

# An Ontology Mediated Multimedia Information Retrieval System

Umberto Straccia  
ISTI - CNR, Pisa, ITALY

**Abstract**—We outline DL-Media, an ontology mediated multimedia information retrieval system, which combines logic-based retrieval with multimedia feature-based similarity retrieval. An ontology layer is used to define (in terms of a fuzzy description logic) the relevant abstract concepts and relations of the application domain, while a content-based multimedia retrieval system is used for feature-based retrieval.

## I. INTRODUCTION

*Multimedia Information Retrieval (MIR)* concerns the retrieval of those multimedia objects of a collection that are relevant to a user information need.

In this paper we outline DL-MEDIA, an ontology mediated MIR system, which combines logic-based retrieval with multimedia feature-based similarity retrieval. An ontology layer is used to define (in terms of a DLR-Lite like description logic) the relevant abstract concepts and relations of the application domain, while a content-based multimedia retrieval system is used for feature-based retrieval. We will illustrate its logical model, its architecture, its representation and query language and the experiments we conducted.

## II. THE LOGIC-BASED MIR MODEL

Overall, DL-MEDIA lies in the context of *Logic-based Multimedia Information Retrieval (LMIR)* (see [8] for an extensive overview on LMIR literature. A recent work is also *e.g.* [6], see also [7] and [2] for a more complex multimedia ontology model). In DL-MEDIA, from each multimedia object  $o \in \mathbb{O}$  (such as a piece of text, image region, etc.) we automatically extract low-level features such as text index term weights (object of type text), colour distribution, shape, texture, spatial relationships (object of type image), mosaiced video-frame sequences and time relationships (object of type video). Furthermore, each multimedia object  $o \in \mathbb{O}$  may also have associated a metadata record in some format. Specifically, in DL-MEDIA the data is stored in MPEG-7 format [9]. All this data belongs to the *multimedia data layer*. On top of it we have the so-called *ontology layer* in which we define the relevant concepts of our application domain through which we may retrieve the multimedia objects  $o \in \mathbb{O}$ . In DL-MEDIA this layer consists of an ontology of concepts defined in a description logic [3], which is a variant of DLR-Lite [4] with concrete domains.

## III. ARCHITECTURE

The DL-MEDIA architecture has two basic components: the DL-based ontology component and the (feature-based) multimedia retrieval component (see Figure 1).

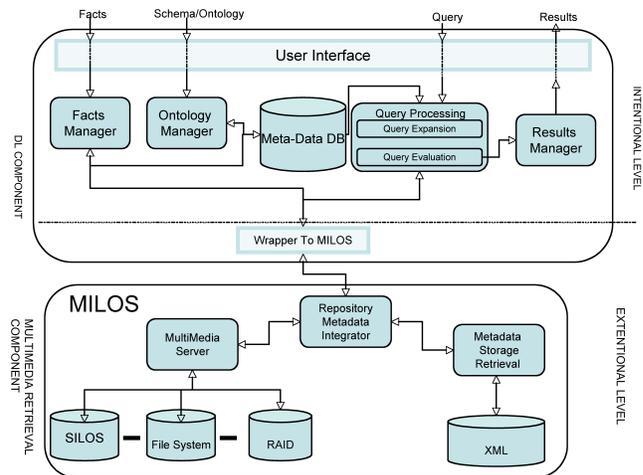


Fig. 1. DL-MEDIA architecture.

The DL-component supports both the definition of the ontology and query answering. In particular, it provides a logical query and representation language, which is an extension of the DL language DLR-Lite [4], [11], [12] without negation.

The (feature-based) multimedia retrieval component, supports the retrieval of text and images based on low-level feature indexing. Currently, we rely on our MIR system MILOS<sup>1</sup>, but other MIR systems can be used by means of appropriate wrappers. MILOS (Multimedia Content Management System) is a general purpose software component that supports the storage and content based retrieval of any multimedia documents whose descriptions are provided by using arbitrary metadata models represented in XML. MILOS is flexible in the management of documents containing different types of data and content descriptions; it is efficient and scalable in the storage and content based retrieval of these documents [1]. In addition to support XML query language standards such as XPath and XQuery, MILOS offers advanced multimedia search and indexing functionality with new operators that deal with approximate match and ranking of XML and multimedia data (see the MILOS web page for more about it). Approximate match of multimedia data is based on metric spaces theory, where the similarity among multimedia objects is measured in terms of a distance function.

Operationally, a user submits a conceptual query (a logic-

<sup>1</sup><http://milos.isti.cnr.it/>

based conjunctive query) by means of the DL-component. The DL-component will then use the ontology to *reformulate* the initially query into one or several queries (query expansion). These queries are then translated and submitted to MILOS (using the wrapper). MILOS then provides back the top-k answers for each of the issued queries. The ranked lists will then be merged into one final result list and displayed to the user (using the the query evaluation module).

#### IV. ONTOLOGY AND QUERY LANGUAGE

For computational reasons the particular logic DL-MEDIA adopts is based on the DLR-Lite [4] Description Logic (DL) [3] without negation, which has LogSpace data complexity, *i.e.* the same complexity of SQL over relational databases. The DL will be used in order define the relevant abstract concepts and relations of the application domain. On the other hand, conjunctive queries will be used to describe the information need of a user.

The DL-MEDIA logic extends DLR-Lite by enriching it with built-in predicates allowing to address three categories of retrieval: feature-based, semantic-based and their combination.

**Ontology language syntax.** DL-MEDIA supports concrete domains with specific predicates on it. The *concrete predicates* that DL-MEDIA allows are not only relational predicates such as  $([i] \leq 1500)$  (*e.g.* the value of the  $i$ -th column of an  $n$ -ary relation is less or equal than 1500), but also similarity predicates such as  $([i] \text{ simText } 'logic, image, retrieval')$ , which given a piece of text  $x$  appearing in the  $i$ -th column of a tuple returns the system's degree (in  $[0, 1]$ ) of being  $x$  about the keywords 'logic, image, retrieval'.

Formally, a *concrete domain* in DL-MEDIA is a pair  $\langle \Delta_D, \Phi_D \rangle$ , where  $\Delta_D$  is an interpretation domain and  $\Phi_D$  is the set of *domain predicates*  $d$  with a predefined arity  $n$  and an interpretation  $d^D: \Delta_D^n \rightarrow [0, 1]$  (see also [10]). The list of the specific domain predicates is presented below.

DL-MEDIA allows to specify the ontology by relying on axioms. Consider an alphabet of  $n$ -ary relation symbols (denoted  $R$ ), *e.g.* MyMetadata with signature MyMetadata(docID, title, image, tag), and an alphabet of unary relations, called *atomic concepts* (and denoted  $A$ ), *e.g.* ItalianCity with signature ItalianCity(id). Then an *axiom* is of the form

$$Rl_1 \sqcap \dots \sqcap Rl_m \sqsubseteq Rr ,$$

where  $m \geq 1$ , all  $Rl_i$  and  $Rr$  have the same arity and where each  $Rl_i$  is a so-called *left-hand relation* and  $Rr$  is a *right-hand relation*. Informally, such an axiom may be read as “if  $Rl_1$  and  $Rl_2 \dots$  and  $Rl_m$  then  $Rr$ ”. As illustrative purpose, a simple ontology axiom may be of the form ItalianCity  $\sqsubseteq$  EuropeanCity with informal reading “any italian city is a european city”. Here ItalianCity and EuropeanCity are unary relations with signature ItalianCity(id) and EuropeanCity(id), respectively. Similarly, the ontology axiom ItalianCity  $\sqcap$  BigCity  $\sqsubseteq$  BigEuropeanCity has informal reading “any italian city, which is also big is a big european city” (BigCity

and BigEuropeanCity are again unary relations with signature BigCity(id) and BigEuropeanCity(id), respectively). But we may also involve  $n$ -ary realtions in ontology axioms. For instance, assume that we have metadata records (*e.g.* in MPEG-7 [9]) about multimedia objects, with signature MyMetadata(docID, title, image, tag), where the image attribute contains the image raw data, and the title and tag attributes contain the textual image title and image content description. Suppose now that from the metadata records, we would like to extract just the document ID and the image, and call this new relation ImageDescr(docID, image). In database terminology this amounts in a projection of MyMetadata relation on the first and third column. In our language, the projection of an  $n$ -ary relation  $R$  on the columns  $i_1, \dots, i_k$  ( $1 \leq i_1, i_2, \dots, i_k \leq n, 1 \leq i \leq n$ ), will be indicated with  $\exists[i_1, \dots, i_k]R$ . Hence, *e.g.*  $\exists[1, 3]$ MyMetadata is the binary relation that is the projection on the first and third column of the MyMetadata relation. As a consequence, the axioms

$$\begin{aligned} \exists[1, 3]MyMetadata &\sqsubseteq \exists[1, 2]ImageDescr \\ \exists[1, 4]MyMetadata &\sqsubseteq \exists[1, 2]Tag \\ \exists[1, 2]MyMetadata &\sqsubseteq \exists[1, 2]Title \end{aligned}$$

state that the relation ImageDescr contains the projection of the MyMetadata relation on the first and third column (the other axioms are interpreted similarly).

In case of a projection, we may further restrict it according to some conditions. For instance, suppose we have a relation with signature Person(firstname, lastname, age, email, sex) then  $\exists[2, 4]Person.((\exists[3] \geq 25))$  corresponds to the set of tuples  $\langle lastname, email \rangle$  such that the third column of the relation Person, *i.e.* the person's age, is equal or greater than 25. Examples of axioms are

$$\begin{aligned} \exists[2, 3]Person &\sqsubseteq \exists[1, 2]hasAge \\ \exists[2, 4]Person &\sqsubseteq \exists[1, 2]hasEmail \\ \exists[2, 1, 4]Person.((\exists[3] \geq 18) \sqcap ((\exists[5] = 'male'))) &\sqsubseteq \exists[1, 2, 3]AdultMalePerson \end{aligned}$$

Note that in the last axiom, we have multiple conditions: we require that the age is greater or equal than 18 and the gender is male. This axiom defines the relation AdultMalePerson(lastname, firstname, email).

Finally, we also allow to specify similarity conditions. So, the condition  $([i] \text{ simText } 'k_1 \dots k'_n)$  evaluates the degree of being the text of the  $i$ -th column similar to the list of keywords  $k_1 \dots k_n$ , while  $([i] \text{ simImg } URN)$  returns the system's degree of being the image identified by the  $i$ -th column similar to the object  $o$  identified by the URN (*Uniform Resource Name*<sup>2</sup>). Examples axioms involving similarity conditions are,

$$\begin{aligned} (\exists[1]ImageDescr.((\exists[3] \text{ simImg } urn1))) \sqcap (\exists[1]Tag.((\exists[2] = 'sunrise'))) & \quad (1) \\ &\sqsubseteq Sunrise\_On\_Sea \\ \exists[1]Title.(\exists[2] \text{ simTxt } 'lion') &\sqsubseteq Lion \end{aligned}$$

where  $urn1$  identifies the image in Fig. 2. The former axiom (axiom 2) informally states that an image similar to the image depicted in Fig. 2 with a tag labelled 'sunrise' is about a Sunrise\_On\_Sea (to a system computed degree in  $[0, 1]$ ). Similarly, in axiom (2) informally states that an image whose

<sup>2</sup>[http://en.wikipedia.org/wiki/Uniform\\_Resource\\_Name](http://en.wikipedia.org/wiki/Uniform_Resource_Name)



Fig. 2. Sun rise

metadata record contains an attribute Title which is about 'lion' is about a Lion. Note that Sunrise\_On\_Sea and Lion have signature Sunrise\_On\_Sea(*docID*) and Lion(*docID*), respectively.

The exact syntax of the relations appearing on the left-hand and right-hand side of ontology axioms is specified below (where  $h \geq 1$ ):

$$\begin{aligned}
Rr &\longrightarrow A \mid \exists[i_1, \dots, i_k]R \\
Rl &\longrightarrow A \mid \exists[i_1, \dots, i_k]R \mid \\
&\quad \exists[i_1, \dots, i_k]R.(Cond_1 \sqcap \dots \sqcap Cond_h) \\
Cond &\longrightarrow ([i] \leq v) \mid ([i] < v) \mid ([i] \geq v) \mid ([i] > v) \mid \\
&\quad ([i] = v) \mid ([i] \neq v) \mid \\
&\quad ([i] \text{ simText } k_1, \dots, k'_n) \mid ([i] \text{ simImg URN})
\end{aligned}$$

where  $A$  is an atomic concept,  $R$  is an  $n$ -ary relation with  $1 \leq i_1, i_2, \dots, i_k \leq n$ ,  $1 \leq i \leq n$  and  $v$  is a value of the concrete interpretation domain of the appropriate type. Here  $\exists[i_1, \dots, i_k]R$  is the projection of the relation  $R$  on the columns  $i_1, \dots, i_k$  (the order of the indexes matters). Hence,  $\exists[i_1, \dots, i_k]R$  has arity  $k$ . On the other hand,  $\exists[i_1, \dots, i_k]R.(Cond_1 \sqcap \dots \sqcap Cond_h)$  further restricts the projection  $\exists[i_1, \dots, i_k]R$  according to the conditions specified in  $Cond_i$ . For instance,  $([i] \leq v)$  specifies that the values of the  $i$ -th column have to be less or equal than the value  $v$ ,  $([i] \text{ simText } k_1 \dots k'_n)$  evaluates the degree of being the text of the  $i$ -th column similar to the list of keywords  $k_1 \dots k_n$ , while  $([i] \text{ simImg URN})$  returns the system's degree of being the image identified by the  $i$ -th column similar to the object  $o$  identified by the URN. Finally, a DL-MEDIA ontology  $\mathcal{O}$  consists of a set of axioms.

**Query language syntax.** Concerning queries, a DL-MEDIA query consists of a conjunctive query of the form

$$q(\mathbf{x}) \leftarrow R_1(\mathbf{z}_1) \wedge \dots \wedge R_l(\mathbf{z}_l),$$

where  $q$  is an  $n$ -ary predicate, every  $R_i$  is an  $n_i$ -ary predicate,  $\mathbf{x}$  is a vector of variables, and every  $\mathbf{z}_i$  is a vector of constants, or variables. We call  $q(\mathbf{x})$  its *head* and  $R_1(\mathbf{z}_1) \wedge \dots \wedge R_l(\mathbf{z}_l)$  its *body*.  $R_i(\mathbf{z}_i)$  may also be a concrete unary predicate of the form  $(z \leq v), (z < v), (z \geq v), (z > v), (z = v), (z \neq v), (z \text{ simText } k_1, \dots, k'_n), (z \text{ simImg URN})$ , where  $z$  is a variable,  $v$  is a value of the appropriate concrete domain,  $k_i$

is a keyword and  $URN$  is an URN. Example queries are:

```

q(x) ← Sunrise_On_Sea(x)
      // find objects about a sunrise on the sea
q(x) ← CreatorName(x, y) ∧ (y = 'paolo') ∧ Title(x, z), (z simText 'tour')
      // find images made by Paolo whose title is about 'tour'
q(x) ← ImageDescr(x, y) ∧ (y simImg urn2)
      // find images similar to a given image identified by urn2
q(x) ← ImageObject(x) ∧ isAbout(x, y1) ∧ Car(y1) ∧ isAbout(x, y2) ∧ Racing(y2)
      // find image objects about cars racing

```

We note that a query may also be written as

$$q(\mathbf{x}) \leftarrow \exists \mathbf{y} \phi(\mathbf{x}, \mathbf{y}),$$

where  $\phi(\mathbf{x}, \mathbf{y})$  is  $R_1(\mathbf{z}_1) \wedge \dots \wedge R_l(\mathbf{z}_l)$  and no variable in  $\mathbf{y}$  occurs in  $\mathbf{x}$  and vice-versa. Here,  $\mathbf{x}$  are the so-called *distinguished variables*, while  $\mathbf{y}$  are the so-called *non distinguished variables*, which are existentially quantified. For a query atom  $q$ , we will write  $\langle q(\mathbf{c}), s \rangle$  to denote that the tuple  $\mathbf{c}$  is instance of the query atom  $q$  to degree at least  $s \in [0, 1]$ .

**Semantics.** From a semantics point of view, DL-MEDIA is based on mathematical fuzzy logic [5] because

- the underlying MIR system MILOS is based on fuzzy aggregation operators to combine the similarity degrees among low-level image and textual features; and
- then the DL-component allows for low data-complexity reasoning (LogSpace).

Given a concrete domain  $\langle \Delta_D, \Phi_D \rangle$ , an *interpretation*  $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}} \rangle$  consists of a *fixed infinite domain*  $\Delta$ , containing  $\Delta_D$ , and an *interpretation function*  $\cdot^{\mathcal{I}}$  that maps

- every atom  $A$  to a function  $A^{\mathcal{I}}: \Delta \rightarrow [0, 1]$
- maps an  $n$ -ary predicate  $R$  to a function  $R^{\mathcal{I}}: \Delta^n \rightarrow [0, 1]$
- constants to elements of  $\Delta$  such that  $a^{\mathcal{I}} \neq b^{\mathcal{I}}$  if  $a \neq b$  (unique name assumption).

Intuitively, rather than being an expression (e.g.  $R(\mathbf{c})$ ) either true or false in an interpretation, it has a degree of truth in  $[0, 1]$ . So, given a constant  $c$ ,  $A^{\mathcal{I}}(c)$  determines to which degree the individual  $c$  is an instance of atom  $A$ . Similarly, given an  $n$ -tuple of constants  $\mathbf{c}$ ,  $R^{\mathcal{I}}(\mathbf{c})$  determines to which degree the tuple  $\mathbf{c}$  is an instance of the relation  $R$ . We also assume to have one object for each constant, denoting exactly that object. In other words, we have standard names, and we do not distinguish between the alphabet of constants and the objects in  $\Delta$ . Furthermore, we assume that the relations have a typed signature and the interpretations have to agree on the relation's type. For instance, the second argument of the Title relation (see axiom 2) is of type String and any interpretation function requires that the second argument of  $\text{Title}^{\mathcal{I}}$  is of type String. To the easy of presentation, we omit the formalization of this aspect and leave it at the intuitive level.

In the following, we use  $\mathbf{c}$  to denote an  $n$ -tuple of constants, and  $\mathbf{c}[i_1, \dots, i_k]$  to denote the  $i_1, \dots, i_k$ -th components of  $\mathbf{c}$ . For instance,  $(a, b, c, d)[3, 1, 4]$  is  $(c, a, d)$ .

Concerning concrete comparison predicates, the interpretation function  $\cdot^{\mathcal{I}}$  has to satisfy

$$([i] \leq v)^{\mathcal{I}}(\mathbf{c}) = \begin{cases} 1 & \text{if } \mathbf{c}[i] \leq v \\ 0 & \text{otherwise} \end{cases}$$

and similarly for the other comparison constructs,  $([i] < v)$ ,  $([i] \geq v)$ ,  $([i] > v)$  and  $([i] = v) \mid ([i] \neq v)$ .

Concerning the concrete similarity predicates, the interpretation function  $\cdot^{\mathcal{I}}$  has to satisfy

$$\begin{aligned} ([i] \text{ simTtxt}' k_1, \dots, k_n)^{\mathcal{I}}(\mathbf{c}') &= \text{simTtxt}^{\mathcal{D}}(\mathbf{c}[i], k_1, \dots, k_n) \in [0, 1] \\ ([i] \text{ simImgURN})^{\mathcal{I}}(\mathbf{c}) &= \text{simImg}^{\mathcal{D}}(\mathbf{c}[i], \text{URN}) \in [0, 1] . \end{aligned}$$

where  $\text{simTtxt}^{\mathcal{D}}$  and  $\text{simImg}^{\mathcal{D}}$  are the textual and image similarity predicates supported by the underlying MIR system MILOS.

Concerning axioms, as in an interpretation each  $Rl_i(\mathbf{c})$  has a degree of truth, we have to specify how to combine them to determine the degree of truth of the conjunction  $Rl_1 \sqcap \dots \sqcap Rl_m$ . Usually, in mathematical fuzzy logic one uses a so-called T-norm  $\otimes$  to combine the truth of “conjunctive” expressions<sup>3</sup> (see [5]). Some typical T-norms are

$$\begin{aligned} x \otimes y &= \min(x, y) && \text{Gödel conjunction} \\ x \otimes y &= \max(x + y - 1, 0) && \text{Łukasiewicz conjunction} \\ x \otimes y &= x \cdot y && \text{Product conjunction} . \end{aligned}$$

In DL-MEDIA, to be compliant with the underlying MILOS system, the T-norm is fixed to be Gödel conjunction.

The interpretation function  $\cdot^{\mathcal{I}}$  has to satisfy: for all  $\mathbf{c} \in \Delta^k$  and  $n$ -ary relation  $R$ :

$$\begin{aligned} (\exists [i_1, \dots, i_k] R)^{\mathcal{I}}(\mathbf{c}) &= \sup_{\mathbf{c}' \in \Delta^n, \mathbf{c}'[i_1, \dots, i_k] = \mathbf{c}} R^{\mathcal{I}}(\mathbf{c}') \\ (\exists [i_1, \dots, i_k] R. (\text{Cond}_1 \sqcap \dots \sqcap \text{Cond}_l))^{\mathcal{I}}(\mathbf{c}) &= \\ \sup_{\mathbf{c}' \in \Delta^n, \mathbf{c}'[i_1, \dots, i_k] = \mathbf{c}} \min(R^{\mathcal{I}}(\mathbf{c}'), \text{Cond}_1^{\mathcal{I}}(\mathbf{c}'), \dots, \text{Cond}_l^{\mathcal{I}}(\mathbf{c}')) & \end{aligned}$$

Some explanation is in place. Consider  $(\exists [i_1, \dots, i_k] R)$ . Informally, from a classical semantics point of view,  $(\exists [i_1, \dots, i_k] R)$  is the projection of the relation  $R$  over the columns  $i_1, \dots, i_k$  and, thus, corresponds to the set of tuples

$$\{\mathbf{c} \mid \exists \mathbf{c}' \in R \text{ s.t. } \mathbf{c}'[i_1, \dots, i_k] = \mathbf{c}\} .$$

Note that for a fixed tuple  $\mathbf{c}$  there may be several tuples  $\mathbf{c}' \in R$  such that  $\mathbf{c}'[i_1, \dots, i_k] = \mathbf{c}$ . Now, if we switch to fuzzy logic, for a fixed tuple  $\mathbf{c}$  and interpretation  $\mathcal{I}$ , each of the previous mentioned  $\mathbf{c}'$  is instance of  $R$  to a degree  $R^{\mathcal{I}}(\mathbf{c}')$ . It is usual practice in mathematical fuzzy logic to consider the supremum among these degrees (the existential is interpreted as supremum), which motivates the expression  $\sup_{\mathbf{c}' \in \Delta^n, \mathbf{c}'[i_1, \dots, i_k] = \mathbf{c}} R^{\mathcal{I}}(\mathbf{c}')$ . The argument is similar for the  $\exists [i_1, \dots, i_k] R. (\text{Cond}_1 \sqcap \dots \sqcap \text{Cond}_l)$  construct except that we consider also the additional conditions as conjuncts.

Now given an interpretation  $\mathcal{I}$ , the notion of  $\mathcal{I}$  is a model of (satisfies) an axiom  $\tau$ , denoted  $\mathcal{I} \models \tau$ , is defined as follows:  $\mathcal{I} \models Rl_1 \sqcap \dots \sqcap Rl_m \sqsubseteq Rr$  iff for all  $\mathbf{c} \in \Delta^n$ ,  $\min(Rl_1^{\mathcal{I}}(\mathbf{c}), \dots, Rl_m^{\mathcal{I}}(\mathbf{c})) \leq Rr^{\mathcal{I}}(\mathbf{c})$ , where we assume that the arity of  $Rr$  and all  $Rl_i$  is  $n$ . An interpretation  $\mathcal{I}$  is a model of (satisfies) an ontology  $\mathcal{O}$  iff it satisfies each element in it.

<sup>3</sup>Given truth degrees  $x$  and  $y$ , the conjunction of  $x$  and  $y$  is  $x \otimes y$ .  $\otimes$  has to be symmetric, associative, monotone in its arguments and  $x \otimes 1 = x$ .

Concerning queries, an interpretation  $\mathcal{I}$  is a model of (satisfies) a query  $q$  the form  $q(\mathbf{x}) \leftarrow \exists \mathbf{y} \phi(\mathbf{x}, \mathbf{y})$ , denoted  $\mathcal{I} \models q$ , iff for all  $\mathbf{c} \in \Delta^n$ :

$$q^{\mathcal{I}}(\mathbf{c}) \geq \sup_{\mathbf{c}' \in \Delta^x \times \dots \times \Delta^y} \phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}') ,$$

where  $\phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')$  is obtained from  $\phi(\mathbf{c}, \mathbf{c}')$  by replacing every  $R_i$  by  $R_i^{\mathcal{I}}$ , and Gödel conjunction is used to combine all the truth degrees  $R_i^{\mathcal{I}}(\mathbf{c}'')$  in  $\phi^{\mathcal{I}}(\mathbf{c}, \mathbf{c}')$ . Furthermore, we say that an interpretation  $\mathcal{I}$  is a model of (satisfies)  $\langle q(\mathbf{c}), s \rangle$ , denoted  $\mathcal{I} \models \langle q(\mathbf{c}), s \rangle$ , iff  $q^{\mathcal{I}}(\mathbf{c}) \geq s$ . We say  $\mathcal{O}$  entails  $q(\mathbf{c})$  to degree  $s$ , denoted  $\mathcal{O} \models \langle q(\mathbf{c}), s \rangle$ , iff each model  $\mathcal{I}$  of  $\mathcal{O}$  is a model of  $\langle q(\mathbf{c}), s \rangle$ . The greatest lower bound of  $q(\mathbf{c})$  relative to  $\mathcal{O}$  is

$$\text{glb}(\mathcal{O}, q(\mathbf{c})) = \sup\{s \mid \mathcal{O} \models \langle q(\mathbf{c}), s \rangle\} .$$

As now each answer to a query has a degree of truth, the basic inference problem that is of interest in DL-MEDIA is the top- $k$  retrieval problem, formulated as follows. Given  $\mathcal{O}$  and a query with head  $q(\mathbf{x})$ , retrieve  $k$  tuples  $\langle \mathbf{c}, s \rangle$  that instantiate the query predicate  $q$  with maximal degree, and rank them in decreasing order relative to the degree  $s$ , denoted

$$\text{ans}_k(\mathcal{O}, q) = \text{Top}_k\{\langle \mathbf{c}, s \rangle \mid s = \text{glb}(\mathcal{O}, q(\mathbf{c}))\} .$$

**Query answering.** From a query answering point of view, the DL-MEDIA system extends the DL-Lite/DLR-Lite reasoning method [4] to the fuzzy case. The algorithm is an extension of the one described in [4], [11], [12]). Roughly, given a query  $q(\mathbf{x}) \leftarrow R_1(\mathbf{z}_1) \wedge \dots \wedge R_l(\mathbf{z}_l)$ , (i) by considering  $\mathcal{O}$ , the user query  $q$  is reformulated into a set of conjunctive queries  $r(q, \mathcal{O})$ . Informally, the basic idea is that the reformulation procedure closely resembles a top-down resolution procedure for logic programming, where each axiom is seen as a logic programming rule. For instance, given the query  $q(x) \leftarrow A(x)$  and suppose that  $\mathcal{O}$  contains the axioms  $B_1 \sqsubseteq A$  and  $B_2 \sqsubseteq A$ , then we can reformulate the query into two queries  $q(x) \leftarrow B_1(x)$  and  $q(x) \leftarrow B_2(x)$ , exactly as it happens for top-down resolution methods in logic programming; (ii) from the set of reformulated queries  $r(q, \mathcal{O})$  we remove redundant queries; (iii) the reformulated queries  $q' \in r(q, \mathcal{O})$  are translated to MILOS queries and evaluated. The query evaluation of each MILOS query returns the top- $k$  answer set for that query; (iv) all the  $n = |r(q, \mathcal{O})|$  top- $k$  answer sets have to be merged into the unique top- $k$  answer set  $\text{ans}_k(\mathcal{O}, q)$ . As  $k \cdot n$  may be large, we apply the *Disjunctive Threshold Algorithm* (DTA, see [12] for the details) to merge all the answer sets.

## V. DL-MEDIA AT WORK

A prototype of the DL-MEDIA system has been implemented. The main interface is shown in Fig. 3. In the upper panel, the currently loaded ontology component  $\mathcal{O}$  is shown. Below it and to the right, the current query is shown (“find an image about sunrises on the sea”, we also do not report here the concrete syntax of the DL-MEDIA DL). So far, in DL-MEDIA, given a query, it will be transformed, using the ontology, into several queries (according to the query reformulation

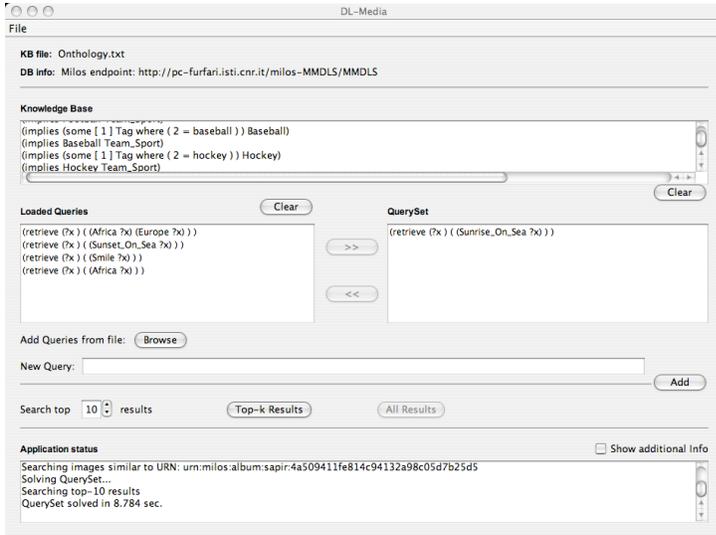


Fig. 3. DL-MEDIA main interface.

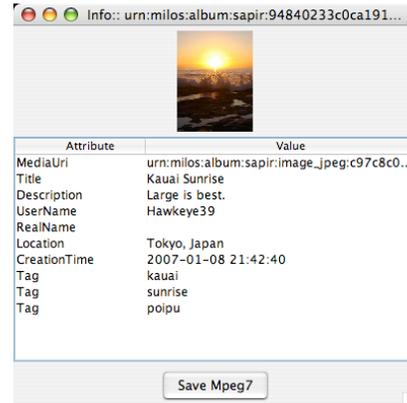


Fig. 4. Image metadata.

```

(map-role ImageDescr /Mpeg7/Description/MultimediaContent/Image (docID[string] image[string]))
(map-role MediaUri /MediaLocator/MediaUri (docID[string] mediauri[string]))
(map-role Title /photo/title (docID[string] title[string]))
(map-role OwnerUserName /photo/owner/@username (docID[string] creator-username[string]))
(map-role OwnerRealName /photo/owner/@realname (docID[string] creator-realname[string]))
(map-role OwnerLocation /photo/owner/@location (docID[string] creator-location[string]))
(map-role Mpeg7Content /Mpeg7/Description (docID[string] data[string]))
(map-role Location /photo/owner/@location (docID[string] location[string]))
(map-role CreationTime /photo/dates/@taken (docID[string] date[string]))
(map-role Description /photo/description (docID[string] descr[string]))
(map-role Tag /photo/tags/tag (docID[string] tag[string]))
(map-role Comment /photo/comments/comment (docID[string] comment[string]))

```

Fig. 5. DL-MEDIA mapping rules.

step described above) and then the conjunctive queries are transformed into appropriate queries (this component is called wrapper) in order to be submitted to the underlying database and multimedia engine. To support the query rewriting phase, DL-MEDIA allows also to write *schema mapping* rules, which map e.g. a relation name *R* into the concrete name of a XML tag (see Fig.5) and excerpt of the metadata format is shown in Fig.4. For instance, the execution of the query shown in Fig. 3 produces the ranked list of images shown in Fig. 6.

Related to each image, we may also access to its metadata, which is in our case an excerpt of MPEG-7 (the data can be edited by the user as well). We may also select an image of the result pane and further refine the query to retrieve images similar to the selected one.

## VI. EXPERIMENTS

We conducted an experiment with the DL-MEDIA system. We considered an image set of around 560.000 images together with their MPEG-7 metadata. The data has been provided by Flickr<sup>4</sup> as a courtesy and for experimental purposes only. In MILOS we have indexed the images' low-level features as

<sup>4</sup><http://www.flickr.com/>.



Fig. 6. DL-MEDIA results pane.

well as their associated XML metadata. We build an ontology with 356 concept definitions, 12 relations. Totally, we have 746 DL-MEDIA axioms. We build 10 queries to be submitted to the system and measured for each of them

TABLE I  
EXPERIMENTAL EVALUATION.

Query	Prec.	$q'_r$	$q_r$	$t_r$	$q_M$	$t_M$	$t_{DTA}$	$t_{Img}$	$t_{tot}$
Q1	1.0	2	2	0.005	1	0.3	0	0.613	1.045
Q2	0.8	48	48	2.125	1	0.327	0	0.619	3.073
Q3	0.9	3	2	0.018	1	2.396	0	0.617	3.036
Q4	0.8	6	6	0.03	1	0.404	0	0.642	1.147
Q5	0.9	10	6	0.113	1	0.537	0	0.614	1.359
Q6	0.8	10	6	0.254	1	1.268	0	0.86	2.387
Q7	1.0	4	4	0.06	3	15.101	0.004	0.635	15.831
Q8	0.9	522	420	0.531	7	13.620	0.009	0.694	14.895
Q9	0.1	360	288	0.318	20	40.507	0.029	0.801	41.631
Q10	0.9	37	36	0.056	20	36.073	0.018	0.184	36.320

- 1) the precision at 10, *i.e.* the percentage of relevant images within the top-10 results.
- 2) the number of queries generated after the reformulation process ( $q'_r$ );
- 3) the number of reformulated queries after redundancy elimination ( $q_r$ );
- 4) the time of the reformulation process ( $t_r$ );
- 5) the number of queries effectively submitted to MILOS ( $q_M$ );
- 6) the query answering time of MILOS for each submitted query ( $t_M$ );
- 7) the time of merging process using the DTA ( $t_{DTA}$ );
- 8) the time needed to visualize the images in the user interface ( $t_{Img}$ );
- 9) the total time from the submission of the initial query to the visualization of the final result ( $t_{tot}$ ).

The results are shown in Table I below (time is measured in seconds, precision has been evaluated subjectively).

Let's comment some points. The number of queries generated after query reformulation varies significantly and depends both on the structure of the ontology and the concepts involved in the original query. For instance, a query about African animals formulated as  $q_8(x) \leftarrow Animal(x) \wedge Africa(x)$  will be reformulated into several queries involving the sub-concepts of both Animal and Africa, which in our case is quite large. Also interesting is that, e.g. for query 8, we may remove more than 100 queries from  $r(q_8, \mathcal{O})$  by a simple query subsumption test check (see appendix). Besides the possibility to have large query reformulation sets, the query reformulation time is quite low (less than 0.5 seconds). Also negligible is the time spent by the DTA merging algorithm. The MILOS response time is quite reasonable once we submit one query only (the answer is provided within some seconds). Clearly, as we submit the queries sequentially to the MILOS system, the total time sums up. Of course, an improvement may be expected once we submit the queries to MILOS in parallel. This part is under development as a joint activity with the MILOS development group. Also note that effective number of queries  $q_M$  may not coincide with  $q_r$ , as we do not submit queries to MILOS, which involve abstract concepts only as they do not have a translation into a MILOS query (for instance, the query  $q_8$ , which despite belonging to the set of reformulated queries  $r(q_8, \mathcal{O})$  is not submitted, while the reformulated query  $q_{8_1}(x) \leftarrow Tag(x, animal) \wedge Tag(x, africa)$  is). Also, if we have already retrieved 10 images with score 1.0, we stop the

MILOS query submission phase.

From a qualitative point of view of the retrieved images, the precision is satisfactory, though more extensive experiments are needed to assess the effectiveness of the DL-MEDIA system.

## VII. CONCLUSIONS

In this work, we have outlined the DL-MEDIA system, *i.e.* an ontology mediated multimedia retrieval system. Main features (so far) of DL-MEDIA are that: (i) it uses an extension of DLR-Lite(D) like language as query and ontology representation language; (ii) it supports feature-based queries, semantic-based queries and their combination; and (iii) is promisingly scalable.

There are several points, which we are further investigating: (i) so far, we consider all reformulated queries as equal relevant in response to information need. However, it seems reasonable to assume that the more specific the reformulated query becomes the less relevant may be its answers. (ii) multithreading of reformulated queries (iii) from a language point of view, we would like to extend it by using rules on top of axioms and adding more concrete predicates. Currently we are investigating how to scale both to a DL-component with  $10^3$  concepts and to a MIR component indexing  $10^6$  images.

## REFERENCES

- [1] G. Amato, P. Bolettieri, F. Debole, F. Falchi, F. Rabitti, and P. Savino. Using MILOS to build a multimedia digital library application: The PhotoBook experience. In *ECDL-06*, number 4172 in LNCS, pages 379–390. Springer Verlag, 2006.
- [2] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura. COMM: Designing a well-founded multimedia ontology for the web. In *ASWC-07*, number 4825 in LNCS, pages 30–43. Springer Verlag, 2007.
- [3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Data complexity of query answering in description logics. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR-06)*, pages 260–270, 2006.
- [5] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.
- [6] S. Hammiche, S. Benbernou, and A. Vakali. A logic based approach for the multimedia data representation and retrieval. In *Seventh IEEE International Symposium on Multimedia (ISM-05)*, pages 241–248. IEEE Computer Society, 2005.
- [7] J. S. Hare, P. A. S. Sinclair, P. H. Lewis, K. Martinez, P. G. B. Enser, and C. J. Sandom. Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches. In *ESWC-06*, number 4011 in LNCS. Springer Verlag, 2006.
- [8] C. Meghini, F. Sebastiani, and U. Straccia. A model of multimedia information retrieval. *Journal of the ACM*, 48(5):909–970, 2001.
- [9] I. MultiMedia. MPEG-7: The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87, 2002.
- [10] U. Straccia. Description logics with fuzzy concrete domains. In F. Bachus and T. Jaakkola, editors, *21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland, 2005. AUAI Press.
- [11] U. Straccia. Answering vague queries in fuzzy DL-Lite. In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-06)*, pages 2238–2245. E.D.K., Paris, 2006.
- [12] U. Straccia. Towards top-k query answering in description logics: the case of DL-Lite. In *JELIA-06*, number 4160 in LNCS, pages 439–451, Liverpool, UK, 2006. Springer Verlag.