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 \( o \in \mathcal{O} \) (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 \( o \in \mathcal{O} \) 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 \( o \in \mathcal{O} \). 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).

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 \(^1\), 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-

---

\(^1\)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 DLMEDIA 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. DLMEDIA supports concrete domains with specific predicates on it. The concrete predicates that DLMEDIA 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] 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
(\Delta_0, \Phi_0), where \Delta_0 is an interpretation domain and \Phi_0 is the set of domain predicates d with a predefined arity n
and an interpretation \Phi : \Delta_0^n \to [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

\[ R_1 \cap \ldots \cap R_{m} \subseteq R_{r} \]

where \( m \geq 1 \), all \( R_i \) and \( R_r \) have the same arity and where
each \( R_i \) is a so-called left-hand relation and \( R_r \) is a right-hand
relation. Informally, such an axiom may be read as “if \( R_1 \) and \( R_2 \) \ldots and \( R_m \) then \( R_r \)”. As illustrative purpose,
a simple ontology axiom may be of the form ItalianCity \subseteq
EuropeanCity with informal reading “any italian city is an
european city”. Here ItalianCity and EuropeanCity are unary
relations with signature ItalianCity(id) and EuropeanCity(id),
respectively. Similarly, the ontology axiom ItalianCity \cap
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 signa-
ture 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 title image 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 \( i_1, \ldots, i_k \)
(\( 1 \leq i_1, i_2, \ldots, i_k \leq n, 1 \leq i \leq n \)), will be indicated with
\( \exists_{i_1, \ldots, i_k} R \). 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

\[ \exists_{[1, 3]} \text{MyMetadata} \sqsubseteq \exists_{[3, 2]} \text{ImageDescr} \]
\[ \exists_{[1, 4]} \text{MyMetadata} \sqsubseteq \exists_{[3, 2]} \text{Tag} \]
\[ \exists_{[1, 2]} \text{MyMetadata} \sqsubseteq \exists_{[3, 2]} \text{Title} \]

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.([([3] \geq 25)] corresponds to the set of tuples
(lastname,email) 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

\[ \exists_{[2,3]} \text{Person} \sqsubseteq \exists_{[1,2]} \text{hasAge} \]
\[ \exists_{[2,4]} \text{Person} \sqsubseteq \exists_{[1,2]} \text{hasEmail} \]
\[ \exists_{[2,1,4]} \text{Person.([([3] \geq 18] \cap ([5] = 'male'))} \sqsubseteq \exists_{[1,2,3]} \text{AdultMalePerson} \]

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 \( ([i] \text{simTxt}'k_1 \ldots k_n) \) evaluates the degree of
being the text of the i-th column similar to the list of keywords
\( k_1 \ldots k_n \), while \( ([i] \text{simImg} \text{URN}) \) 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) \(^2\). Examples axioms involving similarity conditions are,

\[ \exists_{[1]} \text{ImageDescr.([([3] \text{simImg urn1])}) \sqsubseteq \exists_{[1]} \text{Tag.([([2] = 'sunrise')}) \]
\[ \exists_{[2]} \text{On} \text{Sea} \]

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 \text{Sunrise} \text{On} \text{Sea} (to a system computed degree in [0, 1]).
Similarly, in axiom (2) informally states that an image whose

\(^2\)http://en.wikipedia.org/wiki/Uniform_Resource_Name
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 \( h \geq 1 \)):

\[
\begin{align*}
R_r & \quad \rightarrow \quad \text{A} | \exists[i_1, \ldots, i_k] R \\
R_l & \quad \rightarrow \quad \text{A} | \exists[i_1, \ldots, i_k] R | \\
& \quad \quad \quad \quad \exists[i_1, \ldots, i_k] R, (\text{Cond} \_1 \land \ldots \land \text{Cond} \_h) \\
\text{Cond} & \quad \rightarrow \quad ([i] \leq v) \lor ([i] < v) \lor ([i] \geq v) \lor ([i] > v) \lor \\
& \quad \quad \quad \quad ([i] = v) \lor ([i] \neq v) \lor \\
& \quad \quad \quad \quad ([i] \sim \text{Txt} \_k_1, \ldots, k_n) \lor ([i] \sim \text{Img} \_\text{URN})
\end{align*}
\]

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. Here \( \exists[i_1, \ldots, i_k] R \) 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, \( \exists[i_1, \ldots, i_k] R, (\text{Cond} \_1 \land \ldots \land \text{Cond} \_h) \) further restricts the projection \( \exists[i_1, \ldots, i_k] R \) according to the conditions specified in \( \text{Cond} \_h \). For instance, \( ([i] \leq v) \) specifies that the values of the \( i \)-th column have to be less or equal than the value \( v \). \( ([i] \sim \text{Txt} \_k_1, \ldots, k_n) \) evaluates the degree of being the text of the \( i \)-th column similar to the list of keywords \( k_1, \ldots, k_n \), while \( ([i] \sim \text{Img} \_\text{URN}) \) 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 \( O \) consists of a set of axioms.

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

\[ q(x) \leftarrow R_1(z_1) \land \ldots \land R_n(z_n), \]

where \( q \) is an \( n \)-ary predicate, every \( R_i \) is an \( i \)-ary predicate, \( x \) is a vector of variables, and every \( z_q \) is a vector of constants, or variables. We call \( q(x) \) its head and \( R_1(z_1) \land \ldots \land R_n(z_n) \) its body. \( R_i(z_i) \) may also be a concrete unary predicate of the form \( (z \leq v), (z < v), (z \geq v), (z > v), (z = v), (z \neq v), (z \sim \text{Txt} \_k_1, \ldots, k_n) \), \( (z \sim \text{Img} \_\text{URN}) \), where \( z \) is a variable, \( v \) is a value of the appropriate concrete domain, \( k_i \)

is a keyword and \( \text{URN} \) is an URN. Example queries are:

\[ q(x) \leftarrow \text{Sunrise}_\text{On}_\text{Sea}(s) \quad // \text{find objects about a sunrise on the sea} \]

\[ q(x) \leftarrow \text{CreatorName}(x, y) \land (y \text{ is} \text{‘paolo’}) \land \text{Title}(x, z), (z \sim \text{Txt} \text{‘tour’}) \quad // \text{find images made by Paolo whose title is about ‘tour’} \]

\[ q(x) \leftarrow \text{ImageDescr}(x, y) \land (y \sim \text{Img} \_\text{URN}2) \quad // \text{find images similar to a given image identified by \text{urn2}} \]

\[ q(x) \leftarrow \text{ImageObject}(x) \land \text{isAbout}(x, z) \land \text{Car}(y_1) \land \text{isAbout}(x, y_2) \land \text{Racing}(y_2) \quad // \text{find image objects about cars racing} \]

We note that a query may also be written as

\[ q(x) \leftarrow \exists y \phi(x, y), \]

where \( \phi(x, y) \) is \( R_1(z_1) \land \ldots \land R_n(z_n) \) and no variable in \( y \) occurs in \( x \) and vice-versa. Here, \( x \) are the so-called distinguished variables, while \( y \) are the so-called non distinguished variables, which are existentially quantified. For a query atom \( q \), we will write \( \langle q(c), s \rangle \) to denote that the tuple \( c \) 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 \( (\Delta_0, \Phi_0) \), an interpretation \( I = (\Delta, \mathcal{I}) \) consists of a fixed infinite domain \( \Delta \), containing \( \Delta_0 \), and an interpretation function \( \mathcal{I} \) 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. \( R(c) \)) 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 \( c \), \( R^\mathcal{I}(c) \) determines to which degree the tuple \( c \) 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 Title\(^\mathcal{I} \) is of type String. To the ease of presentation, we omit the formalization of this aspect and leave it at the intuitive level.

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

Concerning concrete comparison predicates, the interpretation function \( \mathcal{I} \) has to satisfy

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

331
and similarly for the other comparison constructs, \((i[i < v], (i[i ≥ v], (i[i > v] \text{ and } \langle i[i = v \mid \langle i \neq v).}

Concerning the concrete similarity predicates, the interpretation function \(\mathcal{I}\) has to satisfy

\[
\begin{align*}
(i)[\text{simTxt}^\phi k_1, \ldots, k_n^\phi] & = \text{simTxt}^\phi (\varepsilon[i], k_1, \ldots, k_n) \in [0, 1] \\
(i)[\text{simImg}^\phi URN] & = \text{simImg}^\phi (\varepsilon[i], URN) \in [0, 1].
\end{align*}
\]

where \(\text{simTxt}^\phi\) and \(\text{simImg}^\phi\) are the textual and image similarity predicates supported by the underlying MIR system MILOS.

Concerning axioms, as in an interpretation each \(Rl_i(\varepsilon)\) has a degree of truth, we have to specify how to combine them to determine the degree of truth of the conjunction \(Rl_1 \land \ldots \land Rl_m\). Usually, in mathematical fuzzy logic one uses a so-called T-norm \(\otimes\) to combine the truth of “conjunctive” expressions \(^3\) (see [5]). Some typical T-norms are

\[
\begin{align*}
x \otimes y & = \min(x, y) \quad \text{Gödel conjunction} \\
x \otimes y & = \max(x + y - 1, 0) \quad \text{Łukasiewicz conjunction} \\
x \otimes y = x \cdot y \quad \text{Product conjunction}.
\end{align*}
\]

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

The interpretation function \(\mathcal{I}\) has to satisfy: for all \(\varepsilon \in \Delta^n\) and \(n\)-ary relation \(R:\n
\[
\begin{align*}
(i)[\exists i_1, \ldots, i_k R]^\mathcal{I}(\varepsilon) & = \sup_{\varepsilon'[\Delta^n, \varepsilon'[i_1, \ldots, i_k] = \varepsilon} R^\mathcal{I}(\varepsilon') \\
(i)[\exists i_1, \ldots, i_k R.(\text{Cond}_1 \land \ldots \land \text{Cond}_l)]^\mathcal{I}(\varepsilon) & = \sup_{\varepsilon' \in \Delta^n, \varepsilon'[i_1, \ldots, i_k] = \varepsilon} \min(R^\mathcal{I}(\varepsilon'), \text{Cond}_1^\mathcal{I}(\varepsilon'), \ldots, \text{Cond}_l^\mathcal{I}(\varepsilon'))
\end{align*}
\]

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

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

Note that for a fixed tuple \(\varepsilon\) there may be several tuples \(\varepsilon' \in R\) such that \(\varepsilon'[i_1, \ldots, i_k] = \varepsilon\). Now, if we switch to fuzzy logic, for a fixed tuple \(\varepsilon\) and interpretation \(\mathcal{I}\), each of the previous mentioned \(\varepsilon'\) is instance of \(R\) to a degree \(R^\mathcal{I}(\varepsilon')\). 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 \(\sup_{\varepsilon' \in \Delta^n, \varepsilon'[i_1, \ldots, i_k] = \varepsilon} R^\mathcal{I}(\varepsilon')\). The argument is similar for the \(\exists[i_1, \ldots, i_k] R.(\text{Cond}_1 \land \ldots \land \text{Cond}_l)\) 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 \land \ldots \land Rl_m \iff \text{for all } \varepsilon \in \Delta^n, \min(Rl_1^\mathcal{I}(\varepsilon), \ldots, Rl_m^\mathcal{I}(\varepsilon)) \leq R^\mathcal{I}(\varepsilon), \text{where we assume that the arity of } R \text{ and all } Rl_i \text{ is } n. \text{An interpretation } \mathcal{I} \text{ is a model of (satisfies) an ontology } O \iff \text{it satisfies each element in it.}

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

\[
\phi^\mathcal{I}(\varepsilon) \geq \sup_{\varepsilon' \in \Delta^n} \phi^\mathcal{I}(\varepsilon'),
\]

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

\[
glb(O, q(\varepsilon)) = \sup\{s \mid O \models (q(\varepsilon), s)\}.
\]

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

**Query answering.** From a query answering point of view, the DL-MEDIA system extends the DL-Lite/DL-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 \(q(\varepsilon) \iff Rl_1(z_1) \land \ldots \land Rl_n(z_n)\), (i) by considering \(O\), the user query \(q\) is reformulated into a set of conjunctive queries \(r(q, O)\). 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 \(q(\varepsilon) \iff A(x)\) and suppose that \(O\) contains the axioms \(B_1 \subseteq A\) and \(B_2 \subseteq A\), then we can reformulate the query into two queries \(q_1(\varepsilon) \iff B_1(\varepsilon)\) and \(q_2(\varepsilon) \iff B_2(\varepsilon)\), exactly as it happens for top-down resolution methods in logic programming; (ii) from the set of reformulated queries \(r(q, O)\) we remove redundant queries; (iii) the reformulated queries \(q' \in r(q, O)\) 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 \(n = |r(q, O)|\) top-k answer sets have to be merged into the unique top-k answer set \(ans_k(O, q)\). 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 \(O\) 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

\[322\]
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\(^4\) as a courtesy and for experimental purposes only. In MILOS we have indexed the 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.

\(^4\)http://www.flickr.com/.
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 ($q_g$);
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 ($t_{IM}$);
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).

<table>
<thead>
<tr>
<th>Query</th>
<th>$Pr_m$</th>
<th>$q_r$</th>
<th>$t_r$</th>
<th>$q_M$</th>
<th>$t_{IM}$</th>
<th>$t_{DTA}$</th>
<th>$t_{IMG}$</th>
<th>$t_{tot}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>1.0</td>
<td>2</td>
<td>2</td>
<td>0.085</td>
<td>1</td>
<td>0.3</td>
<td>0.613</td>
<td>1.095</td>
</tr>
<tr>
<td>Q2</td>
<td>0.9</td>
<td>4</td>
<td>1</td>
<td>0.120</td>
<td>1</td>
<td>0.427</td>
<td>0.656</td>
<td>1.179</td>
</tr>
<tr>
<td>Q3</td>
<td>0.7</td>
<td>9</td>
<td>1</td>
<td>0.011</td>
<td>1</td>
<td>0.395</td>
<td>0.514</td>
<td>0.996</td>
</tr>
<tr>
<td>Q4</td>
<td>0.5</td>
<td>10</td>
<td>6</td>
<td>0.111</td>
<td>1</td>
<td>0.537</td>
<td>0.614</td>
<td>1.355</td>
</tr>
<tr>
<td>Q5</td>
<td>0.3</td>
<td>4</td>
<td>2</td>
<td>0.056</td>
<td>1</td>
<td>0.286</td>
<td>0.286</td>
<td>0.286</td>
</tr>
<tr>
<td>Q6</td>
<td>0.2</td>
<td>2</td>
<td>3</td>
<td>0.068</td>
<td>1</td>
<td>0.004</td>
<td>0.068</td>
<td>0.072</td>
</tr>
<tr>
<td>Q7</td>
<td>0.05</td>
<td>22</td>
<td>4</td>
<td>0.511</td>
<td>1</td>
<td>1.451</td>
<td>0.059</td>
<td>1.451</td>
</tr>
<tr>
<td>Q8</td>
<td>0.05</td>
<td>35</td>
<td>2</td>
<td>0.056</td>
<td>20</td>
<td>0.057</td>
<td>0.057</td>
<td>20.181</td>
</tr>
<tr>
<td>Q9</td>
<td>0.07</td>
<td>37</td>
<td>36</td>
<td>0.056</td>
<td>20</td>
<td>0.057</td>
<td>0.057</td>
<td>20.181</td>
</tr>
</tbody>
</table>

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