DL-Media: an Ontology Mediated Multimedia Information Retrieval System

Umberto Straccia and Giulio Visco

ISTI-CNR
Pisa, ITALY,
straccia@isti.cnr.it

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 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.

1 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.

2 The Logic-based MIR Model in DL-MEDIA

Overall, DL-MEDIA lies in the context of *Logic-based Multimedia Information Retrieval* (LMIR) (see [16] for an extensive overview on LMIR literature. A recent work is also *e.g.* [13], see also [14] and [4] for a more complex multimedia ontology model). In DL-MEDIA, from each multimedia objects $o \in \mathbb{O}$ (such as pieces of text, images regions, 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. 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 variant of DLR-Lite like description logic with concrete domains.

3 The DL-MEDIA architecture

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

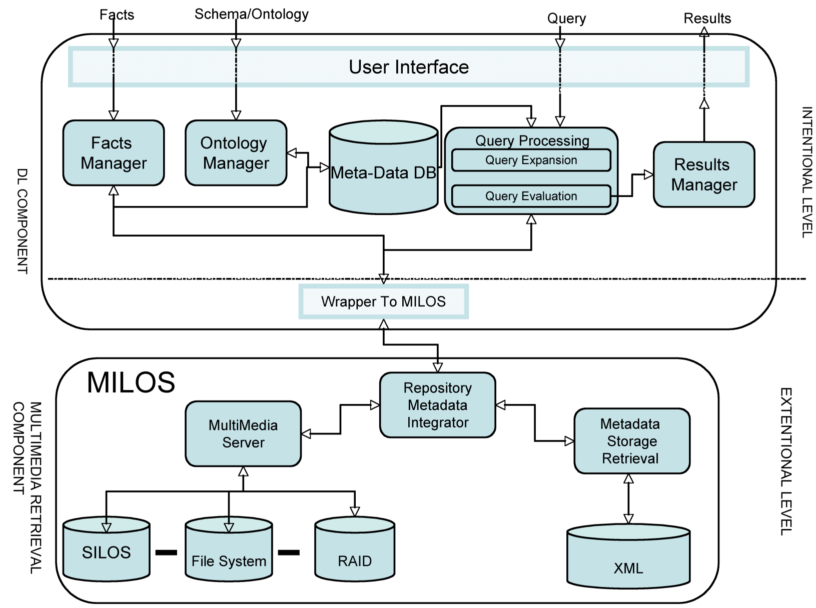

Fig. 1. DL-MEDIA architecture.

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 [8, 19, 18, 20] without negation.

The (feature-based) multimedia retrieval component, supports the retrieval of text and images based on low-level feature indexing. Specifically, we rely on our MIR system MILOS [1]. 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 [3, 2, 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 [21].

[1] `http://milos.isti.cnr.it/`

Operationally, a user submits a conceptual query (a 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 to be submitted to MILOS (that acts as a Web Service), which 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.

4 The DL-MEDIA query and representation language

For computational reasons the particular logic DL-MEDIA adopt is based on an extension of the DLR-Lite [8] Description Logic (DL) [5] without negation. 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 build-in predicates allowing to address three categories of retrieval: feature-based, semantic-based and their combination.

4.1 DL-MEDIA 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 is less or equal than 1500), but also similarity predicates such as $([i]\ simTxt\ 'logic, image, retrieval')$, 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 $\langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}} \rangle$, where $\Delta_{\mathtt{D}}$ is an interpretation domain and $\Phi_{\mathtt{D}}$ is the set of *domain predicates* d with a predefined arity n and an interpretation $d^{\mathtt{D}} \colon \Delta_{\mathtt{D}}^n \rightarrow [0, 1]$ (see also [17]). 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) and an alphabet of unary relations, called *atomic concepts* (and denoted A). An *axiom* is of the form

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

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*. They have the following syntax ($h \geq 1$):

$$\begin{array}{lll} Rr & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \\ Rl & \longrightarrow & A \mid \exists[i_1, \ldots, i_k]R \mid \exists[i_1, \ldots, i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_h) \\ Cond & \longrightarrow & ([i] \leq v) \mid ([i] < v) \mid ([i] \geq v) \mid ([i] > v) \mid ([i] = v) \mid ([i] \neq v) \mid \\ & & ([i]\ simTxt\ 'k_1, \ldots, k_n') \mid ([i]\ simImg\ URN) \end{array}$$

where A is an atomic concept, R is an n-ary relation with $1 \leq i_1, i_2, \ldots, i_k \leq n$, $1 \leq i \leq n$ and v is a value of the concrete interpretation domain of the appropriate type.

Informally, $\exists[i_1, \ldots, i_k]R$ is the projection of the relation R on the columns $i_1, \ldots, i_k$ (the order of the indexes matters). Hence, $\exists[i_1, \ldots, i_k]R$ has arity k.

On the other hand, $\exists[i_1,\ldots,i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l)$ further restricts the projection $\exists[i_1,\ldots,i_k]R$ 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. So, *e.g.* suppose we have a relation $\mathsf{Person}(\mathsf{firstname}, \mathsf{lastname}, \mathsf{age}, \mathsf{email}, \mathsf{sex})$ then

$$\exists[2,4]\mathsf{Person}.(([3] \geq 25))$$

corresponds to the set of tuples $\langle \mathsf{lastname}, \mathsf{email}\rangle$ such that the person's age is equal or greater than 25. Instead, $([i]\, simTxt\,'k_1 \ldots k_n')$ evaluates the degree of being the text of the i-th column similar to the list of keywords $k_1 \ldots k_n$, while $([i]\, simImg\, URN)$ 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* [2]). For instance, the following are axioms:

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

Note that in the last axiom, we require that the age is greater or equal than 18 and the gender is female. This axiom defines the relation $\mathsf{AdultMalePerson}(\mathsf{lastname}, \mathsf{firstname}, \mathsf{email})$. Examples axioms involving similarity predicates are,

$$(\exists[1]\mathsf{ImageDescr}.(([2]\,\mathsf{simImg}\,\mathsf{urn1}))) \sqcap (\exists[1]\mathsf{Tag}.(([2] = \mathsf{sunrise}))) \sqsubseteq \mathsf{Sunrise\_On\_Sea} \quad (1)$$

$$\exists[1]\mathsf{Title}.([2]\,\mathsf{simTxt}\,'\mathsf{lion}') \sqsubseteq \mathsf{Lion} \quad (2)$$

where $urn1$ identifies the image in Fig. 2. The former axiom (axiom 1) assumes that we have an ImageDescr relation, whose first column is the application specific image identifier and the second column contains the image URN. We use also a binary relation Tag. Then, this axiom (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) we assume that an image is annotated with a metadata format,

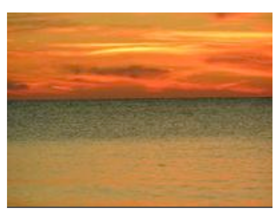

Fig. 2. Sun rise

e.g. MPEG-7, the attribute Title is seen as a binary relation, whose first column is the identifier of the metadata record, and the second column contains the title (piece of text) of the annotated image. Then, this axiom (informally) states that an image whose metadata record contains an attribute Title which is about 'lion' is about a Lion.

[2] http://en.wikipedia.org/wiki/Uniform_Resource_Name

A DL-MEDIA *ontology* $\mathcal{O}$ consists of a set of axioms.

Concerning queries, a DL-MEDIA *query* consists of a conjunctive query of the form

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

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 $R_1(\mathbf{z}_1) \wedge \ldots, \wedge R_l(\mathbf{z}_l)$ its *body*. $R_i(\mathbf{z_i})$ may also be a concrete unary predicate of the form $(z \leq v), (z < v), (z \geq v), (z > v), (z = v), (z \neq v), (z\, simTxt\, 'k_1, \ldots, k'_n), (z\, simImg\, URN)$, 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:

$\mathsf{q(x) \leftarrow Sunrise\_On\_Sea(x)}$
// find objects about a sunrise on the sea

$\mathsf{q(x) \leftarrow CreatorName(x, y) \wedge (y =' paolo') \wedge Title(x, z), (z\, simTxt\, 'tour')}$
// find images made by Paolo whose title is about 'tour'

$\mathsf{q(x) \leftarrow ImageDescr(x, y) \wedge (y\, simImg\, urn2)}$
// find images similar to a given image identified by $urn2$

$\mathsf{q(x) \leftarrow ImageObject(x) \wedge isAbout(x, y_1) \wedge Car(y_1) \wedge isAbout(x, y_2) \wedge Racing(y_2)}$
// 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 $R_1(\mathbf{z}_1) \wedge \ldots \wedge R_l(\mathbf{z}_l)$ 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.

4.2 DL-MEDIA semantics

From a semantics point of view, DL-MEDIA is based on mathematical fuzzy logic [12] 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 $\langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}}\rangle$, an *interpretation* $\mathcal{I} = \langle \Delta, \cdot^{\mathcal{I}}\rangle$ consists of a *fixed infinite domain* Δ, containing $\Delta_{\mathtt{D}}$, and an *interpretation function* $\cdot^{\mathcal{I}}$ that maps

- every atom A to a function $A^{\mathcal{I}} \colon \Delta \to [0, 1]$
- maps an n-ary predicate R to a function $R^{\mathcal{I}} \colon \Delta^n \to [0, 1]$
- constants to elements of Δ 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 Δ. 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 $\mathsf{Title}^{\mathcal{I}}$ 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 $\mathbf{c}[i_1,...,i_k]$ to denote the $i_1,\ldots,i_k$-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

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

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

$$\begin{array}{l} ([i]\, simTxt\,'k_1,\ldots,k_n')^{\mathcal{I}}(\mathbf{c}') = simTxt^{\mathsf{D}}(\mathbf{c}'[i],'k_1,\ldots,k_n') \in [0,1] \\ ([i]\, simImg\, URN)^{\mathcal{I}}(\mathbf{c}') = simImg^{\mathsf{D}}(\mathbf{c}'[i], URN) \in [0,1]\,. \end{array}$$

where $simTxt^{\mathsf{D}}$ and $simImg^{\mathsf{D}}$ 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 $Rl_1 \sqcap \ldots \sqcap Rl_m$. Usually, in fuzzy logic one uses a so-called T-norm $\otimes$ to combine the truth of "conjunctive" expressions [3] (see [12]). Some typical T-norms are

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

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

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

$$(\exists[i_1,\ldots,i_k]R)^{\mathcal{I}}(\mathbf{c}) = \sup_{\mathbf{c}' \in \Delta^n,\, \mathbf{c}'[i_1,\ldots,i_k]=\mathbf{c}} R^{\mathcal{I}}(\mathbf{c}')$$

$$\begin{array}{l} (\exists[i_1,\ldots,i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l))^{\mathcal{I}}(\mathbf{c}) = \\ \qquad \sup_{\mathbf{c}' \in \Delta^n,\, \mathbf{c}'[i_1,\ldots,i_k]=\mathbf{c}} \min(R^{\mathcal{I}}(\mathbf{c}'), Cond_1{}^{\mathcal{I}}(\mathbf{c}'),\ldots,Cond_l{}^{\mathcal{I}}(\mathbf{c}')) \end{array}$$

[3] 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 $x \otimes 1 = x$.

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

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

Note that for a fixed tuple $\mathbf{c}$ there may be several tuples $\mathbf{c}' \in R$ such that $\mathbf{c}'[i_1,\ldots,i_k] = \mathbf{c}$. 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 $\sup_{\mathbf{c}'\in\Delta^n,\,\mathbf{c}'[i_1,\ldots,i_k]=\mathbf{c}} R^{\mathcal{I}}(\mathbf{c}')$. The argument is similar for the $\exists[i_1,\ldots,i_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_l)$ 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 τ, denoted $\mathcal{I} \models \tau$, is defined as follows:

$$\mathcal{I} \models Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq Rr \text{ iff for all } \mathbf{c} \in \Delta^n, \min({Rl_1}^{\mathcal{I}}(\mathbf{c}),\ldots,{Rl_l}^{\mathcal{I}}(\mathbf{c})) \leq Rr^{\mathcal{I}}(\mathbf{c})\ ,$$

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.

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 $\mathbf{c} \in \Delta^n$:

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

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

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

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

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

From a query answering point of view, the DL-MEDIA system extends the DL-Lite/DLR-Lite reasoning method [8] to the fuzzy case. The algorithm is an extension of the one described in [8, 19, 18]). Roughly, given a query $q(\mathbf{x}) \leftarrow R_1(\mathbf{z}_1) \wedge \ldots \wedge R_l(\mathbf{z}_l)$,

1. 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;
2. from the set of reformulated queries $r(q, \mathcal{O})$ we remove redundant queries;
3. 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;
4. all the $n = |r(q, \mathcal{O})|$ top-k answer sets have to be merged into the unique top-k answer set $ans_k(\mathcal{O}, q)$. As $k \cdot n$ may be large, we apply the *Disjunctive Threshold Algorithm* (DTA, see [19] for the details) to merge all the answer sets.

In the appendix we provide a detailed description of the query reformulation procedure.

5 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 pane, the currently loaded ontology component $\mathcal{O}$ is shown. Below it and to the right, the current query is shown ("find a images 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 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.4) and excerpt of the metadata format is shown in Fig.5.

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.

6 Experiments

We conducted an experiment with the DL-MEDIA system. We considered an image set of around 560.000 images together with their metadata. The data has been provided by Flickr [4] as a courtesy and for experimental purposes only. In MILOS we have indexed the

[4] `http://www.flickr.com/`.

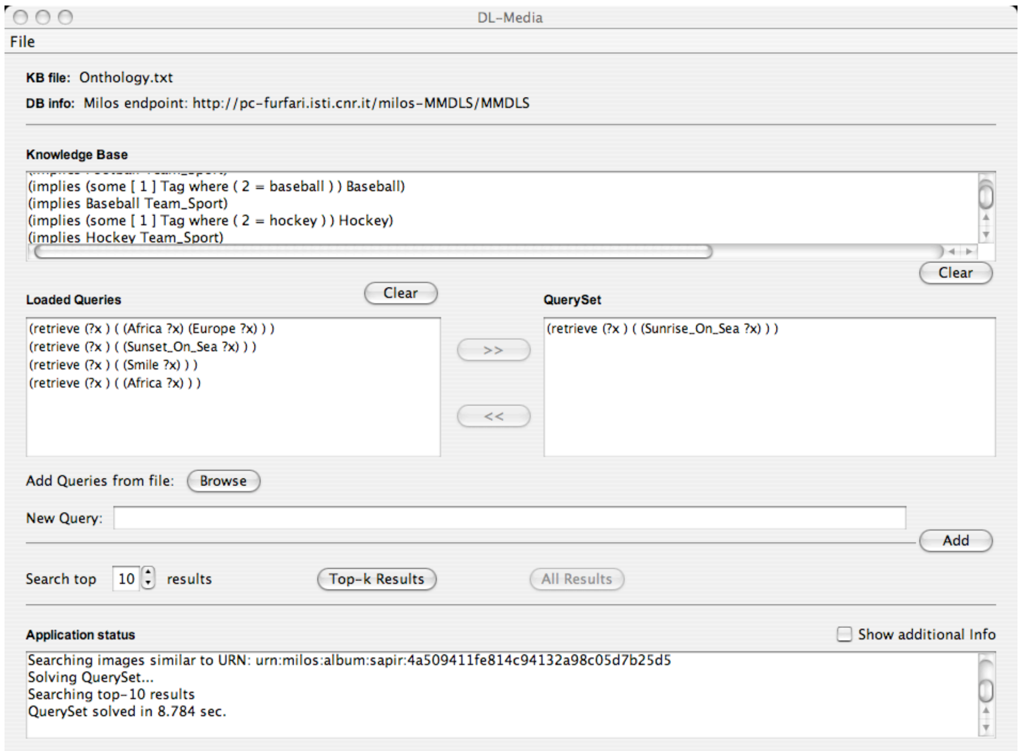

Fig. 3. DL-MEDIA main interface.

```
(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. 4. DL-MEDIA mapping rules.

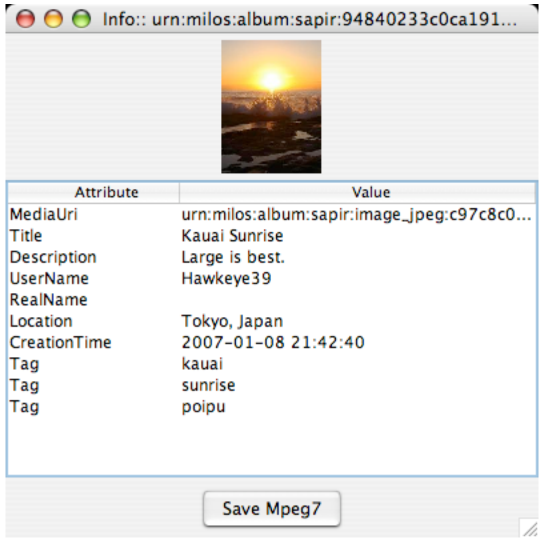

Fig. 5. Image metadata.

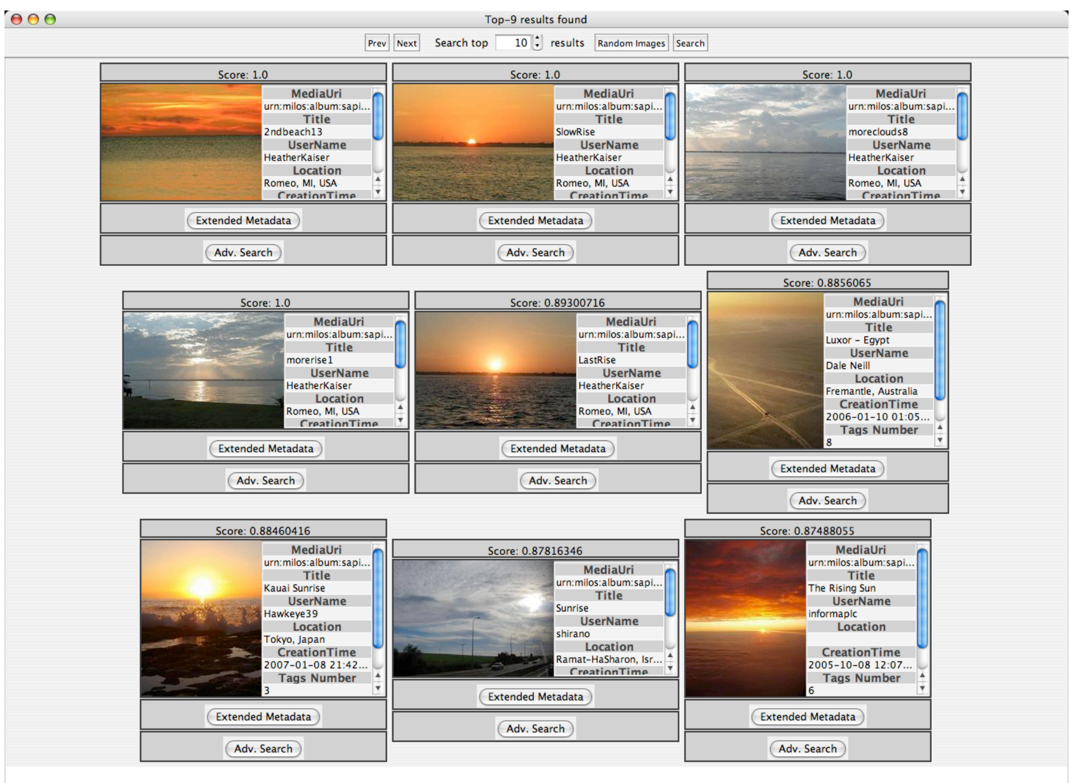

Fig. 6. DL-MEDIA results pane.

images' low-level features as 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

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'_{ref});
3. the number of reformulated queries after redundancy elimination (q_{ref});
4. the time of the reformulation process (t_{ref});
5. the number of queries effectively submitted to MILOS (q_{MILOS});
6. the query answering time of MILOS for each submitted query (t_{MILOS});
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}).

Query	Precision	q'_{ref}	q_{ref}	t_{ref}	q_{MILOS}	t_{MILOS}	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

Table 1. Experimental evaluation.

The results are shown in Table 1 below (time is measures in seconds). 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_{MILOS} may not coincide with $q_{ref} =$, 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 asses the effectiveness of the DL-MEDIA system. Worth noting is query 9

$$q_9(x) \leftarrow Europe(x) \wedge Africa(x)$$

in which we considered as relevant one image only, which dealt with a postcard sent from Johannesburg (South Africa) to Norwich (UK).

7 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:

- 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.
- multithreading of reformulated queries
- 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.

A Query answering in DL-MEDIA

At first, the input query q is *reformulated* into a set of conjunctive queries $r(q, \mathcal{O})$, by using $\mathcal{O}$ only. After having submitted the queries in $r(q, \mathcal{O})$ to MILOS, we merge the returned ranked lists using the DTA. The DTA is exactly the same as in [19] so we do not report it here, and restrict the presentation to the query reformulation step only.

Query reformulation. The query reformulation step is adapted from [8, 19, 18] to our case and is as follows.

We say that a variable in a conjunctive query is *bound* if it corresponds to either a distinguished variable or a shared variable, *i.e.*, a variable occurring at least twice in the query body, or a constant, while we say that a variable is *unbound* if it corresponds to a non-distinguished non-shared variable (as usual, we use the symbol "_" to represent non-distinguished non-shared variables). Note that an expression $\exists[i_1, \ldots, i_k]R$ can be seen as the Relation $R(\mathbf{x})$, where the variables in position $i_1, \ldots, i_k$ are unbound. We write also $R(\_, \ldots, \_, x_{i_1}, \ldots, x_{i_k}, \_, \ldots, \_)$ to denote the relation $R(\mathbf{x})$ in which all variables except those in position $i_1, \ldots, i_k$ are unbound. Given a vector of variables $\mathbf{x}$, and a condition $Cond$ occurring in the left-hand side of an axiom then $Cond(\mathbf{x})$ is defined as follows:

$$
\begin{aligned}
([i] \leq v)(\mathbf{x}) &= (x_i \leq v) \\
([i] < v)(\mathbf{x}) &= (x_i < v) \\
([i] \geq v)(\mathbf{x}) &= (x_i \geq v) \\
([i] > v)(\mathbf{x}) &= (x_i > v) \\
([i] = v)(\mathbf{x}) &= (x_i = v) \\
([i] \neq v)(\mathbf{x}) &= (x_i \neq v) \\
([i]\, simTxt\,'k_1, \ldots, k_n')(\mathbf{x}) &= (x_i\, simTxt\,'k_1, \ldots, k_n') \\
([i]\, simImg\, URN)(\mathbf{x}) &= (x_i\, simImg\, URN)\ .
\end{aligned}
$$

An axiom τ is *applicable* to an atom $A(x)$ in a query body, if τ has A in its right-hand side, while τ is applicable to an atom $R(\_, \ldots, \_, x_{i_1}, \ldots, x_{i_k}, \_, \ldots, \_)$ in a query body, if the right-hand side of τ is $\exists[i_1, \ldots, i_k]R$. We indicate with $gr(g; \tau)$ the expression obtained from the atom or relation g by applying the inclusion axiom τ.

Specifically,

- if $g = A(x)$ and τ is $Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq A$ then $gr(g; \tau)$ is $C_1(x) \wedge \ldots \wedge C_m(x)$, where for each $t \in \{1, \ldots, m\}$,
  - if $Rl_t = A_t$ then $C_t(x) = A_t(x)$;
  - if $Rl_t = \exists[j]R$ then $C_t(x) = \exists z_1, \ldots, z_l.R_t(\mathbf{z})$, where l is the cardinality of R and $\mathbf{z} = \langle z_1, \ldots, z_{j-1}, x, z_{j+1}, \ldots, z_l \rangle$;
  - if $Rl_t = \exists[i]R.(Cond_1 \sqcap \ldots \sqcap Cond_h)$ then $C_t(x) = \exists z_1, \ldots, z_l.R_t(\mathbf{z}) \wedge Cond_1(\mathbf{z}) \wedge \ldots \wedge Cond_h(\mathbf{z})$, where $\mathbf{z} = \langle z_1, \ldots, z_{j-1}, x, z_{j+1}, \ldots, z_l \rangle$ and l is the cardinality of R.
- if $g = R(\_, \ldots, \_, x_{i_1}, \ldots, x_{i_k}, \_, \ldots, \_)$ and τ is $Rl_1 \sqcap \ldots \sqcap Rl_m \sqsubseteq \exists[i_1, \ldots, i_k]R$ then $gr(g; \tau)$ is $C_1(x_{i_1}, \ldots, x_{i_k}) \wedge \ldots \wedge C_m(x_{i_1}, \ldots, x_{i_k})$, where for each $t \in \{1, \ldots, m\}$,
  - if $Rl_t = A_t$ and $k = 1$ then $C_t(x_{i_1}, \ldots, x_{i_k}) = A_t(x_{i_1}, \ldots, x_{i_k})$;
  - if $Rl_t = \exists[j_1, \ldots, j_k]R$ then $C_t(x_{i_1}, \ldots, x_{i_k}) = \exists z_1, \ldots, z_l.R_t(\mathbf{z})$, where l is the cardinality of R and $\mathbf{z}$ is such that the variables in position $j_1, \ldots, j_k$ are $x_{i_1}, \ldots, x_{i_k}$;
  - if $Rl_t = \exists[j_1, \ldots, j_k]R.(Cond_1 \sqcap \ldots \sqcap Cond_h)$ then $C_t(x_{i_1}, \ldots, x_{i_k}) = \exists z_1, \ldots, z_l.R_t(\mathbf{z}) \wedge Cond_1(\mathbf{z}) \wedge \ldots \wedge Cond_h(\mathbf{z})$, where l is the cardinality of R and $\mathbf{z}$ is such that the variables in position $j_1, \ldots, j_k$ are $x_{i_1}, \ldots, x_{i_k}$.

We are now ready to present the query reformulation algorithm. Given a query q and a set of axioms $\mathcal{O}$, the algorithm reformulates q in terms of a set of conjunctive queries $r(q, \mathcal{O})$, which then can be evaluated over the facts $\mathcal{A}$. In the algorithm, $q[g/g']$ denotes the query obtained from q by replacing the atom g with a new atom g'. At step 8, for each

Algorithm 1 `QueryRef`$(q, \mathcal{O})$

Input: A query q, DL-MEDIA axioms $\mathcal{O}$.
Output: Set of reformulated conjunctive queries $r(q, \mathcal{O})$.

1: $r(q, \mathcal{O}) := \{q\}$
2: **repeat**
3: $\quad S = r(q, \mathcal{O})$
4: $\quad$ **for all** $q \in S$ **do**
5: $\quad\quad$ **for all** $g \in q$ **do**
6: $\quad\quad\quad$ **if** $\tau \in \mathcal{O}$ is applicable to g **then**
7: $\quad\quad\quad\quad r(q, \mathcal{O}) := r(q, \mathcal{O}) \cup \{q[g/gr(g, \tau)]\}$
8: $\quad\quad$ **for all** $g_1, g_2 \in q$ **do**
9: $\quad\quad\quad$ **if** g_1 and g_2 unify **then**
10: $\quad\quad\quad\quad r(q, \mathcal{O}) := r(q, \mathcal{O}) \cup \{\kappa(reduce(q, g_1, g_2))\}$
11: **until** $S = r(\mathbf{q}, \mathcal{O})$
12: $r(q, \mathcal{O}) := removeSubs(r(q, \mathcal{O}))$
13: **return** $r(q, \mathcal{O})$

pair of atoms g_1, g_2 that unify, the algorithm computes the query $q' = reduce(q, g_1, g_2)$, by applying to q the most general unifier between g_1 and g_2 [5].

Due to the unification, variables that were bound in q may become unbound in q'. Hence, inclusion axioms that were not applicable to atoms of q, may become applicable to atoms of q' (in the next executions of step (5)). Function κ applied to q' replaces with $\_$ each unbound variable in q'. Finally, in step 12 we remove from the set of queries $r(q, \mathcal{O})$, those which are already subsumed in $r(q, \mathcal{O})$, in the sense that we remove q_1 from $r(q, \mathcal{O})$ if there is a $q_2 \in r(q, \mathcal{O})$ and a variable substitution θ such that for each predicate $P(\mathbf{z}_2)$ occurring in the rule body of q_2 there is a predicate $P(\mathbf{z}_1)$ occurring in the rule body of q_1 such that $P(\mathbf{z}_2) = P(\mathbf{z}_1)\theta$. This concludes the query reformulation step.

References

1. Giuseppe Amato, Paolo Bolettieri, Franca Debole, Fabrizio Falchi, Fausto Rabitti, and Pasquale Savino. Using milos to build a multimedia digital library application: The photobook experience. In *10th European Conference on Research and Advanced Technology for Digital Libraries*, number 4172 in Lecture Notes in Computer Science, pages 379–390. Springer Verlag, 2006.
2. Giuseppe Amato and Franca Debole. A native xml database supporting approximate match search. In *ECDL*, pages 69–80, 2005.
3. Giuseppe Amato, Claudio Gennaro, Fausto Rabitti, and Pasquale Savino. Milos: A multimedia content management system for digital library applications. In *Proceedings of the 8th European Conference Research and Advanced Technology for Digital Libraries (ECDL-04)*, pages 14–25, 2004.
4. Richard Arndt, Raphaël Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. Comm: Designing a well-founded multimedia ontology for the web. In *6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC-07, ASWC-07)*, number 4825 in Lecture Notes in Computer Science, pages 30–43. Springer Verlag, 2007.

[5] We say that two atoms $g_1 = r(x_1, \ldots, x_n)$ and $g_2 = r(y_1, \ldots, y_n)$ *unify*, if for all i, either $x_i = y_i$ or $x_i = \_$ or $y_i = \_$. If g_1 and g_2 unify, then the unification of g_1 and g_2 is the atom $r(z_1, \ldots, z_n)$, where $z_i = x_i$ if $x_i = y_i$ or $y_i = \_$, otherwise $z_i = y_i$ [7].

5. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
6. M.R. Boutell, J. Luo, X. Shen, and C.M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, September 2004.
7. Andrea Calì, Domenico Lembo, and Riccardo Rosati. Query rewriting and answering under constraints in data integration systems. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 16–21, 2003.
8. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo 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.
9. S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J.Y. Zien. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *The Twelfth International World Wide Web Conference (WWW-03)*, pages 178–186, Budapest, Hungary, 2003. ACM Press.
10. Yuli Gao and Jianping Fan. Semantic image classification with hierarchical feature subset selection. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval (MIR-05)*, pages 135–142, New York, NY, USA, 2005. ACM.
11. Theo Gevers and A. W. M. Smeulders. *Content-Based Image Retrieval: An Overview*, chapter 8. IMSC Press Multimedia Series. Prentice Hall, 1st edition, July 2004.
12. Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.
13. Samira Hammiche, Salima Benbernou, and Athena 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.
14. 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 *3rd European Semantic Web Conference (ESWC-06)*, number 4011 in Lecture Notes in Computer Science. Springer Verlag, 2006.
15. Raphaël Marée, Pierre Geurts, Giorgio Visimberga, Justus Piater, and Louis Wehenkel. An empirical comparison of machine learning algorithms for generic image classification. In F. Coenen, A. Preece, and A.L. Macintosh, editors, *Proceedings of the 23rd SGAI international conference on innovative techniques and applications of artificial intelligence, Research and development in intelligent systems XX*, pages 169–182. Springer, 2003.
16. Carlo Meghini, Fabrizio Sebastiani, and Umberto Straccia. A model of multimedia information retrieval. *Journal of the ACM*, 48(5):909–970, 2001.
17. Umberto Straccia. Description logics with fuzzy concrete domains. In Fahiem Bachus and Tommi Jaakkola, editors, *21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland, 2005. AUAI Press.
18. Umberto Straccia. Answering vague queries in fuzzy dl-lite. In *Proceedings of the 11th International Conference on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-06)*, pages 2238–2245. E.D.K., Paris, 2006.
19. Umberto Straccia. Towards top-k query answering in description logics: the case of DL-Lite. In *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA-06)*, number 4160 in Lecture Notes in Computer Science, pages 439–451, Liverpool, UK, 2006. Springer Verlag.
20. Umberto Straccia and Giulio 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.
21. Pavel Zezula, Giuseppe Amato, Vlastislav Dohnal, and Michal Batko. *Similarity Search: The Metric Space Approach (Advances in Database Systems)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.