Modelling the Retrieval of Structured Documents Containing Texts and Images

Carlo Meghini, Fabrizio Sebastiani and Umberto Straccia

Consiglio Nazionale delle Ricerche,
Istituto di Elaborazione della Informazione,
Via S. Maria 46, I-56123 Pisa, Italy,
E-mail: {meghini,fabrizio,straccia}@iei.pi.cnr.it
Phone: +39 50 593405, Fax: +39 50 554342

Abstract. We present a model for complex documents possibly consisting of a hierarchically structured set of images or texts. Documents are represented both at the *form* level (as sets of physical features of the *representing* objects), at the content level (as sets of properties of the *represented* entities), and at the *structure* level. A uniform and powerful query language allows queries to be issued that transparently combine features pertaining to form, content and structure alike. Queries are expressions of a (fuzzy) logical language. While that part of the query that pertains to (medium-independent) content is "directly" processed by an inferential engine, that part that pertains to (medium-dependent) form is entrusted to specialised document processing procedures linked to the logical language by a *procedural attachment* mechanism. The model thus combines the power of state-of-the-art document processing techniques with the advantages of a clean, logically defined framework for understanding multimedia document retrieval.

1 Introduction

Research on multimedia document (MD) retrieval is still in its early stages, due to the inherent difficulty of indexing documents pertaining to media other than text in a way that faithfully reflects their information content and, as a consequence, that significantly impacts on retrieval. Nonetheless, a number of retrieval systems for media other than text have been built (see e.g. [9]) and, in some cases, even turned into commercial products [3, 7]. We think that the common trait of these multimedia retrieval systems (MRSs) and of the underlying research models is the lack of a proper representation and use of the *content* of non-textual documents: only features pertaining to their *form*, being amenable to automatic extraction through digital signal processing (DSP) techniques, are used upon retrieval. But this is disturbing, as documents, irrespective of the representation medium they employ, should properly be regarded as *information carriers*, and as such should be considered along two parallel dimensions, that of *form* (or *syntax*, or *symbol*) and that of *content* (or *semantics*, or *meaning*).

We present a model where MDs are represented both at the form level (as sets of physical features of the objects *representing* a slice of the world) and at

the content level (as sets of properties of the real-world objects *represented*). At the form level the representation is *medium-dependent*, so we envisage (and allow for) different document processing techniques in order to specifically deal with each different media the sub-documents are expressed in. At the content level, instead, the representation is *medium-independent*, and a unique language for content representation is thus adopted. This model deals with complex documents possibly consisting of a hierarchically structured set of "atomic" sub-documents, which may in turn be either images or chunks of text. Besides allowing a rich representation of these atomic sub-documents, the model also allows the explicit representation of the hierarchical structure of the document. Features pertaining to form, content and structure alike may thus participate in the representation and in queries, thus corroborating the view of a document as a multifaceted entity. Actually, we deem important a fourth, orthogonal facet of documents, namely its *profile*; for reasons of space deal with it only in the full paper. Also, although we only consider images and text, the way this model enforces the interaction between these two media is illustrative of how other media might also be allowed in.

The model we present here is logic-based, in the sense that our language for content representation is based on a *description logic* (DL)[1]. Although document properties pertaining to form are not represented explicitly in the DL, they affect the DL-based reasoning through a mechanism of *procedural attachments*. That is, those symbols of the DL that pertain to form may be viewed as calls to (non-logical) routines that implement the document processing (e.g. DSP) techniques specific to the representation medium at hand, thus *computing* (rather than logically inferring) form-related document properties. This implements the connection between (logical) reasoning about content and (non-logical) reasoning about form from a "practical" point of view. From the standpoint of the *semantics* of the representation and query languages, this latter connection is instead established by restricting the set of interpretations of the logical language to those that verify the constraints imposed at form level by the results of the document processing analysis. This mechanism for giving semantics to procedural attachments is known as the *method of concrete domains* [2].

Our DL-based query language thus allows the expression of retrieval requests addressing, among other things, both form- and content-related similarity, and its underlying logic permits to bring to bear domain knowledge (whose representation DLs are especially good at) in the retrieval process. The query language also includes facilities for fuzzy reasoning (which actually make it a full-fledged *fuzzy DL*) so as to address the inherently quantitative nature of notions like "similarity" between images/text or between their features (colour, shape, morphology, and the like). The model is extensible, in that the set of symbols representing similarity can be enriched at will to account for different, possibly medium-specific notions of similarity, and for different methods for computing

[1] DLs have already been applied to IR in [12]; we believe this work is different, however, in that rather than on the tool itself we focus more on the way to make concrete use of it and to link it to other extant MD retrieval technology.

them. The resulting retrieval model thus extends that of current MRSs with the use of semantic information processing and reasoning about image/text content. So far, the only attempts in this direction had been based on textual annotations to non-textual documents (see e.g. [17]), in some cases supported by the use of thesauri to semantically connect the terms occurring in the text [10]; this means that text is seen as mere comment on the non-textual document, and not as an object of independent interest and therefore subject to retrieval *per se*. In our model images and text are both first-class citizens, and this clearly indicates how the extension to other media could be accomplished.

The paper is organised as follows. Section 2 concisely introduces the fuzzy DL that will constitute our main tool throughout the paper. Sections 3 to 5 deal with the aspects of documents that our model addresses (namely: form, content and structure); for each of them we first discuss issues related to modelling and then switch to the semantics of the related query facilities. Section 6 presents a unified, hierarchically structured query language which brings together all the issues discussed in Sections 3 to 5. In Section 7 we deal with retrieval and show how the degree of relevance of a document to a query may be seen in terms of the fuzzy DL that underlies both the representation and query languages. We conclude by briefly touching on issues of implementation, while leaving the discussion on the computational complexity of the model to the full paper.

2 Fuzzy $\mathcal{ALC}$

The formalism we have chosen for representing and querying document contents is a *Description Logic* (DL – see e.g. [4]). DLs are contractions of first order logic (FOL), and have an "object-oriented" character that makes them especially suitable for reasoning about hierarchies of structured objects. The specific DL that we use in this paper is $\mathcal{ALC}$ [16]; however, we stress that our model is not tied in any way to this particular choice, and any other (possibly much more expressive) DL would easily fit into it[2]. The language of $\mathcal{ALC}$ includes unary and binary predicate symbols, called *primitive concepts* (indicated by the metavariable A with optional subscripts) and *primitive roles* (metavariable R), respectively. These are the basic constituents by means of which *concepts* (metavariable C), i.e. "non-primitive unary predicate symbols", are built via *concept constructors*[3] according to the BNF rule $C \longrightarrow A \mid C_1 \sqcap C_2 \mid \neg C \mid \forall R.C$.

For example, the complex concept $\texttt{Person} \sqcap \forall \texttt{Friend}.\neg \texttt{Musician}$ is obtained by combining the primitive concepts `Person` and `Musician` and the primitive

[2] The reason why we have opted for $\mathcal{ALC}$ is that it is universally considered the "minimal" DL (as much as K is considered the "minimal" modal logic) and is therefore regarded as the most convenient workbench for carrying out logical work of an experimental nature. Reverting to one's DL of choice may then be considered the very last (and usually straightforward) step in the development of a logical DL-based model.

[3] This DL does not contain *role constructors*; thus, the terms "primitive role" and "role" are equivalent in $\mathcal{ALC}$.

role `Friend` by the conjunction ($\sqcap$), universal quantification ($\forall$) and negation ($\neg$) constructors, and denotes the persons none of whose friends are musicians. As customary, disjunction $C_1 \sqcup C_2$ and existential quantification $\exists R.C$ will be used as abbreviations of expressions $\neg(\neg C_1 \sqcap \neg C_2)$ and $\neg(\forall R.\neg C)$, respectively. The language of $\mathcal{ALC}$ also includes *(crisp) assertions*, i.e. expressions built out of concepts, roles and *individual constants* (metavariable a with an optional subscript) according to the following BNF rules:

1. $C(a)$, meaning that a is an instance of C; `(Musician ⊓ Teacher)(tim)` makes the individual constant `tim` a `Musician` and a `Teacher`;
2. $R(a_1, a_2)$, meaning that a_1 is related to a_2 by means of R (e.g. `Friend(tim,tom)`);
3. $T \sqsubseteq T'$, where T and T' are both concepts or both roles, meaning that T is more specific than T' (e.g. `PianoPlayer` $\sqsubseteq$ (`Musician` $\sqcap \exists$`Plays.Keyboard`)).

Assertions of type 1 and 2 are called *simple assertions*, while assertions of type 3 are called *axioms*. In order to deal with the uncertainty inherent in similarity-based retrieval, we extend $\mathcal{ALC}$ with *fuzzy assertions* (see e.g. [5]), i.e. expressions of the form $\langle \alpha, n \rangle$ where α is a crisp assertion and $n \in [0, 1]$, meaning that α holds "to degree n"[4]. We will use the terms *fuzzy simple assertion* and *fuzzy axiom* with the obvious meaning, and call the resulting logic *fuzzy* $\mathcal{ALC}$.

The semantics of fuzzy $\mathcal{ALC}$ is based on *fuzzy interpretations*, i.e. pairs $\mathcal{I} = (\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty set (the *domain of discourse*) and $(\cdot)^{\mathcal{I}}$ is a function (the *interpretation function*) mapping i) each concept into a function from $\Delta^{\mathcal{I}}$ to $[0, 1]$, ii) each role into a function from $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to $[0, 1]$, and iii) each individual constant into $\Delta^{\mathcal{I}}$ in such a way that, for all $d \in \Delta^{\mathcal{I}}$, $(C_1 \sqcap C_2)^{\mathcal{I}}(d) = min\{{C_1}^{\mathcal{I}}(d), {C_2}^{\mathcal{I}}(d)\}$, $(\neg C)^{\mathcal{I}}(d) = 1 - C^{\mathcal{I}}(d)$ and $(\forall R.C)^{\mathcal{I}}(d) = min_{d' \in \Delta^{\mathcal{I}}}\{max\{1 - R^{\mathcal{I}}(d, d'), C^{\mathcal{I}}(d')\}\}$. Note that the condition for the "$\forall$" constructor is obtained by interpreting universal quantification as infinite conjunction.

A fuzzy interpretation $\mathcal{I}$ is said to be a *model* of a fuzzy assertion $\langle C(a), n \rangle$ iff $C^{\mathcal{I}}(a^{\mathcal{I}}) \geq n$, of a fuzzy assertion $\langle R(a_1, a_2), n \rangle$ iff $R^{\mathcal{I}}({a_1}^{\mathcal{I}}, {a_2}^{\mathcal{I}}) \geq n$, and of a fuzzy assertion $\langle T \sqsubseteq T', n \rangle$ iff $min_{d \in \Delta^{\mathcal{I}}}\{max\{1 - T^{\mathcal{I}}(d), {T'}^{\mathcal{I}}(d)\}\} \geq n$ for all $d \in \Delta^{\mathcal{I}}$. A set of fuzzy assertions Σ is said to *entail* a fuzzy assertion $\langle \alpha, n \rangle$ (written $\Sigma \models^f \langle \alpha, n \rangle$) iff all models of all fuzzy assertions in Σ are models of $\langle \alpha, n \rangle$. Given Σ and a crisp assertion β, the *maximal degree of truth* of β w.r.t. Σ (written $Maxdeg(\Sigma, \beta)$) is defined as the $n \in [0, 1]$ such that $\Sigma \models^f \langle \beta, n \rangle$ and there is no $m > n$ such that $\Sigma \models^f \langle \beta, m \rangle$.

The pivotal role that fuzzy $\mathcal{ALC}$ has in the context of our model will become clear in the next sections. The connection between logical reasoning in fuzzy $\mathcal{ALC}$ and non-logical computation through medium-specific document processing

[4] The logic we are actually experimenting with is more expressive than this, as it includes features for selective closed-world reasoning and for inconsistency-tolerant, shallow reasoning; we omit discussion of these features for brevity. The interested reader may refer to [13].

techniques will be realised by identifying a number of *special* $\mathcal{ALC}$ *predicate symbols* (SPSs) and imposing that their semantics be not a *generic* subset of $\Delta^{\mathcal{I}}$ (or $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$) but one that complies with the results of the document processing analysis.

3 Form

We now proceed to discussing the "form" dimension of images and text; we present models for *image layouts* and *text layouts*, which consist of the symbolic representations of the form-related aspects of an image or text, respectively. Each notion is endowed with a *mereology*, i.e. a theory of parts, based on notions such as *atomic region*, *region* and *grounded region*[5]. We also introduce SPSs for querying such models, which will be used in the unified query language discussed in Section 6. The reader will note the evident parallelism, even down to many details, in our treatment of image form and text form; given that form is the only medium-specific aspect of documents, this shows the potential of this model for extension to other media.

3.1 Image Layouts

Modelling Image Layouts We recall some elementary notions from digital geometry (see e.g. [15, Chapter 11]). Let $\mathbb{N}$ be the set of natural numbers. A *zone* is any subset of $\mathbb{N}^2$, i.e. a set of *points*. A zone S is *aligned* if $min\{x \mid \langle x, y\rangle \in S\} = 0$ and $min\{y \mid \langle x, y\rangle \in S\} = 0$. The *neighbours* of a point $P = \langle x, y\rangle$, when both x and y are non-zero, are the points $\langle x-1, y\rangle$, $\langle x, y-1\rangle$, $\langle x, y+1\rangle$, and $\langle x+1, y\rangle$. If only one of P's coordinates is 0, then P has only 3 neighbours; $\langle 0, 0\rangle$ has only two neighbours. Two zones are said to be *neighbour* to each other if they are disjoint and a point in one of them is a neighbour of a point in the other one. A *path* of length n from point P to point P' is a sequence of points $P = P_0, P_1, \ldots, P_n = P'$ such that P_i is a neighbour of P_{i-1}, $1 \leq i \leq n$. Let S be a zone and P and P' points of S: P is *connected* to P' in S if there is a path from P to P' consisting entirely of points of S. For any P in S, the set of points that are connected to P in S is called a *connected component* of S. If S has only one component, it is called a *simply connected* zone.

Given a set of *colours* $\mathcal{C}$, an *image layout* (see also [11]) is a triple $i = \langle A^i, \pi^i, f^i\rangle$, where A^i, the *domain*, is a finite, aligned, rectangular zone; π^i is a partition of A^i into non-empty connected zones $\{T_1, ..., T_n\}$, called *atomic regions*; f^i is a total function (the *colour function*) from π^i to $\mathcal{C}$ assigning a colour to each atomic region in such a way that no two neighbour atomic regions have the same colour. The *regions* of an image layout $i = \langle A^i, \pi^i, f^i\rangle$ are defined as the set $\pi^i_e = \{S \mid \exists T_1, ..., T_k \in \pi^i, k \geq 1,\ S = \cup_{i=1}^{k} T_i,\ S \text{ connected}\}$; i.e. a

[5] Each of these three notions will be defined *twice*, once for images and once for text. The context will obviously tell which notion is meant from time to time. Note also that the term "layout" is used elsewhere in the literature in a different sense, namely to denote the rendering of a document on a display device.

region is a connected zone obtained by the union of one or more atomic regions. The fact that we do not require S to be *simply* connected allows some interesting visual objects (e.g. the figure of a goalkeeper partly covered by an approaching ball) to be classified as regions.

The *extended colour function* of an image layout $i = \langle A^i, \pi^i, f^i \rangle$ is defined as the function f_e^i that assigns to each region S a *colour distribution* (i.e. a mapping $f_e^i(S)$ from $\mathcal{C}$ to [0,1] such that $\sum_{\{c \in \mathcal{C}\}} f_e^i(S)(c) = 1$) defined as $f_e^i(S)(c) = \frac{\sum_{T_j \in Z} |T_j|}{|S|}$, where Z is the set containing all and only the atomic regions T_j in $\{T_1, \ldots, T_k\}$ that have colour c (i.e. $f^i(T_j) = c$) and $|S|$ is the cardinality of a region S viewed as a set of points. Intuitively, this function determines the percentage of a region that has a given colour. A region S is not bound to a particular image layout, but is just a "window" that can be opened on many of them. This binding is instead realized in the notion of *grounded region*, which we define as a pair $\langle i, S \rangle$ where $i = \langle A^i, \pi^i, f^i \rangle$ is an image layout and $S \in \pi_e^i$. Finally, we define *the image universe* $\mathcal{IU}$ as the set of all possible image layouts of any domain.

Querying Image Layouts Queries referring to the form dimension of images are called *visual* queries, and can be partitioned as follows:

1. *concrete visual queries:* these consist of full-fledged images that are submitted to the system as a way to indicate a request to retrieve "similar" images;
2. *abstract visual queries:* these are artificially constructed image elements (thus, "abstractions" of image layouts) that address specific aspects of image similarity; they can be further categorised into:
   (a) *colour queries:* specifications of colour distributions, used to indicate a request to retrieve those images that have a similar colour distribution;
   (b) *shape queries:* specifications of one or more shapes (closed simple curves in the 2D space) and possibly of their spatial relationships, used to indicate a request to retrieve those images in which the specified shapes occur as contours of significant objects, in the specified relationships;

   and other categories, such as spatial and texture queries [8], which for reasons of space will not be dealt with here.

Concrete visual queries are processed by "global matching", i.e. by matching a vector of features extracted from the query image, with each of the homologous vectors extracted from the images candidate for retrieval. Abstract visual queries are treated analogously, but at a different level of granularity, i.e. by "local matching": only the visual features indicated in the query (such as shape or colour) are represented in the vectors involved. There are a number of different techniques for performing image matching, each based on a specific set of features and a specific way for combining them in order to obtain a similarity assessment. These techniques are mostly application-dependent, in that their effectiveness

is a function of the type of candidate images and, most importantly, of the goal of retrieval, which greatly affects the relevant similarity criteria. For all these reasons our model does not provide the machinery for defining similarity functions; the choice of which technique to adopt is not important for the rest of the model, so we leave it unspecified and introduce only the SPSs that link it to the rest of the language. Before this, however, we need to introduce two SPSs whose function is to allow queries to be addressed to a *portion* of an image layout, rather than to the image layout as a whole:

- $\texttt{HAIR}(i, r)$ (standing for Has Atomic Image Region): relates the image layout i to one of its grounded atomic regions r;
- $\texttt{HIR}(i, r)$ (Has Image Region): relates the image layout i to one of its grounded regions r.

The intended semantics of HAIR is:

$\texttt{HAIR}^{\mathcal{I}}$: $\mathcal{IU} \times (\mathcal{IU} \times 2^{\mathbb{N}^2}) \rightarrow [0, 1]$, taking an image layout and a grounded atomic image region into $\{0, 1\}$ depending on whether the latter belongs to the former, that is:
$\texttt{HAIR}^{\mathcal{I}}(i, \langle i', S\rangle) = \begin{cases} 1 \text{ if } i = i' \\ 0 \text{ otherwise.} \end{cases}$

The condition for HIR is analogous and will thus not be spelled in detail. We can now discuss the SPSs specifically dealing with visual queries; these are categorised into:

- *SPSs for "global matching"*: in general, there will be a set of such SPSs, each capturing a specific similarity criterion. Since from the conceptual viewpoint these SPSs form a uniform class, we will just discuss one of them, to be understood as a representative of the whole class. Any other symbol of the same sort can be added without altering the structure and philosophy of the language. So, for global matching we use the SPS
  - $\texttt{SI}(i, j)$ (Similar Image): assesses the similarity between two image layouts i and j;
- *SPSs for "local matching"*: these come in two sorts. First we have *selectors*, which are SPSs needed to select the type of feature that needs to be used in matching:
  - $\texttt{HS}(r, s)$ (Has Shape): relates a grounded region r to its shape s;
  - $\texttt{HC}(r, c)$ (Has Colour): relates a grounded region r to its colour c.

  Second, we have true SPSs for local matching, assessing the similarity between individual features of images. Similarly for what we have done for global matching, we include in the language one SPS for each type of feature to be matched; so we have:
  - $\texttt{SC}(c, c')$ (Similar Colour): returns the similarity between two colours c and $\texttt{c}'$;

- $\texttt{SS}(s, s')$ ($\underline{\texttt{S}}\texttt{imilar } \underline{\texttt{S}}\texttt{hape}$): returns the similarity between two shapes $\texttt{s}$ and $\texttt{s}'$.

The semantic clauses for the symbols introduced so far is defined as follows:

$$\texttt{SI}^{\mathcal{I}} : \mathcal{IU} \times \mathcal{IU} \rightarrow [0,1], \text{assigning to each pair of image layouts their degree of similarity.}$$

The semantics of SC and SS is analogous.

$$\texttt{HS}^{\mathcal{I}} : (\mathcal{IU} \times 2^{\mathbb{N}^2}) \times 2^{\mathbb{N}^2} \rightarrow [0,1], \text{assigning to each pair } \langle \text{grounded image region, shape} \rangle \text{ their degree of similarity;}$$

$$\texttt{HC}^{\mathcal{I}} : (\mathcal{IU} \times 2^{\mathbb{N}^2}) \times \mathcal{C} \rightarrow [0,1], \text{assigning to each pair } \langle \text{grounded image region, colour} \rangle \text{ the percentage of the latter in the former, that is, } \texttt{HC}^{\mathcal{I}}(\langle i, S \rangle, c) = f_e^i(S)(c).$$

SPSs for querying image layouts are the first SPSs we encounter, so a word of explanation is in order. The semantic clauses above specify their intended semantics, or desired behaviour. But how do we turn a *desired* behaviour into an *actual* behaviour? From the "practical" point of view, we interpret every occurrence of these SPSs not as the occurrence of an uninterpreted predicate symbol, but as a call to a routine that implements the desired image processing technique. In knowledge representation, this would be called a *procedural attachment*. From the semantic point of view, instead, we apply the so-called *method of concrete domains* [2]: instead of defining retrieval in terms of all the interpretations $\mathcal{I}$ that simply satisfy the set of $\mathcal{ALC}$ assertions representing our document base, we *also* require these interpretations to satisfy the semantic clauses above (see Section 7.1). A fuzzy interpretation $\mathcal{I}$ will thus be called an *image interpretation* if it satisfies the semantic conditions for the SPSs introduced in this section.

3.2 Text Layouts

Modelling Text Layouts Let $\mathbb{N}$ be the set of natural numbers. We define an interval $S \subset \mathbb{N}$ to be *aligned* iff $min\{x \mid x \in S\} = 0$. Given the set of *words* Λ^+ on an alphabet Λ, we define a *text layout* as a triple $t = \langle A^t, \pi^t, f^t \rangle$ where A^t (the *domain*) is a finite aligned interval, π^t is a partition of A^t into non-empty intervals $\{T_1, ..., T_n\}$ called *atomic regions*, and f^t is a total function (the *word function*) assigning a word to each atomic region. The *regions* of a text layout $t = \langle A^t, \pi^t, f^t \rangle$ are defined as the set $\pi_e^t = \{S \mid \exists T_1, ..., T_k \in \pi^t, k \geq 1,\ S = \cup_{i=1}^{k} T_i,\ S \text{ is an interval}\}$; i.e. a region is the interval obtained by the union of one or more pairwise-adjacent atomic regions. Similarly to the case of images, a region S is not bound to a particular text layout, but is just a "window" that can be opened on many of them. This binding is realized in the notion of *grounded text region*, which we define as a pair $\langle t, S \rangle$, where $t = \langle A^t, \pi^t, f^t \rangle$ is a text layout and $S \in \pi_e^t$. Finally, we define *the text universe* $\mathcal{TU}$ as the set of all possible text layouts of any domain.

Querying Text Layouts We distinguish between two categories of queries addressing text layouts:

1. *full-text* queries, requesting texts that share some given syntactic features with a given *text pattern*, which *de facto* identifies a set of texts;
2. *semantic similarity* queries, aimed at retrieving texts which are similar in semantic content to a given text.

In a query of type 1, the text pattern can be specified in many different ways, e.g. by enumeration, via a regular expression, or via *ad hoc* operators specific to text structure such as proximity, positional and inclusion operators (for instance, in the style of the model for text structure presented in [14]). As in the case of images (Section 3.1), the choice as to what sub-language for text patterns to adopt in our model is not important for the rest of the model, so we will leave this piece of the query language unspecified and limit ourselves to specifying how to link it to the main body of the language. To this end, we simply need the SPS `ISyST` (standing for `Is Syntactically Similar To`), whose purpose will be to relate a text layout with the text pattern that it matches. The semantics of the `ISyST` role is the following:

$\texttt{ISyST}^{\mathcal{I}} : \mathcal{TU} \times \mathcal{TU} \rightarrow [0,1]$, assigning 1 to each pair ⟨text layout, text layout⟩ such that the former is equal to the latter, that is:

$$\texttt{ISyST}^{\mathcal{I}}(t, t') = \begin{cases} 1 \text{ if } t = t' \\ 0 \text{ otherwise;} \end{cases}$$

For instance, if we allowed regular expressions in our language for text patterns, the $\mathcal{ALC}$ concept $\exists\texttt{ISyST}.*\sqcup info*$ would denote the text layouts in which at least one word with "info" as a prefix occurs.

Queries of type 2 involve instead semantic similarity matching between text layouts. They are processed on the basis of *automatically constructed* document and query representations, i.e. representations obtained without any human intervention (we will see another semantics-related type of queries, namely *semantic content-based queries*, in Section 4). In this sense, the "semantics" of a text layout is typically a set of terms occurring in the text and which, based on statistical properties, are deemed significant for assessing semantic similarity. Here too, we do not commit to a specific technique for establishing semantic similarity, as there are various plausible candidates for this; instead, our model allows the use, for processing this particular kind of queries, of any available semantic similarity engine. To this end, the language provides a class of SPSs, each modelling semantic similarity according to the specific engine which the SPS invokes. Here we just discuss a generic representative of this class of SPSs, i.e. the $\mathcal{ALC}$ role $\texttt{ISeST}(t, t')$ (standing for `Is Semantically Similar To`) which, given two text layouts as input, returns their degree of similarity. Formally,

$\texttt{ISeST}^{\mathcal{I}} : \mathcal{TU} \times \mathcal{TU} \rightarrow [0,1]$, such that $\texttt{ISeST}(t_1, t_2)$ gives the degree of similarity of text layout t_1 to text layout t_2

Finally, similarly to what we did for images, we introduce two SPSs whose function is to allow queries (of various kinds, included queries of type 1 and 2 above) to be addressed to a portion of a text layout, rather than to the text layout as a whole. Not surprisingly, the SPSs are:

- $\texttt{HATR}(t, r)$ ($\underline{\texttt{H}}\texttt{as}$ $\underline{\texttt{A}}\texttt{tomic}$ $\underline{\texttt{T}}\texttt{ext}$ $\underline{\texttt{R}}\texttt{egion}$): relates the text layout t to one of its grounded atomic regions r;
- $\texttt{HTR}(t, r)$ ($\underline{\texttt{H}}\texttt{as}$ $\underline{\texttt{T}}\texttt{ext}$ $\underline{\texttt{R}}\texttt{egion}$): relates the text layout t to one of its grounded regions r.

whose semantic conditions parallel those for $\texttt{HAIR}(i, r)$ and $\texttt{HIR}(i, r)$ and will thus not be spelled out. A fuzzy interpretation $\mathcal{I}$ will be called a *text interpretation* if it satisfies the semantic conditions for the SPSs introduced in this section.

4 Modelling and Querying Content

4.1 Modelling Content

We take the content of a document (be it a text, or an image, or any combination of the two) to be a *situation*, i.e. the set of all states of affairs "compatible" with the information contained in the document. For instance, the content of an image will be the set of states of affairs that verify the facts depicted in the image, irrespective e.g. of when the action takes place, of what the people represented therein are thinking of, and of other facts taking place outside the setting of the image. Let l be a layout (either text or image) uniquely identified (in a way to be made precise later) by the individual constant $\texttt{l}$. A *content description* δ for l is a set of fuzzy assertions, consisting of the union of four component subsets:

1. the *layout identification*, a set containing only a single fuzzy assertion of the form $\langle \texttt{Ego}(\texttt{l}), 1 \rangle$, whose role is to associate, along with the layout naming functions ν_I and ν_T (see Section 7.1), a content description with the layout it refers to. In what follows $\sigma(\texttt{l})$ will denote the set of the (possibly many) content descriptions whose identification is $\langle \texttt{Ego}(\texttt{l}), 1 \rangle$;
2. the *object anchoring*, a set of fuzzy assertions of the form $\langle \texttt{Rep}(\texttt{r}, \texttt{o}), n \rangle$, where $\texttt{r}$ is an individual constant that uniquely identifies a grounded region of l and $\texttt{o}$ is an individual constant that identifies the object represented in the region[6];
3. the *situation anchoring*, a set of fuzzy assertions of the form $\langle \texttt{About}(\texttt{l}, \texttt{o}), n \rangle$, where $\texttt{l}$ and $\texttt{o}$ are as above. By using these assertions, it can be stated what the situation described by the layout is "globally" about;

[6] The combined effect of components 1 and 2 could have been achieved by eliminating the $\texttt{Ego}$ predicate symbol and making the $\texttt{Rep}$ predicate symbol ternary, i.e. $\texttt{Rep(l,r,o)}$. While extended DLs capable of dealing with predicate symbols of arity ≥ 2 do exist, we prefer to use Ockham's razor and stick to the simpler, orthodox DLs of which $\mathcal{ALC}$ is the standard representative.

4. the *situation description*, a set of fuzzy simple assertions (where neither the predicates `Ego`, `Rep` and `About` occur), describing important facts stated in the layout about the individual constants identified by assertions of the previous two kinds.

While the task of components 1 to 3 is actually that of binding the form and content dimensions of the same layout, component 4 pertains to the content dimension only.

As an example, let us consider a photograph showing a singer, Kiri, performing as Zerlina in Mozart's "Don Giovanni". Part of a plausible content description for this image, named `i`, could be (for simplicity, in this example we only use crisp assertions):

```
{Ego(i), About(i,DonGiovanni), Rep(r,Kiri), Plays(Kiri,Zerlina)}
```

Note that there may be more than one content description for the same layout l; this reflects the fact that an image may be considered under multiple viewpoints. In Section 7.1 we will see that, as a result of this, the "degrees of relevance" of a layout to a query resulting from different content descriptions do not add up. Any of components 2 to 4 can be missing in a content description.

4.2 Querying Content

Queries pertaining to content are called *content-based* queries, and involve conditions on the semantics of a text or image. In the case of text we have discussed a type of queries that also lay a claim of being grounded on semantics, namely "semantic similarity queries". In the case of images, some types of concrete visual queries (depending on the underlying technique being used) also lay a similar claim. The main difference between these two types of queries and content-based queries is that, while the former are processed on the basis of *automatically constructed* document and query representations, the representations used in content-based queries reflect a (human) conceptualisation, even though this may have been derived with the aid of an automatic support. We can therefore see these two categories of queries as addressing two different notions of content: content "as understood by program" versus content "as understood by mind".

Note that there are no SPSs specific to content-based queries, as there exists no underlying content-assessing engine that we want to hook our representations to! Reasoning about content will be performed "directly" (i.e. without procedural attachments) by fuzzy $\mathcal{ALC}$, using component 4 of content descriptions as input. The results of this logical reasoning activity will be transparently merged to the results of non-logical computations (obtained through the procedural attachments to the various SPSs) by fuzzy $\mathcal{ALC}$ using components 1 to 3 of content descriptions.

5 Modelling and Querying Document Structure

As mentioned in the introduction, we take multimedia documents to be complex documents consisting in general of a hierarchically structured set of "atomic" sub-documents, which may in turn be either chunks of text or images. It is just natural, then, to allow our model to deal not only with the features of these sub-documents, but also with the way these are structured into a complex document. We hence define the notion of *document* and characterize a set of SPSs for addressing its features within queries.

5.1 Modelling Structure

A *document* is a pair $d = \langle w_n, R \rangle$, where

1. w_n is a pair $\langle n, w \rangle$ where $n \in \mathbb{N}$ is the *order* of the layout and $w : [1, n] \rightarrow (\mathcal{IU} \cup \mathcal{TU})$;
2. $R = \{\rho_1, \ldots, \rho_m\}$ is a set of intervals such that 1) $\rho_i \subseteq [1, n]$ for all $1 \leq i \leq m$; 2) $[1, n] \in R$; and 3) for all $\rho_i, \rho_j \in R$, $\rho_i \subseteq \rho_j$ or $\rho_j \subseteq \rho_i$ or $\rho_i \cap \rho_j = \emptyset$.

A *grounded region* of a document $d = \langle w_n, R \rangle$ is defined as a pair $\langle d, \rho \rangle$ such that $\rho \in R$; its *extent* is defined as the set of image or text layouts to which elements in ρ are mapped by w_n. The *structure* of a document $d = \langle w_n, R \rangle$ is defined by the the tree $S_d = \langle R, E \rangle$ (where R are the nodes of the tree and $E \subset R^2$ are its edges) such that $(\rho_1, \rho_2) \in E$ iff $\rho_2 \subset \rho_1$ and there is no $\rho_3 \in R$ such that $\rho_2 \subset \rho_3 \subset \rho_1$. It can be easily verified that S_d is a tree with root $[1, n]$. By E^+ we indicate the transitive closure of E. We let $\mathcal{D}$ be the set of all documents and $\mathcal{R}$ be the set of all intervals $[m_1, m_2]$, with $m_1 \leq m_2$ and $m_1, m_2 \in \mathbb{N}$.

5.2 Querying Structure

In querying documents, a user would like to be able to perform the following kinds of operations:

- navigate along the structure of documents; SPSs for expressing this navigation will be called *structural*;
- access the basic constituents of a grounded region, i.e. the image and text layouts that are in the extent of that region; SPSs for expressing these accesses will be termed *extensional*;
- query these image and text layouts. These queries (called *ground queries*) are to be expressed by means of the SPSs introduced in Sections 3 and 4.

Structural symbols, in turn, can be categorised as follows:

- *generic* SPSs, allowing one to access any grounded region of a document, regardless of the region's type or position; for this we just need the SPS `HN`

(standing for Has Node), relating a document to one of its grounded regions. Its semantics is given by:

$\texttt{HN}^{\mathcal{I}}$: $\mathcal{D} \times (\mathcal{D} \times \mathcal{R}) \to [0,1]$, assigning 1 to the pairs ⟨document, grounded region⟩ such that the latter is a grounded region of the former; i.e.:

$$\texttt{HN}^{\mathcal{I}}(d, \langle d', \rho \rangle) = \begin{cases} 1 \text{ if } d = d' \\ 0 \text{ otherwise.} \end{cases}$$

– *positional* SPSs, allowing one to navigate in the structure of the document. Among the many primitives that might be envisaged in order to model tree navigation, we adopt the following SPSs:

- `Root`, a primitive concept denoting roots of documents;
- `Leaf`, the concept denoting leaf nodes (i.e. image or text layouts) of documents;
- `HasChild`, a role denoting the link between nodes and their children nodes;
- `HasParent`, a role denoting the link between nodes and their parent node;
- `HasDes`, the transitive closure of `HasChild`;
- `HasAncestor`, the transitive closure of `HasParent`.

We just show the semantics of two positional symbols, leaving that of the others for the reader to work out.

$\texttt{Leaf}^{\mathcal{I}}$: $\mathcal{D} \times \mathcal{R} \to [0,1]$, assigning 1 solely to "leaf" grounded regions:

$$\texttt{Leaf}^{\mathcal{I}}(\langle d, \rho \rangle) = \begin{cases} 1 \text{ if } \rho \text{ is a leaf node in } d\text{'s structure } S_d \\ 0 \text{ otherwise;} \end{cases}$$

$\texttt{HasDes}^{\mathcal{I}}$: $(\mathcal{D} \times \mathcal{R}) \times (\mathcal{D} \times \mathcal{R}) \to [0,1]$, assigning 1 to the pairs of grounded regions such that the latter is in the offspring of the former:

$$\texttt{HasDes}^{\mathcal{I}}(\langle d, \rho \rangle, \langle d', \rho' \rangle) = \begin{cases} 1 \text{ if } d = d' \text{ and } \langle \rho, \rho' \rangle \in E^+ \\ \quad \text{where } S_d = \langle R, E \rangle \\ 0 \text{ otherwise.} \end{cases}$$

As for extensional SPSs, we include two of them in the language, relating a grounded region of a document to the image and text layouts it contains. The SPSs are `HasImage` and `HasText`; the semantics of the former is:

$\texttt{HasImage}^{\mathcal{I}}$: $(\mathcal{D} \times \mathcal{R}) \times \mathcal{IU} \to [0,1]$, such that, given $d = \langle w_n, R \rangle$

$$\texttt{HasImage}^{\mathcal{I}}((d, \rho), i) = \begin{cases} 1 \text{ if } \rho \in R \text{ and } w(k) = i \text{ for some } k \in \rho \\ 0 \text{ otherwise.} \end{cases}$$

while that of the latter is perfectly analogous. A fuzzy interpretation $\mathcal{I}$ will be called a *document interpretation* if it satisfies the semantic conditions for the SPSs introduced in this section.

6 A Unified Query Language

We are now in the position of defining the *query language* of the model. This language satisfies the two basic requirements necessary for complying with the philosophy of our model, namely: 1) it is a concept language of a DL, so that matching queries against documents can be done in the logical framework defined so far; and 2) it complies with the semantics of the symbols for addressing form, content and structure introduced in the previous sections.

In order to closely reflect a query specification process, the language will be presented in a top-down fashion, starting from concepts addressing documents and their structure, and proceeding down to the queries addressing the basic components of documents, i.e. text and images.

6.1 Document Queries

The following grammar defines the *document query language.*

$$
\begin{aligned}
\langle \textit{document-query} \rangle &::= \langle \textit{document-concept} \rangle \mid \\
&\quad \langle \textit{document-query} \rangle \sqcap \langle \textit{document-query} \rangle \mid \\
&\quad \langle \textit{document-query} \rangle \sqcup \langle \textit{document-query} \rangle \\
\langle \textit{document-concept} \rangle &::= \exists \texttt{HN}.\langle \textit{node-concept} \rangle \\
\langle \textit{node-concept} \rangle &::= \langle \textit{extent-concept} \rangle \mid \\
&\quad \langle \textit{structure-concept} \rangle \mid \exists \langle \textit{structure-role} \rangle.\langle \textit{node-concept} \rangle \mid \\
&\quad \langle \textit{node-concept} \rangle \sqcap \langle \textit{node-concept} \rangle \mid \\
&\quad \langle \textit{node-concept} \rangle \sqcup \langle \textit{node-concept} \rangle \\
\langle \textit{extent-concept} \rangle &::= \exists \texttt{HasImage}.\langle \textit{image-query} \rangle \mid \exists \texttt{HasText}.\langle \textit{text-query} \rangle \\
\langle \textit{structure-concept} \rangle &::= \texttt{Root} \mid \texttt{Leaf} \\
\langle \textit{structure-role} \rangle &::= \texttt{HasChild} \mid \texttt{HasParent} \mid \texttt{HasDes} \mid \texttt{HasAncestor}
\end{aligned}
$$

A document query is a combination, via the conjunction and disjunction constructors[7], of *document-concepts*, each of which references, via the role `HN`, any node of the document structure, on which a condition is then stated by a *node-concept*. In its simplest form, a *node-concept* is a condition on the basic constituents of the document (*extent-concept*), stating what kind of components is being addressed (i.e. a text layout or an image layout, indicated respectively by the `HasText` and `HasImage` $\mathcal{ALC}$ roles) and followed by a query of the appropriate kind. Otherwise, a *node-concept* may contain any $\sqcap$-/$\sqcup$-combination of navigational conditions, which are couched in terms of structural symbols. Each such navigational conditions may or may not involve an *extent-concept*.

[7] The reason why we do not allow to use the negation constructor here is analogous to the one that, in the relational calculus for DBs, justifies the restriction to *safe queries*. See e.g. [1, page 97].

6.2 Image Queries

The syntax of image queries is given by the following BNF rules:

$$
\begin{aligned}
\langle \textit{image-query} \rangle &::= \langle \textit{image-concept} \rangle \mid \langle \textit{image-query} \rangle \sqcap \langle \textit{image-query} \rangle \mid \\
&\quad \langle \textit{image-query} \rangle \sqcup \langle \textit{image-query} \rangle \\
\langle \textit{image-concept} \rangle &::= \exists \texttt{SI}.\{\langle \textit{layout-name} \rangle\} \mid \\
&\quad \exists \texttt{About}.\langle \textit{content-concept} \rangle \mid \\
&\quad \exists \texttt{HAIR}.\langle \textit{region-concept} \rangle \mid \\
&\quad \exists \texttt{HIR}.\langle \textit{bound-region-concept} \rangle \\
\langle \textit{region-concept} \rangle &::= \exists \texttt{HC}.\langle \textit{colour-concept} \rangle \mid \exists \texttt{HS}.\langle \textit{shape-concept} \rangle \mid \\
&\quad \exists \texttt{Rep}.\langle \textit{content-concept} \rangle \mid \\
&\quad \langle \textit{region-concept} \rangle \sqcap \langle \textit{region-concept} \rangle \mid \\
&\quad \langle \textit{region-concept} \rangle \sqcup \langle \textit{region-concept} \rangle \\
\langle \textit{bound-region-concept} \rangle &::= \exists \texttt{Rep}.\langle \textit{content-concept} \rangle \mid \\
&\quad \langle \textit{bound-region-concept} \rangle \sqcap \langle \textit{region-concept} \rangle \mid \\
&\quad \langle \textit{bound-region-concept} \rangle \sqcup \langle \textit{region-concept} \rangle \\
\langle \textit{colour-concept} \rangle &::= \{\langle \textit{colour-name} \rangle\} \mid \exists \texttt{SC}.\{\langle \textit{colour-name} \rangle\} \\
\langle \textit{shape-name} \rangle &::= \{\langle \textit{shape-name} \rangle\} \mid \exists \texttt{SS}.\{\langle \textit{shape-name} \rangle\}
\end{aligned}
$$

Note that a *layout-name*, a *colour-name* and a *shape-name* are not concepts, but individual constants. Image queries are thus *not* concepts of $\mathcal{ALC}$, but of the DL $\mathcal{ALCO}$, extending $\mathcal{ALC}$ with the "singleton" constructor "{}", which given an individual constant `i` returns a concept `{i}`. The singleton constructor is necessary in queries because it allows the reference to specific individual constants. This added expressive power has no impact on the complexity of the image retrieval problem, as we will discuss in the full paper.

An image query is a combination, via the conjunction and disjunction constructors, of so-called *image-concepts*, each of which may have one of four forms (following the order of the syntax):

1. a global similarity match request ("concrete visual query");
2. a query on some content-related object described by *content-concept*, which is any $\mathcal{ALCO}$ concept built with the symbols used for situation descriptions;
3. a query on an atomic region, which is required to satisfy the property expressed by the embedded *region-concept*;
4. a query on a region. In this case, the embedded concept is the same as a *region-concept*, but it must include a `Rep` clause; this prevents the specification of queries involving arbitrary regions, of which there are an exponential number.

A *region-concept* gives conditions on a region, and is built as an $\sqcap/\sqcup$-combination of three basic conditions: one concerns the colour of the region, which must be the same as, or similar to, a specified colour (*colour-concept*); another analogously

concerns the shape of a region (*shape-concept*); the third involves the real-world entity represented by a region, and is a *content-concept*.

Let us reconsider the example introduced in Section 4. The images about the Don Giovanni are retrieved by the query ∃About.{DonGiovanni}. Those showing the singer Kiri are described by ∃HIR.∃Rep.{kiri}. Turning to visual queries, the request to retrieve the images similar to a given one, named this, is expressed by ∃SI.{this}, and can be easily combined with any conceptual query, e.g. yielding ∃SI.{this} ⊔ ∃About.{DonGiovanni}, which would retrieve the images that are either similar to the given one or are about Don Giovanni. As for abstract visual queries, the images in which there is a blue region whose contour has a shape similar to a given curve s are retrieved by ∃HAIR.(∃HC.{blue} ⊓ ∃HS.∃SS.{s}). Finally, the user interested in retrieving the images in which Kiri plays Zerlina and wears a blue-ish dress, can use the query ∃HIR.∃Rep.({Kiri}⊓∃Plays.{Zerlina})⊓(∃HC.∃SC.{blue}).

6.3 Text Queries

The syntax of text queries is given by the following BNF rules:

$$\begin{aligned}
\langle \textit{text-query} \rangle ::= {} & \langle \textit{text-concept} \rangle \mid \langle \textit{text-query} \rangle \sqcap \langle \textit{text-query} \rangle \mid \\
& \langle \textit{text-query} \rangle \sqcup \langle \textit{text-query} \rangle \\
\langle \textit{text-concept} \rangle ::= {} & \exists \texttt{ISyST}.\langle \textit{text-pattern} \rangle \mid \exists \texttt{ISeST}.\{\langle \textit{text-layout-name} \rangle\} \mid \\
& \exists \texttt{HTR}.\exists \texttt{Rep}.\langle \textit{content-concept} \rangle \mid \exists \texttt{HATR}.\exists \texttt{Rep}.\langle \textit{content-concept} \rangle \mid \\
& \exists \texttt{About}.\langle \textit{content-concept} \rangle
\end{aligned}$$

A text query is a combination, via the conjunction and disjunction constructors, of so-called *text-concepts*, each of which may have one of four forms (following the order of the syntax):

1. an exact match query, in which *text-pattern* can be a single text fragment (full-text search), a regular expression or a complex structural query;
2. a similarity match request;
3. a query on a content object related to a segment of text via some Rep assertion;
4. a query on some content-related object related to a whole text via an About assertion.

7 Retrieval

The behaviour of our query language is specified by formally defining the notion of a document base and of document retrieval. We model a document base as a collection including an image base, a text base, and additional knowledge concerning how the individual images and texts belonging to them are structured into more complex aggregates.

7.1 Document Bases and Document Retrieval

An image layout base is a 4-tuple $ILB = \langle IL, \nu_I, \Sigma_{IC}, \Sigma_{ID} \rangle$ where IL is a set of image layouts, ν_I is a *naming function* associating $\mathcal{ALC}$ individual constants to the image layouts and grounded image regions in IL, Σ_{IC} is the set of content descriptions associated with the layouts in IL, and Σ_{ID} is the domain knowledge bases for the images in ILB. A text layout base is a 4-tuple $TLB = \langle TL, \nu_T, \Sigma_{TC}, \Sigma_{TD} \rangle$, defined in a completely analogous way. A document base DLB is a similarly defined pair $DLB = \langle DL, \nu_D \rangle$. A *document base* is therefore a triple $DB = \langle ILB, TLB, DLB \rangle$.

In response to a query C addressed to a document base DB, each document $\nu_D(\mathtt{d}) = d = \langle w_n, R \rangle$ is attributed a degree of relevance m determined in the following way. Let O be the set of the n image and text layouts that are leaves of d and are uniquely identified by the $\mathcal{ALC}$ individual constants $a_1, \ldots, a_n$. Let Δ be the Cartesian product $\sigma(a_1) \times \ldots \times \sigma(a_n)$, where σ is as defined in Section 4. For each tuple $\tau_i = \langle \delta_{i_1}, \ldots, \delta_{i_n} \rangle \in \Delta$ calculate

$$n_i = \overline{Maxdeg}((\Sigma_{TD} \cup \Sigma_{ID} \cup \bigcup_{1 \leq j \leq n} \delta_{i_j}), Q(\mathtt{d}))$$

where $\overline{Maxdeg}$ is the same as the $Maxdeg$ function discussed in Section 2 except for the fact that it is not calculated with respect to *all* fuzzy interpretations $\mathcal{I}$, but with respect to only those fuzzy interpretations that are image, text, content and document interpretations (as defined in Sections 3.1, 3.2, 4 and 5, respectively). The value n_i can be interpreted as the degree of relevance of d were it calculated on a specific choice of content representations of the images and texts in O. The degree of relevance of d is then simply obtained by taking the maximum over all such choices, i.e. $m = max_{\tau_i \in \Delta}\{n_i\}$.

As an example, let us consider a document base DB in which the TLB and ILB components are empty and ILB is such that its IL component contains two image layouts named $\mathtt{i}$ and $\mathtt{j}$, its Σ_{IC} component consists of the two content descriptions $\{\langle \mathtt{Ego(i)}, 1\rangle, \langle \mathtt{About(i,o)}, 0.8\rangle, \langle \mathtt{DonGiovanni(o)}, 1\rangle\}$ and $\{\langle \mathtt{Ego(j)}, 1\rangle, \langle \mathtt{About(j,o)}, 0.7\rangle, \langle \mathtt{WestSideStory(o)}, 1\rangle\}$, and its Σ_{ID} component consists of the axioms:

$$\langle \mathtt{DonGiovanni} \sqsubseteq \mathtt{EuropeanOp}, 1\rangle, \langle \mathtt{WestSideStory} \sqsubseteq \mathtt{AmericanOp}, 1\rangle,$$
$$\langle \mathtt{EuropeanOp} \sqsubseteq \mathtt{Op} \sqcap (\exists \mathtt{CondBy.European}), 0.9\rangle, \text{ and}$$
$$\langle \mathtt{AmericanOp} \sqsubseteq \mathtt{Op} \sqcap (\exists \mathtt{CondBy.European}), 0.8\rangle.$$

Suppose we are interested in documents containing images about operas conducted by a European director. To this end, we can use the query:

$$\exists \mathtt{HN}.\exists \mathtt{HasImage}.\exists \mathtt{About}.(\mathtt{Op} \sqcap \exists \mathtt{CondBy.European}).$$

It can be verified that the degree of relevance attributed to $\mathtt{i}$ is 0.8, whereas that of $\mathtt{j}$ is 0.7.

7.2 Implementation of the Model

We close with some implementation considerations. In order to effectively perform document retrieval as prescribed by the model defined so far, we envisage a MRS consisting of the following modules:

1. a *matching engine* for each of the SPSs representing similarity (`SI`, `SC` and `SS` for images; `ISyST` and `ISeST` for text). To this end, each such engine will make use of the feature vectors for the layouts in the document base, stored in a *matching database*;
2. a *fuzzy $\mathcal{ALC}$ theorem prover*, which will handle the semantic information processing, collecting the assertions contained in the Σ_{IC}, Σ_{ID}, Σ_{TC} and Σ_{TD} components of the document base and using them in reasoning about document content;
3. a *query processor*, responsible of decomposing each query into abstract, concrete, and conceptual sub-queries, assigning the evaluation of each sub-query to the appropriate component, and then properly combining the results in order to obtain the final ranked list of images. For its operation, the query processor will use a database, called the *document structure database*, containing a specification of the semantics of selectors as well as naming functions.

The details of these components are outside the scope of this paper. We only remark, at this point, that they are well within reach of the current technology. In particular, we have developed a theorem prover for a significant extension of the DL we use here based on a sound and complete tableaux calculus; this theorem prover is currently being prototyped for subsequent experimental evaluation.

8 Conclusions

We have presented a model for structured documents with images and texts. The model makes two important contributions. First, at single-medium level, it makes full and proper use of semantics and knowledge in dealing with the retrieval of images and text, while offering, at the same time, the similarity-based kind of retrieval that is undoubtedly the most significant contribution of the research carried out in these two areas during the last decade. More importantly, all these forms of retrieval coexist in a well-founded framework, which combines in a neat way the different techniques, notably digital signal processing and semantic information processing, required to deal with the various aspects of the model. Secondly, at the multimedia level, the model addresses the retrieval of structural aggregates of images and texts, casting the single medium models in a framework informed by the same, few principles. At present, to the best of our knowledge, no other model offering the same functionalities as the one presented here, exists.

The breadth in scope of the model and the space limitations of the paper have determined a concise treatment, mainly devoted to outline the model's

basic traits. No doubt many aspects have been treated in a sketchy way, others have been neglected *tout court*, and equally valid alternatives to the proposed solutions have not been discussed. We refer the interested reader to the full paper for more detailed solutions and discussion.

Since the representations handled by the model have a clean semantics, further extensions to the model are possible. For instance, image retrieval by spatial similarity can be added: at the form level, effective spatial similarity algorithms (e.g. [8]) can be embedded in the model via procedural attachment, while significant spatial relationships can be included in content descriptions by drawing from the many formalisms developed within the qualitative spatial reasoning research community [6]. Analogously, the model can be enhanced with the treatment of texture-based similarity image retrieval.

We believe that the presented model can open the way to a novel approach to the modelling of multimedia information, leading to the development of retrieval systems able to cope in a formally neat and practically adequate way with documents including text, graphics and images. More research is needed to attack delay-sensitive media, such as audio and video, but we think that the present model constitutes a good starting point.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of databases*. Addison Wesley, Reading, MA, 1995.
2. F. Baader and P. Hanschke. A schema for integrating concrete domains into concept languages. In *Proceedings of IJCAI-91, International Joint Conference on Artificial Intelligence*, pages 452–457, Sydney, 1991.
3. J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu. The Virage image search engine: an open framework for image management. In *Storage and Retrieval for Still Image and Video Databases IV*, volume 2670 of *SPIE Proceedings*, pages 76–87, San Jose, CA, February 1996.
4. A. Borgida. Description logics in data management. *IEEE Transactions on Data and Knowledge Engineering*, 7:671–682, 1995.
5. J. Chen and S. Kundu. A sound and complete fuzzy logic system using Zadeh's implication operator. In Z. W. Ras and M. Michalewicz, editors, *Proceedings of ISMIS-96, 9th International Symposium on Methodologies for Intelligent Systems*, pages 233–242, Zakopane, PL, 1996.
6. A. G. Cohn. Calculi for qualitative spatial reasoning. In *Proceedings of AISMC-3*, Lecture Notes in Computer Science. Springer Verlag, 1996.
7. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, and W. Niblack. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
8. V. N. Gudivada and V. V. Raghavan. Design and evaluation of algorithms for image retrieval by spatial similarity. *ACM Transactions on Information Systems*, 13(2):115–144, 1995.
9. V. N. Gudivada and V. V. Raghavan, editors. *IEEE Computer. Special Issue on Content-Based Image Retrieval*. IEEE, September 1995.

10. E. J. Guglielmo and N. C. Rowe. Natural-language retrieval of images based on descriptive captions. *ACM Transaction on Information Systems*, 14(3):237–267, 1996.
11. C. Meghini. An image retrieval model based on classical logic. In *Proceedings of SIGIR-95*, pages 300–308, Seattle, WA, 1995.
12. C. Meghini, F. Sebastiani, U. Straccia, and C. Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of SIGIR-93*, pages 298–307, Pittsburgh, PA, July 1993.
13. C. Meghini and U. Straccia. A relevance terminological logic for information retrieval. In *Proceedings of SIGIR-96*, pages 197–205, Zürich, CH, August 1996.
14. G. Navarro and R. Baeza-Yates. A language for queries on structure and contents of textual databases. In *Proceedings of SIGIR-95*, pages 93–101, Seattle, WA, Jul 1995.
15. A. Rosenfeld and A. C. Kak. *Digital picture processing.* Academic Press, New York, 2nd edition, 1982.
16. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.
17. A. F. Smeaton and I. Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of SIGIR96*, pages 174–180, Zurich, CH, August 1996.