Storing and Querying Fuzzy Knowledge in the Semantic Web

Nick Simou, Giorgos Stoilos, Vassilis Tzouvaras, Giorgos Stamou, and Stefanos Kollias

Department of Electrical and Computer Engineering,
National Technical University of Athens,
Zographou 15780, Greece
{nsimou,gstoil,tzouvaras,gstam}@image.ntua.gr

Abstract. The great evolution of ontologies during the last decade, bred the need for storage and querying for the Semantic Web. For that purpose, many RDF tools capable of storing a knowledge base, and also performing queries on it, were constructed. Recently, fuzzy extensions to description logics have gained considerable attention especially for the purposes of handling vague information in many applications. In this paper we investigate on the issue of using classical RDF storing systems in order to provide persistent storing and querying over large-scale fuzzy information. To accomplish this we first propose a novel way for serializing fuzzy information into RDF triples thus classical storing systems can be used without any extensions. Additionally, we extend the existing query languages of RDF stores in order to support expressive fuzzy queries proposed in the literature. These extensions are implemented through the FiRE fuzzy reasoning engine, which is a fuzzy DL reasoner for fuzzy-$\mathcal{SHIN}$. Finally, the proposed architecture is evaluated using an industrial application scenario about casting for TV commercials and spots.

1 Introduction

Ontologies, through the OWL language [11], are expected to play a significant role in the Semantic Web. OWL is mainly based on Description Logics (DLs) [2], a popular family of knowledge representation languages. However, despite their rich expressiveness, they are insufficient to deal with vague and uncertain information which is commonly found in many real-world applications such as multimedia content, medical informatics etc. For that purpose a variety of DLs capable of handling imprecise information in many flavors, like probabilistic [13] and fuzzy [15, 14] have been proposed.

Fuzzy ontologies are envisioned to be very useful in the Semantic Web. Similar to crisp ontologies, they can serve as basic semantic infrastructure, providing shared understanding of certain domains across different applications. Furthermore, the need for handling fuzzy and uncertain information is crucial to the Web. This is because information and data along the Web may often be uncertain or imperfect.

Therefore sophisticated uncertainty representation and reasoning are necessary for the alignment and integration of Web data from different sources. This requirement for uncertainty representation has led W3C to set up the Uncertainty Reasoning for the World Wide Web XG[1]. Recently, fuzzy DL reasoners such as fuzzyDL[2] and FiRE[3] that can handle imprecise information have been implemented. Despite these implementations of expressive fuzzy DLs there is still no other work on persistent storage and querying, besides the work of Straccia [16] and Pan [10], which are based on fuzzy DL-lite and can be considered as closely related to databases, but on the other hand they don't use RDF triple store technologies.

The main contributions of this paper are the following:

1. It presents a novel framework for persistent storage and querying of expressive fuzzy knowledge bases,
2. It presents the first ever integration of fuzzy DL reasoners with RDF triple stores, and
3. It provides experimental evaluation of the proposed architecture using a real-world industrial strength use-case scenario.

The rest of the paper is organized as follows. Firstly, in section 2 a short theoretical description of the fuzzy DL f-$\mathcal{SHIN}$ [6] is made. In section 3 the proposed triples syntax accommodating the fuzzy element used for storing a fuzzy knowledge base in RDF-Stores, is presented. Additionally, the syntax and the semantics of expressive queries that have been proposed in the literature [10] to exploit fuzziness are briefly presented. In the following section (4) the fuzzy reasoning engine FiRE which is based on the fuzzy DL f-$\mathcal{SHIN}$ and the way in which it was integrated with the RDF-Store Sesame are presented. In the last section (5) the applicability of the proposed architecture is demonstrated, presenting a use case based on a database of human models. This database was used by a production company for the purposes of casting for TV commercials and spots. Some entries of the database were first fuzzified and then using an expressive knowledge base, abundant implicit knowledge was extracted. The extracted knowledge was stored to a Sesame repository, and various expressive queries were performed in order to benchmark the proposed architecture.

2 Preliminaries

2.1 The Fuzzy DL f$_{KD}$-$\mathcal{SHIN}$

In this section we briefly present the notation of DL f-$\mathcal{SHIN}$ which is a fuzzy extension of DL $\mathcal{SHIN}$ [7]. Similar to crisp description logic languages, a fuzzy description logic language consist of an alphabet of distinct concepts names (**C**), role names (**R**) and individual names (**I**), together with a set of constructors to

[1] `http://www.w3.org/2005/Incubator/urw3/`

[2] `http://gaia.isti.cnr.it/~straccia/software/fuzzyDL/fuzzyDL.html`

[3] `http://www.image.ece.ntua.gr/~nsimou/FiRE/`

construct concept and role descriptions. If R is a role then R^- is also a role, namely the inverse of R. f-$\mathcal{SHIN}$-concepts are inductively defined as follows,

1. If $C \in \mathbf{C}$, then C is a f-$\mathcal{SHIN}$-concept,
2. If C and D are concepts, R is a role, S is a simple role and $n \in \mathbb{N}$, then $(\neg C)$, $(C \sqcup D)$, $(C \sqcap D)$, $(\forall R.C)$, $(\exists R.C)$, $(\geq nS)$ and $(\leq nS)$ are also f-$\mathcal{SHIN}$-concepts.

In contrast to crisp DLs, the semantics of fuzzy DLs are provided by a *fuzzy interpretation* [15]. A fuzzy interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ where $\Delta^{\mathcal{I}}$ is a non-empty set of objects and $\cdot^{\mathcal{I}}$ is a fuzzy interpretation function, which maps an individual name a to elements of $\mathsf{a}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ and a concept name A (role name R) to a membership function $\mathsf{A}^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0,1]$ ($R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1]$).

By using fuzzy set theoretic operations the fuzzy interpretation function can be extended to give semantics to complex concepts, roles and axioms [8]. FiRE uses the standard fuzzy operators of $1 - x$ for fuzzy negation and max, min for fuzzy union and intersection, respectively.

A f-$\mathcal{SHIN}$ knowledge base Σ is a triple $\langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{T}$ is a fuzzy $TBox$, $\mathcal{R}$ is a fuzzy $RBox$ and $\mathcal{A}$ is a fuzzy $ABox$. $TBox$ is a finite set of fuzzy concept axioms which are of the form $C \sqsubseteq D$ called fuzzy concept inclusion axioms and $C \equiv D$ called fuzzy concept equivalence axioms, where C, D are concepts, saying that C is a sub-concept or C is equivalent of D, respectively. Similarly, $RBox$ is a finite set of fuzzy role axioms of the form $\mathsf{Trans}(R)$ called fuzzy transitive role axioms and $R \sqsubseteq S$ called fuzzy role inclusion axioms saying that R is transitive and R is a sub-role of S respectively. Finally, $ABox$ is a finite set of fuzzy assertions of the form $\langle a : C \bowtie n \rangle$, $\langle (a,b) : R \bowtie n \rangle$, where $\bowtie$ stands for $\geq, >, \leq$ or $<$, or $a \not\doteq b$, for $a, b \in \mathbf{I}$. Intuitively, a fuzzy assertion of the form $\langle a : C \geq n \rangle$ means that the membership degree of a to the concept C is at least equal to n.

3 Storing and Querying a Fuzzy Knowledge Base

3.1 Fuzzy OWL Syntax in triples

In order to use the existing RDF storing systems to store fuzzy knowledge without enforcing any extensions we have to provide a way of serializing fuzzy knowledge into RDF triples. Some work has already been done in this issue. In [9] the authors use RDF *reification*, in order to store membership degrees, while the authors in [17] use datatypes. Our goal is to neither use reification nor datatypes. On the one hand, it is well-known that reification has weak, ill-defined model theoretic semantics and its support by RDF tools is doubtful while on the other hand, we do not want to use a concrete feature like datatypes to represent abstract information such as fuzzy assertions. For those reasons we propose a more clarified way based on the use of blank nodes. First, we define three new entities, namely `frdf:membership`, `frdf:degree` and `frdf:ineqType` as types (i.e. `rdf:type`) of `rdf:Property`.

Our syntax becomes obvious in the following example. Suppose that we want to represent the assertion $\langle (paul : Tall) \geq n \rangle$. The RDF triples representing this information are the following:

```
paul                 frdf:membership    _:paulmembTall .
_:paulmembTall       rdf:type           Tall .
_:paulmembTall       frdf:degree        “n^^xsd:float” .
_:paulmembTall       frdf:ineqType      “=” .
```

where `_:paulmembPaul` is a blank node used to represent the fuzzy assertion of `paul` with the concept Tall.

On the other hand, mapping fuzzy role assertions is more tricky since RDF does not allow the use of blank nodes in the predicate position. Thus, we have to use new properties for each assertion. Thus, the assertion $\langle (paul, frank) : FriendOf \geq n \rangle$ is mapped to

```
paul                     frdf:paulFriendOffrank   frank .
frdf:paulFriendOffrank   rdf:type                 FriendOf .
frdf:paulFriendOffrank   frdf:degree              “n^^xsd:float” .
frdf:paulFriendOffrank   frdf:ineqType            “=” .
```

3.2 Extensions to Query Languages

One of the main advantages of persistent storage systems, like relational databases and RDF storing systems, is their ability to support *conjunctive queries.* Conjunctive queries generalize the classical inference problem of *realization* of Description Logics [2], i.e. get me all individuals of a given concept C, by allowing for the combination (conjunction) of concepts and roles. Formally, a conjunctive query is of the following form:

$$q(X) \leftarrow \exists Y.conj(X, Y) \tag{1}$$

or simply $q(X) \leftarrow conj(X, Y)$, where $q(X)$ is called the head, $conj(X, Y)$ is called the body, X are called the *distinguished variables*, Y are existentially quantified variables called the *non-distinguished variables*, and $conj(X, Y)$ is a conjunction of atoms of the form $\mathsf{A}(v)$, $R(v_1, v_2)$, where A, R are respectively concept and role names, v, v_1 and v_2 are *individual* variables in X and Y or individuals from the ontology.

Since in our case we extend classical assertions to fuzzy assertions, new methods of querying such fuzzy information are possible. More precisely, in [10] the authors extend ordinary conjunctive queries to a family of significantly more expressive query languages, which are borrowed from the fields of fuzzy information retrieval [5]. These languages exploit the membership degrees of fuzzy assertions by introducing weights or thresholds in query atoms. In particular, the authors first define *conjunctive threshold queries*(CTQs) as:

$$q(X) \leftarrow \exists Y. \bigwedge_{i=1}^{n} (atom_i(X, Y) \geq k_i), \tag{2}$$

where $k_i \in [0,1]$, $1 \leq i \leq n$, $atom_i(X,Y)$ represents either a fuzzy-DL concept or role and all $k_i \in (0,1]$ are thresholds. Intuitively, an evaluation $[X \mapsto S]$ (where S is a set of individuals) is a solution if $atom_i^{\mathcal{I}}(X,Y)_{[X\mapsto S, Y\mapsto S']} \geq k_i$ for some S and for $1 \leq i \leq n$. As it is obvious answers of CTQs is a matter of true or false, in other words an evaluation either is or is not a solution to a query. The authors also propose *General Fuzzy Conjunctive Queries* (GFCQs) that further exploit fuzziness and support degrees in query results. The syntax of a GFCQ is the following:

$$q(X) \leftarrow \exists Y. \bigwedge_{i=1}^{n} (atom_i(X,Y) : k_i), \tag{3}$$

where $atom_i(X,Y)$ and k_i are as above. As it is shown in [10], this syntax is general enough to allow various choices of semantics, which emerge by interpreting differently the association of the degree of each fuzzy-DL atom $(atom_i(X,Y))$ with the degree associated weight (k_i). For example if this association is interpreted by a fuzzy implication $(\mathcal{J})$ [8] then we obtain fuzzy threshold queries:

$$d = \sup_{S' \in \Delta^{\mathcal{I}} \times \ldots \times \Delta^{\mathcal{I}}} \{ t_{i=1}^{n} \ \mathcal{J}(k_i, atom_i^{\mathcal{I}}(\bar{v})_{[X\mapsto S, Y\mapsto S']}) \}.$$

Similarly we can use fuzzy aggregation functions [8] or fuzzy weighted t-norms [4]. Variations of semantics of GFCQs can be effectively used to model importance of query atoms, preferences, and many more. The interested reader is referred to [10] for more details on the semantics of GFCQs.

4 Implementation with FiRE and Sesame

FiRE is a JAVA implementation of a fuzzy reasoning engine for imperfect knowledge currently supporting f-$\mathcal{SHIN}$. It can be found at `http://www.image.ece.ntua.gr/~nsimou/FiRE/` together with installation instructions and examples. Its syntax is based the Knowledge Representation System Specification [1] proposal which has been extended to fit uncertain knowledge. In this section the inference services of FiRE are presented and the way in which it was integrated with RDF Store Sesame [4] in order to support CTQs and GFCQs is demonstrated.

4.1 Inference services

Crisp DL reasoners offer reasoning services such as deciding satisfiability, subsumption and entailment of concepts and axioms w.r.t. an ontology. In other words, these tools are capable of answering queries like "Can the concept C have any instances in models of the ontology T?" (satisfiability of C), "Is the concept D more general than the concept C in models of the ontology T ?" (subsumption $C \sqsubseteq D$) or does axiom Ψ logically follows from the ontology (entailment of Ψ).

[4] `http://www.openrdf.org/`

These reasoning services are also available by f-$\mathcal{SHIN}$ together with *greatest lower bound queries* which are specific to fuzzy assertions. FiRE uses the tableau algorithm of f-$\mathcal{SHIN}$, presented by Stoilos et al [6], in order to decide the key inference problems of a fuzzy ontology. In the case of fuzzy DL, satisfiability queries are of the form "Can the concept C have any instances with degree of participation $\bowtie n$ in models of the ontology T ?". Furthermore, it is in our interest to compute the best lower and upper truth-value bounds of a fuzzy assertion. The term *greatest lower bound* of a fuzzy assertion w.r.t. Σ was defined in [15]. Roughly speaking, greatest lower bound are queries like "What is the greatest degree n that our ontology entails an individual a to participate in a concept C?". Entailment queries ask whether our knowledge base logically entails the membership of an individual to a specific concept to a certain degree.

Finally, FiRE allows the user to make greatest lower bound queries (GLB). GLB queries are evaluated by FiRE performing entailment queries of the individual participating in concept of interest for all the degrees contained in the ABox, using the binary search algorithm in order to reduce the degrees search space [15]. Furthermore a user can perform global GLB for a fuzzy knowledge base. Global GLB service of FiRE, creates a file containing the greatest lower bound degree of all the concepts of Σ participating in all the individuals of Σ.

4.2 Sesame Integration with FiRE

FiRE was enhanced by the functionalities of the RDF-Store Sesame (Sesame 2 beta 6). The RDF Store is used as a back-end for storing and querying RDF triples in a sufficient and convenient way. In this architecture the reasoner is the front-end which the user can use in order to store and query a fuzzy knowledge base. Additionally, a user is able to access data from a repository, apply any of the available reasoning services on this data and then store the implicit knowledge extracted from them back in the repository.

Another important benefit from this integration is the use of the query language SPARQL [12] in the implementation of the fuzzy queries of section 3. These queries are performed using the *Queries* inference tab of FiRE, and in the case of generalized fuzzy conjunctive queries, users can choose among semantics, such as fuzzy threshold queries, fuzzy aggregation and fuzzy weighted queries.

Example 1. A threshold query that reveals their syntax in FiRE follows:

```
x,y <- Tall(x) >= 0.8 ^ has-friend(x,y) >= 0.4 ^ Short(y) >= 0.7
```

Queries consist of two parts: the first one specifies the individuals that will be evaluated while the second one states the condition that has to be fulfilled for the individuals. This query asks for individuals x and y, x has to participate in concept Tall to at least the given degree, it also has to be the subject of a has-friend assertion with participation greater than 0.4, having as a role-filler individual y which has to participate in concept Short to at least the given degrees.

Example 2. We can issue a GFCQ by using the symbol ":" followed by the importance of participation for each condition statement instead of inequality types. Hence we can get all female models and rank those who have long hair higher than those who are good-looking:

```
x <- Female(x) : 1 ^ Goodlooking(x) : 0.6
       ^ has-hairLength(x,y) : 1 ^ Long(y) : 0.8
```

In case of CTQs, a query is firstly converted from the FiRE conjunctive query syntax to SPARQL query language. Based on the fuzzy OWL syntax in triples that we have defined in section 3.1 the query of **Example** 1 is as follows in SPARQL. The query results are evaluated by Sesame engine and visualized by FiRE.

```
SELECT ?x WHERE {
    ?x ns5:membership ?Node1 .
    ?Node1 rdf:type ?Concept1 .
    ?Node1 ns5:ineqType ?IneqType1 .
    ?Node1 ns5:degree ?Degree1 .
    FILTER regex (?Concept1 , "CONCEPTS#Tall")
    FILTER regex (?IneqType1 ,">")
    FILTER (?Degree1 >= "0.8^^xsd:float")

    ?BlankRole2 ns5:ineqType ?IneqType2 .
    ?BlankRole2 ns5:degree ?Degree2 .
    ?BlankRole2 rdf:type ?Role2 .
    ?x BlankRole2 ?y .
    FILTER regex (?Role2 , "ROLES#has-friend")
    FILTER regex (?IneqType1 ,">")
    FILTER (?Degree2 >= "1.0^^xsd:float")
    ...
}
```

In case of general fuzzy conjuctive queries the operation is different. The SPARQL query is constructed in a way that retrieves the participation degrees of every Role or Concept used in the atoms criterions, for the results that satisfy all of the atoms. The participation degrees retrieved for each query atom are then used together with the degree associated weight by FiRE for the ranking procedure of the results according to the selected semantics.

It is worth mentioning that the proposed architecture obviously does not provide a complete query answering system for f-$\mathcal{SHIN}$ since queries are issued against the stored assertions of the RDF repository. Hence queries that include, for example, transitive or inverse roles are not correctly evaluated. On the one hand, query answering for fuzzy-DLs is still an open problem even for inexpressive fuzzy-DLs while on the other hand, even for classical DLs it is known that the algorithms are highly complex [3] and no practically scalable system is known. However, in order to limit the effects of incompleteness, the f-$\mathcal{SHIN}$

expressibility is used by FiRE for the extraction of implicit knowledge that is stored in the repository performing GLB tests.

5 Evaluation

5.1 Models use case

In this section we present the use case of human models utilized for the evaluation of our proposal. The data were taken from a production company database containing 2140 human models. The database contained information on each model regarding their height, age, body type, fitness type, tooth condition, eye-condition and color, hair quality, style, color and length, ending with the hands' condition. Apart from the above, there were some additional, special-appearance characteristics for certain models such as good-looking, sexy, smile, sporty, etc. introduced by the casting producer. Finally for a minority of models, a casting-video was stored in the database. The main objective of the production company was to pick a model, based on the above features, who would be suitable for a certain commercial spot. Furthermore, depending on the spot type, inquiries about models with some profession-like characteristics (like teacher, chef, mafia etc.) were also of interest.

Despite the fact that the database information on each model was rich enough, there was great difficulty in querying models of appropriate characteristics. The main reason for that was that the database information was not semantically organized. The various tables of a database made the searching for combined characteristics antiliturgical. Additionally, retrieval of models based on threshold criteria for their age or height was most of the times inaccurate since this kind of information is clearly fuzzy. Hence, selection was restricted among models that had videotaped castings or those that had worked with the producers in previous spots, thus not taking advantage of their database information.

In order to eliminate these limitations we have implemented a fuzzy knowledge base using f-$\mathcal{SHIN}$. For the generation of the fuzzy *ABox* the characteristics given by numerical values being the height and age, were fuzzified defining new concepts, while the remaining characteristics were used as crisp assertions. Therefore, the fuzzification process of age was made by setting fuzzy partitions depending on age by defining the concepts Kid, Teen, Baby, 20s, 30s, 40s, 50s, 60s and Old. Hence, as can be observed from the age fuzzification graph, a model who is 29 years old participates in both concepts 20s and 30s with degrees 0.35 and 0.65 respectively. Similarly for the fuzzification process of height, the concepts Very_Short, Short, Normal_Height, Tall and Very_Tall were defined. In the case of the height characteristic, the fuzzy partition used for female models was different from the one used for males, since the average height of females is lower than that of males. The fuzzification graphs of age and men's height are shown in **Figure** 1. An example of the produced assertions is shown in *Example* 3.

Example 3. An excerpt of ABox for the model michalis1539 is

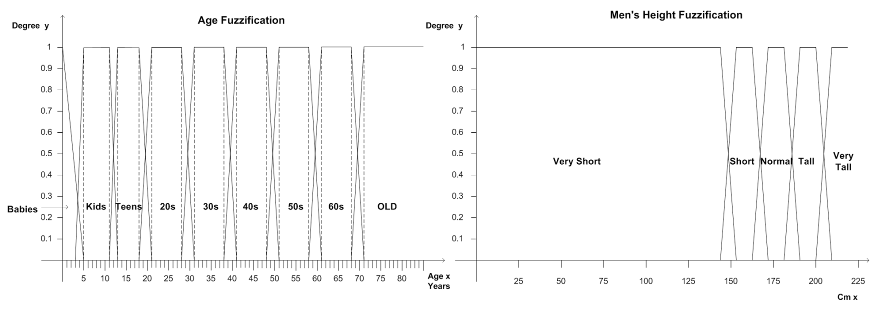

Fig. 1. Fuzzification graphs

$$\langle michalis1539 : 20s \geq 0.66\rangle,\ \langle michalis1539 : 30s \geq 0.33\rangle$$
$$\langle michalis1539 : Normal\_Height \geq 0.5\rangle$$
$$\langle michalis1539 : Tall \geq 0.5\rangle,\ \langle michalis1539 : GoodLooking \geq 1\rangle$$
$$\langle (michalis1539, good) : has-toothCondition \geq 1\rangle,\ \langle good : Good \geq 1\rangle$$

5.2 The fuzzy knowledge base

In order to permit knowledge-based retrieval of human models we have implemented an expressive terminology for a fuzzy knowledge base. The alphabet of concepts used for the fuzzy knowledge base consists of the features described above while some characteristics like hair length, hair condition etc. were represented by the use of roles.

The effective extraction of implicit knowledge from the explicit one requires an expressive terminology capable of defining higher concepts. In our case the higher domain concepts defined for human models lie into five categories: age, height, family, some special categories and the professions. Hence, the profession Scientist has been defined as male, between their 50s or 60s, with classic appearance who also wears glasses. In a similar way we have defined 33 domain concepts; an excerpt of the $TBox$ can be found in Table 1.

At this point we must mention the fact that the proposed fuzzy knowledge base does not fully utilize the expressivity of f-$\mathcal{SHIN}$. This restriction is due to the application domain (i.e transitive and inverse roles or number restrictions are not applicable in this domain), but nevertheless it is more expressive that an fuzzy DL-Lite ontology.

5.3 Results

All the experiments were conducted under Windows XP on a Pentium 2.40 GHz computer with 2. GB of RAM.

The described fuzzy knowledge base was used in the evaluation of our approach. Implicit knowledge was extracted using the greatest lower bound service of FiRE, asking for the degree of participation of all individuals, in all the defined domain concepts. The average number of assertions per individual was 13 while the defined concepts were 33, that together with the 2140 individuals (i.e entries of the database) resulted to 29460 explicit assertions and the extraction of 2430 implicit. These results, together with concept and role axioms, were stored to a Sesame repository using the proposed fuzzy OWL triples syntax to form a repository of 529.926 triples.

The average time for the GLB reasoning process and the conversion of explicit and implicit knowledge to fuzzy OWL syntax in triples was 1112 milliseconds. The time required for uploading the knowledge to a Sesame repository depends on the type of repository (Memory or Native) and also on repository's size. Based on our experiments, we have observed that the upload time is polynomial to the size of the repository but without significant differences. Therefore, the average minimum upload time to an almost empty repository (0-10.000 triples) is 213 milliseconds while the average maximum upload time to a full repository (over 500.000 triples) is 700 milliseconds.

FiRE and Sesame were also examined in the use of expressive fuzzy queries. The performance in this case mainly depended on the complexity of the query but also on the type and size of the repository. Queries using role names in combination with large repositories can dramatically slow down the response. Table 2 illustrates the response times in milliseconds using both types of repositories and different repository sizes. Repository sizes was set by adjusting the number of assertions. As it can be observed, very expressive queries seeking for young female models with beautiful legs and eyes as well as long hair, a popular demand in commercial spots, can be easily performed. It is worth mentioning that these queries consist of higher domain concepts defined in our fuzzy knowledge base.

Since our system is not a sound and complete query answering system for f-$\mathcal{SHIN}$, the GLB service performed before uploading the triples is employed in order to use as much of the expressivity of the language as possible producing new implied assertions.

Furthermore, the results regarding query answering time are also very encouraging, at least for the specific application. Although, compared to crisp querying, over crisp knowledge bases, our method might require several more seconds to be answered (mainly due to post processing steps for GFCQs or due to very lengthy SPARQL queries for CTQs) this time is significantly less, compared to

$$\boxed{\begin{aligned}
\mathcal{T} = \{\mathsf{MiddleAged} &\equiv \mathsf{40s} \sqcup \mathsf{50s},\\
\mathsf{TallChild} &\equiv \mathsf{Child} \sqcap (\mathsf{Short} \sqcup \mathsf{Normal\_Height}),\\
\mathsf{Father} &\equiv \mathsf{Male} \sqcap (\mathsf{30s} \sqcup \mathsf{MiddleAged}),\\
\mathsf{Legs} &\equiv \mathsf{Female} \sqcap (\mathsf{Normal\_Height} \sqcup \mathsf{Tall})\\
&\qquad \sqcap(\mathsf{Normal} \sqcup \mathsf{Perfect}) \sqcap (\mathsf{Fit} \sqcup \mathsf{PerfectFitness}),\\
\mathsf{Teacher} &\equiv (\mathsf{30s} \sqcup \mathsf{MiddleAged}) \sqcap \mathsf{Elegant} \sqcap \mathsf{Classic},\\
\mathsf{Scientist} &\equiv \mathsf{Male} \sqcap \mathsf{Classic} \sqcap (\mathsf{50s} \sqcup \mathsf{60s})\\
&\qquad \sqcap\mathsf{Serious} \sqcap \exists \mathsf{has-eyeCondition.Glasses}\ \}
\end{aligned}}$$

Table 1. An excerpt of the Knowledge Base ($TBox$).

the time spent by producers on casting (usually counted in days), since they usually have to browse through a very large number of videos and images before they decide.

6 Conclusions

Due to the fact that imperfect information is spread along the web, the effective management of imperfect knowledge is very important for the substantial evolution of the Semantic Web. In this paper, we have proposed an architecture that can be used for storing and querying fuzzy knowledge bases for the semantic web. Our proposal which is based on DL f-$\mathcal{SHIN}$, consists of the RDF triples syntax accommodating the fuzzy element, the fuzzy reasoning engine FiRE and its integration with RDF Store Sesame which permits very expressive fuzzy queries.

The proposed architecture was evaluated using an industrial application scenario about casting for TV commercials and spots. The obtained results are very promising from the querying perspective. From the initial 29460 explicit assertions made by database instances for models, 2430 new implicit assertions where extracted and both uploaded in the Sesame repository. In this way expressive semantic queries like "Find me young female models with beautiful legs and eyes as well as long hair", that might have proved very difficult or even impossible using the producing company's database, are applicable through FiRE. This reveals both the strength of knowledge-based applications, and technologies for managing fuzzy knowledge, since a wealth of the information of the databases, like age, height, as well as many high level concepts of the specific application, like "beautiful eyes", "perfect fitness" and "scientist look" are inherently fuzzy.

As far as future directions are concerned, we intend to further investigate on different ways of performing queries using expressive fuzzy description logics. Finally, it would be of great interest to attempt a comparison between the proposed architecture and approaches using fuzzy DL-lite ontologies and approximation.

Query	Native			Memory		
	100.000	250.000	500.000	100.000	250.000	500.000
$x \leftarrow \mathsf{Scientist}(x)$	1042	2461	3335	894	2364	3332
$x \leftarrow \mathsf{Father}(x) \geq 1 \wedge \mathsf{Teacher}(x) \geq 0.8$ $\wedge \mathsf{Normal\_Height}(x) \geq 0.5.$	1068	2694	3935	994	2524	3732
$x \leftarrow \mathsf{Scientist}(x) : 0.8$	2562	4173	5235	3042	4543	6027
$x \leftarrow \mathsf{Father}(x) : 0.6 \wedge \mathsf{Teacher}(x) : 0.7$ $\wedge \mathsf{Normal\_Height}(x) : 0.8.$	4318	6694	8935	4341	7896	9306

Table 2. Fuzzy queries evaluation. Queries performed on repositories of size 100.000 250.000 and 500.000. The response time is in milliseconds

Acknowledgements.

This work is supported by the FP6 Network of Excellence EU project X-Media (FP6-026978) and K-space (IST-2005-027026).

References

1. Description-logic knowledge representation system specification from the KRSS group of the ARPA knowledge sharing effort. `http://dl.kr.org/krss-spec.ps`.
2. F. Baader, D. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications*. Cambridge University Press, 2002.
3. B.Glimm, I.Horrocks, C.Lutz, and U.Sattler. Conjunctive query answering for $\mathcal{SHIQ}$. Technical report, University of Manchester, 2006.
4. A. Chortaras, Giorgos Stamou, and Andreas Stafylopatis. Adaptation of weighted fuzzy programs. In *Proc. of the International Conference on Artificial Neural Networks (ICANN 2006)*, pages 45–54. Springer, 2006.
5. V. Cross. Fuzzy information retrieval. *Journal of Intelligent Information Systems*, 3:29–56, 1994.
6. G.Stoilos, G.Stamou, V.Tzouvaras, J.Z.Pan, and I.Horrocks. Reasoning with very expressive fuzzy description logics. *Journal of Artificial Intelligence Research*, 30(5):273–320, 2007.
7. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$. In David MacAllester, editor, *CADE-2000*, number 1831 in LNAI, pages 482–496. Springer-Verlag, 2000.
8. G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice-Hall, 1995.
9. M.Mazzieri and A.F.Dragoni. A fuzzy semantics for semantic web languages. In *ISWC-URSW*, pages 12–22, 2005.
10. J.Z. Pan, G. Stamou, G. Stoilos, and E. Thomas. Expressive querying over fuzzy DL-Lite ontologies. In *Proceedings of the International Workshop on Description Logics (DL 2007)*, 2007.
11. P.F.Patel-Schneider, P.Hayes, and I.Horrocks. Owl web ontology language semantics and abstract syntax. Technical report, World Wide Web Consortium, 2004.
12. E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF, 2006. W3C Working Draft, http://www.w3.org/TR/rdf-sparql-query/.
13. R.Giugno and T.Lukasiewicz. P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In *JELIA '02: Proceedings of the European Conference on Logics in Artificial Intelligence*, pages 86–97, London, UK, 2002. Springer-Verlag.
14. G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan, and I. Horrocks. Fuzzy OWL: Uncertainty and the Semantic Web. In *Proc. of the OWL-ED 2005.*
15. U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
16. U.Straccia and G.Visco. DLMedia: an ontology mediated multimedia information retrieval system. In *Proceeedings of the International Workshop on Description Logics (DL 07)*, volume 250, Insbruck, Austria, 2007. CEUR.
17. V. Vaneková, J. Bella, P. Gurský, and T. Horváth. Fuzzy RDF in the semantic web: Deduction and induction. In *Proceedings of Workshop on Data Analysis (WDA 2005)*, pages 16–29, 2005.