of the axioms of the ontology. And also, they do not provide appropriate format to represent fuzzy rough ontologies that can be easily managed by current *OWL2* editors. Some other approaches handle vagueness in terms of approximated values. But they do not use annotation properties. Apart from these approaches, to handle uncertainty, we represent annotation properties using fuzzy rough concepts in *OWL2* to represent fuzzy rough ontologies and to enable current *OWL2* editors for fuzzy rough ontology representation in an efficient manner.

3 METHODOLOGY OF FUZZY ROUGH ONTOLOGIES

An efficient methodology to represent fuzzy rough ontologies in *OWL2* is presented in this section. The main theme of our approach is to use *OWL2* ontology and extend their elements with annotation properties representing the characteristics of the fuzzy rough ontology that *OWL2* cannot directly encode. From the literature survey we understood various ways of knowledge representation using fuzzy concepts. But this is the tremendous effort towards the representation of fuzzy rough ontologies.

3.1Needs of Fuzzy Rough Ontologies

There are several syntactic differences between the fuzzy rough ontologies and the non fuzzy ontologies. There are some cases depending on the annotated element. They are as follows:

- Fuzzy rough modifiers have dissimilarity in the crisp case.
- Fuzzy rough data types have a syntactic difference with the crisp case.
- Some Fuzzy rough concepts have dissimilarity in the crisp case.
- Some fuzzy rough roles have non equivalence in the crisp case.
- Some rough axioms require an inequality sign and a degree of truth: *(A1-A5)*,*(A8)*, *(A12-A13).*

.

3.2 Annotations

In fuzzy rough ontology, we use an annotation property `fuzzyroughLabel`. Moreover, for every element of the ontology there can be at most one annotation of this type. Each and every annotation is delimited by start tag a `<fuzzyroughOwl2>` and an end tag </fuzzyroughOwl2>, with an attribute fuzzyroughType specifying the fuzzy rough element being tagged.

3.3 Fuzzy Rough Modifiers

The fuzzy rough modifiers have parameters p, q, r. In this specified case, the value of the fuzzyroughType is modifier and there is a tag Modifier with an attribute type.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="modifier">
<Modifier>
</fuzzyroughOwl2>
<Modifier>:=
<Modifier type="triangular" p="<double>" q="<double>"
r="<double>" />|
<Modifier type="linear" r="<double>"/>
```

3.4 Fuzzy Rough Data types

In this section, we define fuzzy rough atomic data types and fuzzy rough modified data types.

Fuzzy Rough Atomic Data types

The fuzzy rough data types have parameters k1, k2, p, q, r, s. The parameter k1 should contain xsd:minInclusive values and k2 should contain xsd:maxInclusive values where the values may be integer or double.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="datatype">
<Datatype>:=
</fuzzyroughOwl2>
<Datatype type="left shoulder" p="<double>" q="<double>" /> |
<Datatype type="right shoulder" p="<double>" q="<double>" /> |
<Datatype type="triangular" p="<double>" q="<double>"
r="<double>" /> |
<Datatype type="trapezoidal" p="<double>" q="<double>"
r="<double>" s="<double>" />
```

Fuzzy Rough Modified Data types

Fuzzy rough modified data types contain two parameters: They are as the modifier and alter (the name of the fuzzy rough data type that is being modified).

Annotation syntax

```
<fuzzyroughOwl2 fuzzyroughType="datatype">
<Datatype type="modified" modifier="<String>"alter="<String>"/>
</fuzzyroughOwl2>
```

Here modifier is defined as a fuzzy rough modifier and alter is defined as a fuzzy rough data type. Moreover, alter should contain a different name than the annotated data type.

3.5 Fuzzy Rough Concepts

In this case, we create a new concept D. Then, we add an annotation property into the new concept describing the type of the constructor and the value of their parameters. Furthermore, the fuzzy rough concept is approximated by means of *upper approximation, loose upper approximation, tight upper approximation, lower approximation, loose lower approximation and tight lower approximation.* In this paper, we reveal fuzzy rough concepts in terms of fuzzy rough modifiers such as `exceedingly`, `certainly`, `very`, `somewhat`, `relatively`, `nearly`, `indeed`, `veryvery` using annotation properties.

Fuzzy Rough Upper Approximation

In fuzzy rough upper approximation, the value of the `type` is `UpperApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="concept">
<UpperApproximated-Concept>
</fuzzyroughOwl2>
<UpperApproximated-Concept>:=
<Concept                          type="UpperApproximated"
modifier="<String>"alter="<String>" />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

Fuzzy Rough Loose Upper Approximation

In fuzzy rough upper approximation, the value of the `type` is `LooseUpperApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="concept">
<LooseUpperApproximated-Concept>
</fuzzyroughOwl2>
<LooseUpperApproximated -Concept>:=
<Concept                     type="LooseUpperApproximated"
modifier="<String>"alter="<String>" />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

Fuzzy Rough Tight Upper Approximation

In fuzzy rough tight upper approximation, the value of the `type` is `TightUpperApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="concept">
<TightUpperApproximated-Concept>
</fuzzyroughOwl2>
<TightUpperApproximated -Concept>:=
<Concept                     type="TightUpperApproximated"
modifier="<String>"alter="<String>" />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

Fuzzy Rough Lower Approximation

In fuzzy rough lower approximation, the value of the `type` is `LowerApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType=”concept”>
<LowerApproximated-Concept>
</fuzzyroughOwl2>
<LowerApproximated -Concept>:=
<Concept type=”LowerApproximated”
modifier=”<String>”alter=”<String>” />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

Fuzzy Rough Loose Lower Approximation

In fuzzy rough loose lower approximation, the value of the `type` is `LooseLowerApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType=”concept”>
<LooseLowerApproximated-Concept>
</fuzzyroughOwl2>
<LooseLowerApproximated -Concept>:=
<Concept type=”LooseLowerApproximated”
modifier=”<String>”alter=”<String>” />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

Fuzzy Rough Tight Lower Approximation

In fuzzy rough tight lower approximation, the value of the `type` is `TightLowerApproximated`. Furthermore, it also contains two extra attributes, `modifier` (fuzzy rough modifier) and `alter`.

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType=”concept”>
<TightLowerApproximated-Concept>
</fuzzyroughOwl2>
<TightLowerApproximated -Concept>:=
<Concept type=”TightLowerApproximated”
modifier=”<String>”alter=”<String>” />
```

Here `alter` should have a different name than the type of the annotation and `modifier` is defined as the fuzzy rough modifier.

3.6 Fuzzy Rough Weighted Concepts

In this, the value of `type` is `weighted`. It also contains two additional attributes such as `value (0, 1]` and `alter` (the name of the fuzzy rough concept that is being weighted).

```
<fuzzyroughOwl2 fuzzyroughType=”concept”>
<Weighted-Concept>
</fuzzyroughOwl2>
<Weighted-Concept>:=
<Concept type=”weighted” value =”<Double>”alter=”<String>”/>
```

Here `alter` should contain the different name than the annotated type.

3.7 Fuzzy Rough Weighted Sum Concepts

In this case, the value of the `type` is `weightedsum`. Moreover, it also contains two additional tags showing `Weighted Concept`.

Annotation syntax

```
<fuzzyroughowl2 fuzzyroughType= “concept”>
<Concept type = “weighted sum”>
               (<Weighted – Concept>)+
</Concept>
</fuzzyroughowl2>
```

3.8 Fuzzy Rough Nominals

In this case, the value of the `type` is `nominal`. It also contains two additional attributes such as `value (0, 1]` and an `indiv` (the name of the individual that is being modified).

Annotation Syntax

```
<fuzzyroughOWL2 fuzzyroughType= “concept”>
<Fuzzyrough-nominal >
</fuzzyroughOwl2>
<Fuzzyrough-nominal >:=
<Concept type=”nominal” value=”<Double>”indiv=”<String>”/>
```

3.9 Fuzzy Rough Roles

In this case, we create a new role **R**. Then, we add an annotation property describing the type of the constructor and the value of their parameters.

Fuzzy Rough Modified Roles

In this case, the value of `type` is `modified`. It also contains two attributes such as `modifier` and `alter` (the name of the fuzzy rough role that is being modified).

```
<fuzzyroughOwl2 fuzzyroughType=”role”>
<Modified-Role>
</fuzzyroughOwl2>
<Modified-Role>:=
<Role type=”modified” modifier=”<String>”alter=”<String>”/>
```

Here the `modifier` should be a fuzzy rough modifier and `alter` should have a different name than the annotated role.

3.10 Fuzzy Rough Axioms

It is possible to add a degree of truth to some axioms **(A1-A15), (A8), (A12)-(A13)**.Here the value of the `fuzzyroughType` is `axiom`. It also contains additional tag `Degree` with an attribute `value (0, 1].`

Annotation Syntax

```
<fuzzyroughOwl2 fuzzyroughType="axiom">
<Degree value="<double>"/>
</fuzzyroughOwl2>
```

4 Application

4.1 Query Refinement Example

In this part, we provide a query refinement example to show, how to use fuzzy rough ontologies. *Let S* be a fuzzy rough equivalence relation among all query terms. *S* calculates the frequency degree of pair of terms. Furthermore, *S* is expressed in the following Table 3.

Table 1. Fuzzy role assertions

S	**Store**	**vehicle**	**Hero**	**film**
Store	1	0.89	0	0.46
Vehicle		1	1	0
Hero			1	0.94
Film				1

Formally, lower bound always specifies a truth degree. For example, the third cell in Table 1 represents the fuzzy role assertion $\langle (hero, film: s) \geq 0.94 \rangle$. If the degree is 0, it specifies an exact bound. For example, the third cell represents $\langle (store, hero):s=0 \rangle$. Let us consider an example, suppose that we are looking for *film hero.* Hence, we are interested in documents that have to do with all the terms *film and hero.* It is important to note that hero is an ambiguous word, which refers a company name, vehicle name and an actor in a film. Once the fuzzy Knowledgebase K, can be defined, the relevance term **t** to a query can be computed using the fuzzy rough constructors as follows:

glb (K, t : (S↓↑Q))

For example, even if glb ⟨(K,, store: Q)⟩=0, it holds that

glb (K,, store(S↓↑Q)) = 0.45. In fuzzy rough *OWL2* annotations, it can be expressed as

```
<AnnotationAssertion>
<AnnotationProperty IRI='#fuzzyroughLabel'/>
<IRI>#indeedC</IRI>
<Literal datatypeIRI='&rdf;PlainLiteral'>
<fuzzyroughOwl2 fuzzyroughType="concept">
<concept type="TightUpperApproximated" modifier="indeed" base="Q" />
</fuzzyroughOwl2>
</Literal>
```

</ AnnotationAssertoin>
That is, the query refinement process makes the store relevant to the query to degree 0.45. The Table 2 shows the result that we obtain with the original query and with the query refinement process based on tight upper approximation. Additionally, we add weights to the query by using a fuzzy rough extension of the *OWL2* discussed in this paper.

5 IMPLEMENTATION AND EXPERIMENTAL EVALUATION

5.1 OntoSri Plug-in

Ramakrishnan and Vijayan [26] present a study on application of different kinds of protégé visualization tools and categorize their characteristics and features. They also identify some requirements for ontology visualization tools offering cognitive assistance and presents solutions with simple knowledge representations [27]. *OntoSri* is an advanced visualization ontology tool for knowledge representation. *OntoSri* tool [25] is designed and implemented by the research group of department of Computer Science, AVVM Sri Pushpam College, Poondi, affiliated to Bharathidasan University, with the financial support of University Grants Commission, New Delhi, India, under the scheme of major research project. *OntoSri plug-in* facilitates encoding of vagueness into standard *OWL2* ontology by means of annotations.

5.2 Implementation

In this article, we proposed a methodology for fuzzy rough ontology development. At first, we construct the core part of the ontology by using an *OntoSri* ontology editor. This permits reasoning with core part using standard ontology reasoners. Then, we add the fuzzy rough part of the ontology by using annotation properties. Typing annotations is an error-prone and critical task. So, we create the *OntoSri plug-in* named by *FuzzyRough* to handle the annotations makes easier to the users. Once the plug-in is installed, a new tab *FuzzyRough* enables to use the plug-in. The plug-in contains menus with the available options, which corresponds to the possibilities described in section 3 and are as shown in Fig. 1. Furthermore, the plug-in is integrated with fuzzy DL reasoner and makes it possible to submit queries to it. After the creation of the fuzzy rough ontology, it should be translated into the language supported by some fuzzy DL reasoners. For this purpose, we have developed a parser that translates fuzzy rough ontologies with annotation of type fuzzyroughLabel into the languages supported by some fuzzy DL reasoners. This general parser can be adapted to any particular fuzzy DL reasoner.

Table 2. Results of query refinement

	Q	**(S↓↑Q)**
Store	0	0.45
Vehicle	1	1
Hero	1	1
Film	0	0.83

5.3 Features of *FuzzyRough* Plug-in

- The main advantage of our approach compared to fuzzy ontology is that we can apply concept modifiers such as *exceedingly, certainly, very, somewhat, relatively, nearly, indeed, veryvery* using annotation properties for complex concepts.

Other Advantages:

- The user can carry out fuzzy rough knowledge base consistency and fuzzy rough concept satisfiability and computation of the maximum degree of satisfiability of a fuzzy rough concept tasks in terms approximation values such as *upper, lower, tight upper, tight lower, loose upper, loose lower*.
- The user can perform fuzzy rough concept subsumption.
- *FuzzyRough* plug-in doesn't need any extra information about data like probability in statistics.
- *FuzzyRough* plug-in not only represents query terms and their relations, but also some background knowledge related to the domain.

For an illustrative purpose, it has adapted to the languages supported by fuzzy DL reasoners such as *fuzzy DL* & *DeLorean*.

5.4 Experimental Evaluation

We consider a small ontology query refinement in information retrieval. Our experimental result shows that query refinement ontology has 12 annotations: 8 fuzzy rough data types (out of 15 data types) and 3 fuzzy rough concepts (out of 200 concepts). We got the parsing time of the original query refinement is as 191.2 ms, whereas the parsing time of the annotated query refinement is 181.2 ms. The experimental result shows the feasibility of our work.

6 USER STUDY

A large user study was conducted to evaluate users' satisfaction on Fuzzy rough ontology using *OWL2* annotation properties implemented by *FuzzyRough* plug-in. As a part of our research work, we created *Fuzzyrough* plug-in, an effective knowledge representation part of our *OntoSri* tool to represent fuzzy rough information in an efficient manner. The user study described in this work was designed in order to present useful insight regarding three research areas.
They are as

- The evaluation of *FuzzyRough* plug-in created by our research work.
- The techniques and strategies employed by the users while researching Fuzzy rough information.
- The evaluation of the Fuzzy rough ontology represented by our research work.

The focus of this study was not overall fuzzy rough ontology management, but rather fuzzy rough ontology representation and assessing the steadiness of *FuzzyRough* plug-in for user applications where fuzzy rough ontologies are used as browsing assisters. This section describes brief descriptions of the evaluation user group, the fuzzy rough ontology used and the results.

6.1 Test user group

A group of user with both basic domain knowledge and computer skills was chosen to inspect the efficiency of both the *Fuzzy* and *FuzzyRough* plug-in. The option of ontology was such that all the users could have some knowledge with computers and domain. This reality ensured that there would not be major differences in the performance of the users due to complete lack of knowledge of the domain. Most of the users that participated to the user study were research scholars of Computer Science and Information Science departments of Sri Pushpam College of Bharathidasan University, India. User group details are as shown in Table 3. Furthermore, all these users should have some familiarity in computers, fuzzy ontology and fuzzy rough ontology as they are instructed to be well-known for testing. But they differ both in computer and domain knowledge respectively. The user group was divided into two corresponding groups of eight users one that got a brief introduction in how to use these *FuzzyRough* plug-in, and one without any prior knowledge about Fuzzy rough ontology.

In general, *Fuzzy* plug-in of the *OntoSri* tool represents vague information whereas *FuzzyRough* plug-in of the *OntoSri* tool represents rough information combined with fuzzy. Moreover, in our user study, the users are allowed to perform both the fuzzy functions and fuzzy rough functions of the *OntoSri* tool and their results are recorded in Table 4& 5.

6.2 Testing method set-up

We had to carry out some tests in order to decide upon the *FuzzyRough* plug-in set-up to be used in the work, before the start of the evaluation. Bearing in mind that we were investigating the most appropriate *FuzzyRough* plug-in not for ontology developers but for users.

6.3 Fuzzy Rough Ontology Representation Tasks

This user study constructed five different queries to prove the effectiveness of the *FuzzyRough* plug-in. The queries are grouped into various types according to the fuzzy rough ontology criteria.

The recognized query types are presented in the following text:

1. *Does the plug-in support rough concept modifiers to complex concepts?*

In this case, the *FuzzyRough* plug-in supports rough concept modifiers such as *exceedingly, certainly, very, somewhat, veryvery, relatively, nearly, and indeed*. In contrast, the above modifiers are not applicable in the *Fuzzy* plug-in.

2. *If the user can view the sample data in the large database through the plug-in?*

The naive user can view sample data from the large database with the *FuzzyRough* plug-in, because, query terms are included in the *FuzzyRough* plug-in. But, *Fuzzy* plug-in only partially support these features.

3. *Is the plug-in require any additional information like probability in statistics?*

In this case, *FuzzyRough* plug-in doesn't require any additional information like probability in statistics when compared to *Fuzzy* plug-in.

4. *Is it also possible to get background knowledge related to particular domain along with exact information?*

Fuzzy plug-in provides exact information along with the background familiarity related to the domain.

5. *Is the user can perform a fuzzy rough concept tasks in terms of approximated values?*

The user can carry out fuzzy rough knowledge base *consistency* and fuzzy rough concept *satisfiability* and computation of the maximum degree of satisfiability of a fuzzy rough concept tasks in terms of approximation values such as *upper, lower, tight upper, tight lower, loose upper, loose lower*.

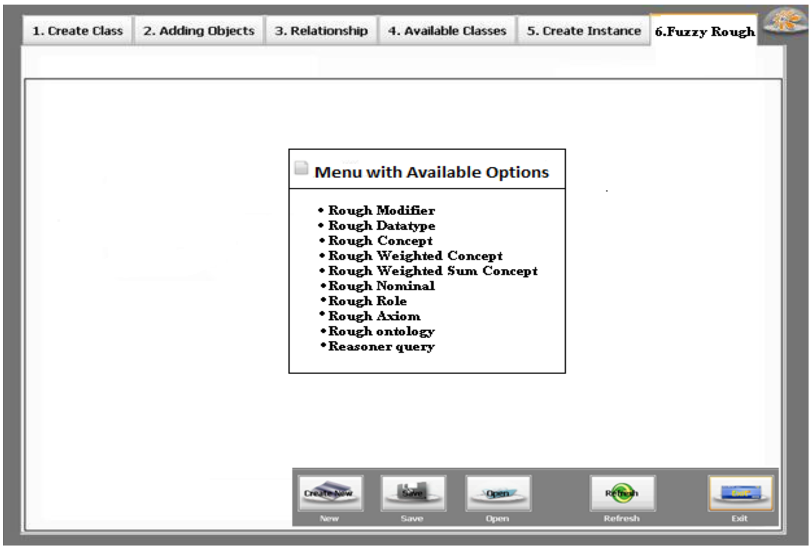

Fig. 1. OntoSri plug-in

6.4 Discussions

The results of the user study are based around the usability questionnaire responses and users' comment. The questionnaire focuses on the use of the fuzzy rough ontology and the scale runs from -2 to +2 (+2 indicates Very positive attitude, +1 – positive attitude, -1 – Negative, -2 – Very Negative and 0 – Neutrality). In addition to that, we calculate a t-test to find which plug-in is significant. User comments about fuzzy and fuzzy rough ontology representations are recorded and it graphical representation is shown in Fig. 3. Overall, users feedback (30 users) about the representation of fuzzy ontologies are rated as x = 1.10 (SD = 0.182), fuzzy rough ontologies are rated as x = 1.38 (SD = 0.150) are shown in Fig. 4 and the value of t = 13.0. The result of the t-test (t = 13.0, level of significance = 5%) shows that the users' satisfaction seems quite high in the *Fuzzy Rough* plug-in of *OntoSri* tool. Furthermore, sample of feedback of the users is listed in Table 4 and Table 5.

Table 3. User Group Details

User group	profession description	Professional group	Technical experience	Age Range
1-5	Research Scholar	Researcher, Dept of Mathematics	Moderate	25+
5-10	Junior Academic	Associate professor, Dept of Mathematics	Moderate	50+
10-15	Senior Academic	Associate professor, Dept of Computer Science	high	30+
15-20	Research Scholar	Researcher, Dept of Computer Science	high	25+
20-25	Junior Academic	Researcher, Dept of	high	30+

25-30	Research Scholar	Information Science Researcher, Dept of high Computer Science	25+

7 RELATED WORK

Questions	Users															Mean
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
Efficiency	2	1	2	2	2	1	1	0	1	2	2	1	2	1	2	1.46
Fast	1	2	1	1	2	1	2	1	1	2	2	1	1	2	1	1.40
Effectiveness	2	2	1	2	1	1	0	2	1	0	2	2	2	1	1	1.33
Useful	2	1	2	2	1	2	2	1	1	2	2	1	1	2	1	1.53
Easy to Learn	1	2	2	2	2	1	2	1	1	1	2	2	2	1	2	1.60
Clear Interaction	1	2	-1	1	2	1	1	2	1	1	1	2	1	0	1	1.13
Flexible	1	2	1	1	1	-1	0	2	1	1	2	2	1	1	2	1.13
Understa	2	1	-1	1	2	1	2	1	2	2	2	1	2	1	1	1.53

Many attempts have been made to represent various forms of uncertainty in ontology languages, but this is the tremendous effort towards the fuzzy rough ontology using *OWL2*. Cuenca-Grau et al., [11] provides fuzzy extension of the ontology using *OWL2*. However, it doesn't support the fuzzy rough concepts. But, our proposed work supports the fuzzy rough extension of the *OWL2* and also supports the fuzzy rough concepts.

Table 4. Users Satisfaction of *Fuzzy* Plug-in (OntoSri tool)

ndable																
Skill	1	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1.26
Easy to use	2	2	1	1	2	1	0	2	1	2	2	1	1	2	2	1.46

Table 5. Users Satisfaction of *FuzzyRough* Plug-in (OntoSri tool)

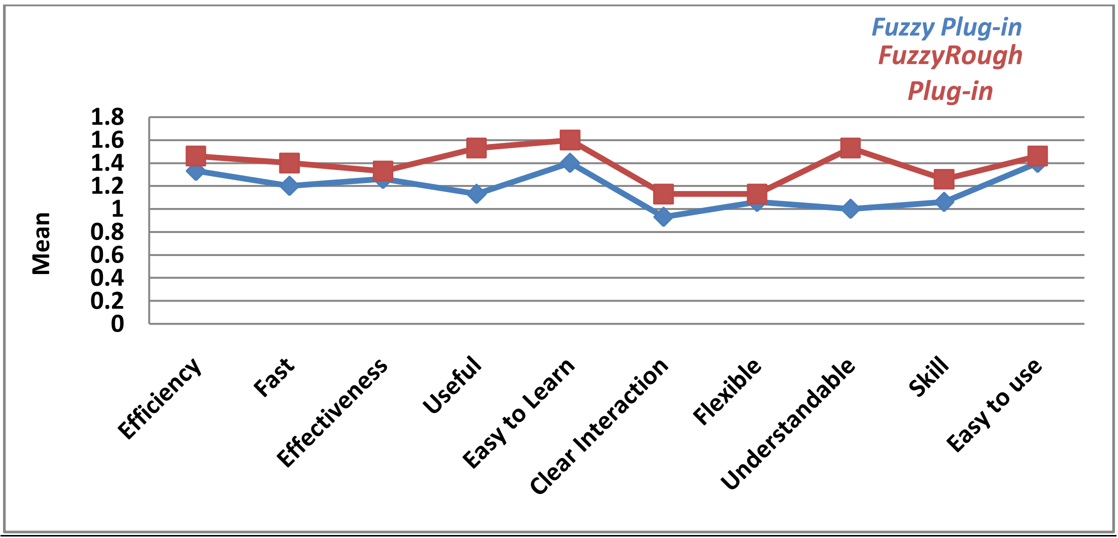

Fig. 3. Users Satisfaction Level of *Fuzzy* and *FuzzyRough* Plug-in

Questions	**Users**															**Mean**
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	
Efficiency	1	2	1	1	2	0	1	1	2	2	1	2	1	2	1	1.33
Fast	1	1	2	1	2	1	2	2	2	1	1	0	1	1	0	1.20
Effectiveness	2	1	2	1	0	1	1	2	1	0	1	2	2	1	1	1.26
Useful	2	-1	1	2	2	1	2	1	1	1	1	1	2	1	0	1.13
Easy to Learn	0	1	2	2	2	1	1	2	1	2	2	1	1	2	1	1.40
Clear Interaction	0	1	2	1	1	-1	2	2	1	1	2	1	1	1	-1	0.93
Flexible	2	1	1	1	0	-1	2	1	1	1	1	2	-1	2	1	1.06
Understandable	0	1	1	2	1	1	2	1	1	2	-1	1	2	-1	1	1.00
Skill	2	1	-1	2	1	2	-1	1	1	0	2	1	2	1	-1	1.06
Easy to use	0	1	2	2	1	0	2	1	2	2	1	1	2	2	1	1.40

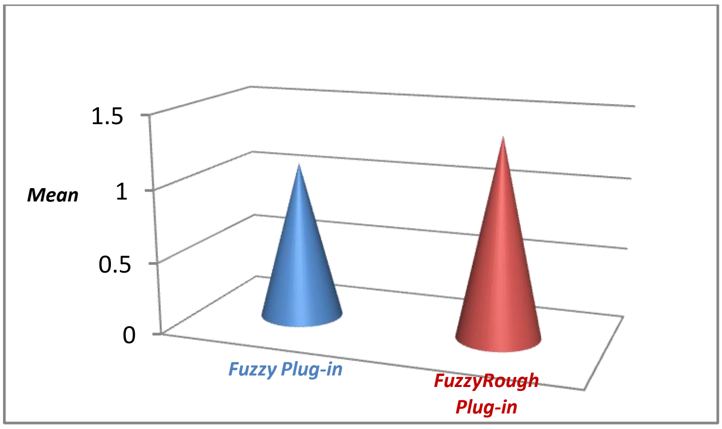

Fig. 4. Overall users' feedback about *Fuzzy* and *FuzzyRough* Plug-in

Another approach closer to our work is provided by Bobillo and Straccia [9]. In their approach, concepts, roles and axioms are represented, but annotation properties are not used. Our representation provides annotation properties to represent concepts, roles and axioms in a well defined manner. Dey ta al., [12] represents fuzzy rough ontology, but its reasoning is restricted to simple form of concept querying. Our approach however makes it possible to use any of the reasoning tasks for the fuzzy rough DLs. There exists a number of reasoners such as *FuzzyDL [5]*, *KAON2 [3]*, *ONTOSEARCH2 [31]*, etc., . They perform translation activities for obtaining an equivalent *OWL2* ontology for the given fuzzy ontology. DeLorean [6] is a recent development in FuzzyDL reasoner for the same translation activity. But, they all do not support any specific format to represent fuzzy rough ontologies that can be easily managed by current *OWL2* editors and understood by human. These reasons motivate us to do the research on offering a specific format to represent fuzzy rough ontologies.

8 CONCLUSIONS AND FUTURE WORK

In this article, we proposed a framework for the representation of the fuzzy rough ontologies using a current standard language. To begin with, we observed that the current fuzzy rough extensions of ontology languages are not expressive enough. So that, we introduced a methodology that the fuzzy rough ontology has to cope with uncertainty. In this article, we presented a very general fuzzy rough extension of the ontology language. It contains many differences with respect to *OWL2*, such as fuzzy rough data type, fuzzy rough modifiers etc. Then, we provided a fuzzy rough ontology representation using the current standard language *OWL2* with annotation properties.

Acknowledgements. *This work has been financed by University Grants Commission Major Research Project – Ref. No. 38-4/2009(SR) (India).*

References

1. Abulaish M, Dey L : "Interoperability among distributed overlapping ontologies – a fuzzy ontology framework", Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, Hong Kong, China, 2006.
2. Abulaish M, Dey L : "A fuzzy ontology generation framework for handling uncertainties and non uniformity in domain knowledge description", In: Proceedings of 2007 International Conference on Computing: Theory and Applications. Kolkata 2007; 287-293.

3. Agarwal S, Hitzler P : “sMart – A semantic matchmaking portal for electronic markets”, In : Proceedings of the 7th international IEEE conference on E-commerce Technology, IEEE Computer Society, 2005.
4. Alexopoulos P, Wallace M, Kafentzis K, Panos AD : “Utilizing imprecise knowledge in ontology-based CBR systems by means of fuzzy algebra”, International Journal of Fuzzy Systems 2010 ; 12(1) : 1-14.
5. Bobillo F, Straccia U : “ FuzzyDL: An expressive fuzzy description logic reasoned”, In Fuzzy systems, 2008. FUZZ – IEEE, IEEE World Congress on Computational Intelligence 2008; 923-930.
6. Bobillo F, Straccia U : “An OWL ontology for fuzzy OWL 2”, In: Proceedings of the 18th International Symposium on Methodologies for Intelligent Systems (ISMIS 2009), Lecture notes in computer science vol. 5722, Springer-Verlag 2009 ; 151–160.
7. Bobillo F, Straccia U : “Generalized fuzzy rough description logics”, Information Sciences 2012; 189:43-62.
8. Bukhari AC, Kim Y : “Integration of a secure type-2 fuzzy ontology with a multi-agent platform : A proposal to automate the personalized flight ticket booking domain”, Information Sciences 2012; 24–47.
9. Calegari S, and Davide C : “A fuzzy ontology-approach to improve semantic information retrieval”, In : Proceedings of the Third ISWC Work-Shop on Uncertainty Reasoning for the Semantic Web, Busan, Korea, 2007.
10. Calegari S, Ciucci D : “Fuzzy ontology, fuzzy description logics and fuzzy-OWL”, In Applications of Fuzzy Sets Theory, Springer Berlin Heidelberg 2007; 118-126.
11. Cuenca-Grau B, Horrocks I, Motik I, Parsia B, Patel-Schneider P.F, Sattler U : “OWL 2: the next step for OWL”, Journal of Web Semantics 2008; 6 (4) : 309–322.
12. Dey L, Abulaish M, Goyal R, Shubham K : “A rough-fuzzy ontology generation framework and its application to bio-medical textprocessing”, In: Proceedings of the 5th Atlantic Web Intelligence Conference (AWIC 2007), Advances in Soft Computing, vol. 43, Springer-Verlag, 2007; 74–79.
13. Dwi HW, John Y : “A fuzzy ontology-based abstract search engine and its user studies”, In : Proceedings of 10th IEEE International Conference on Fuzzy Systems, Melbourne, Australia, 2001.
14. Harshit K, Seong P : “Know-Ont: A Knowledge Ontology for an Enterprise in anIndustrial Domain”, International journal of database theory and application 3(1) : 23-32. (2010)
15. Huang HD, Acampora G, Loia V, Lee VS, Kao HY : “Applying FML and fuzzy ontologies to malware behavioural analysis”, IEEE International Conference on Fuzzy Systems, Taipei, Taiwan, 2011.
16. Inyaem U, Meesad P, Haruechaiyasak C, Tran D : “Construction of fuzzy ontology-based terrorism event extraction”, In Knowledge Discovery and Data Mining. WKDD'10, Third international conference on IEEE 2010, 391-394.
17. Ishizu S, Gehrmann A, Nagai Y, Inukai Y : “Rough ontology: Extension of ontologies by rough sets”, Symposium on Human Interface held as Part of HCI International, Beijing, China, 2007; doi 10.1007/978-3-540-73345-4_52, 456-462.
18. Klinov P, Mazlack LJ : “Fuzzy rough approach to handling imprecision in semantic web ontologies”, fuzzy Information Processing Society, Nafips Annual meeting of the north american. IEEE 2006; 142 – 147.

19. Lee CS, Wang MH, Hagras H : “A type2 fuzzy ontology and its application to personal diabetic – diet recommendation”, IEEE Transactions on Fuzzy Systems 2010; 18 : 374–395.
20. Li R, Sun X, Lu Z, Wen K, Li Y : “Towards a Type-2 Fuzzy Description Logic for Semantic Search Engine”, Advances in Data and Web Management, Lecture Notes in Computer Science 2007; 805-812.
21. Mendel JM, John RI, Liu F : “Interval type-2 fuzzy systems made simple”, IEEE transactions on Logic Fuzzy Systems 2006; 808-821.
22. Mezei J, Wikstrom R : "OWAD operators in Type-2 fuzzy ontologies", IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS) 2013; 848 – 853. doi: 10.1109/IFSA-NAFIPS.2013.6608511
23. Mohd Kamir Y, Mohd Nordin Abdul R, Mat Atar Mat : " Ontology And Semantic Web Approaches For Heterogeneous Database Access”, An International Journal of Database Theory and Application 2011; 4(4) : 13-23.
24. Peña L : “Identity, fuzziness and noncontradiction”, Noûs 1984; 18: 227-259.
25. Ramakrishnan S, Vijayan A : “A study on development of Cognitive support features in recent Ontology Visualization Tools”, Springer, 2014, 4(41), 595-623.
26. Ramakrishnan S, Vijayan A : “Ontology visualization Protégé Tools – A Review”, International Journal of Advanced Information Technology (IJAIT) 2011,Vol. 1, No. 4.
27. Ramakrishnan S, Vijayan A : “Reducing the Complexities in the Cognition of Ontology Knowledge Representation”, International Journal of Information Sciences and Techniques (IJIST) 2011; Vol.1, No.3.
28. Sánchez E, Tettamanzi AGB : “Fuzzy quantification in fuzzy description logics”, In: E. Sanchez (Ed.), Fuzzy Logic and the Semantic Web, Capturing intelligence, Elsevier Science, Heidelberg, Germany 2006; 135–159.
29. Sanchez E, Yamanoi T : “Fuzzy ontologies for the semantic web”, In : Flexible Query Answering Systems, Springer Berlin Heidelberg 2006; 691-699.
30. Soomi Y, Harshit K, Seong P : “An Expert System for Wide Area Surveillance based on Ontology”, International Journal of Database Theory and Application 2012; 5(2) : 1-16.
31. Thomas E, Pan J.Z, Sleeman D : “ONTOSEARCH2: Searching ontologies semantically”, In Proceedings of the 3rd International Workshop on OWL: experiences and directions, CEUR workshop proceedings, 2007.
32. Wang M, Lee C, Chen Z, Hagras H, Kuo S, Kuo H, Cheng H : “A Type-2 FML-Based Fuzzy Ontology for Dietary Assessment”, On the Power of Fuzzy Markup Language Studies in Fuzziness and Soft Computing, Springer 2013; Vol. 296, 149-168.
33. Zadeh L.A: “ Fuzzy sets”, Information and Control 1965; 8: 338-353.