How to retrieve multimedia documents described by MPEG-7

Nastaran Fatemi[1], Mounia Lalmas[2] and Thomas Rölleke[2]

[1] University of Applied Sciences of Western Switzerland at Yverdon, Switzerland
Nastaran.Fatemi@eivd.ch

[2] Queen Mary University of London, United Kingdom
{mounia, thor}@dcs.qmul.ac.uk

Abstract. The "Semantic Web" aims at enhancing the functionality of the current web to bring "meaning" to the content of web pages so that to considerably improve the access to this content. Bringing meaning to multimedia content is the aim of the MPEG-7 standard. However, merely attaching MPEG-7 descriptions to multimedia content does not necessarily make access to this type of content more effective. For MPEG-7 to considerably improve the access to multimedia content, we first require means to make the multimedia content searchable according to specific application contexts and user needs. This paper describes the development and implementation of the Semantic Views Query Language, which provides an abstract model reflecting the user retrieval needs and behaviours. A second requirement is to consider features specific to MPEG-7 descriptions, i.e. a mixture of content and factual knowledge, and structure, in providing a relevance-based ranking of multimedia material according to user information needs. This paper describes the development and implementation of a Retrieval Model for MPEG-7 annotated multimedia content, which encompasses these features in a uniform manner.

1. Introduction

The impact of multimedia data in our information-driven society is growing since tools for creating, manipulating and exchanging multimedia data are becoming widely available. While the first generation of multimedia processing concentrated mainly on "re-playing" the data and users consumed the information directly, the second generation of multimedia tools supports the increasingly digital creation, manipulation and exchange of multimedia data. In the first generation, multimedia data was mainly gained from translating analogous data sources into digital data sources. In the second generation, we find real-time recording of digital data.

Such multimedia data sources need to be effectively and efficiently searched for information of interest to users or filtered to receive only information satisfying user preferences. This may be the case for scenarios such as the recording and use of broadcast programmes, multimedia teaching material in educational and training institutes, or general multimedia data in security agencies, national archival centres and libraries, journalism, tourism and medical applications. This is even more the case on the world-wide-web (web), which has witnessed and is still witnessing a vast and overwhelming increase in the amount of multimedia content.

Some years ago, Tim Berners-Lee introduced the term "Semantic Web" foreseeing the creation of a web that can only be managed well when applying "intelligent" computer programs, such as search engines and agents, as it will become impossible for humans to process the gigantic amount of information available. One central idea of the Semantic Web is the idea of seamless operation for the users and screening them from all the underlying matching and inferring processes. For search engines, this means more effective information retrieval, and for agents, better opportunities to provide meaningful services. While much progress has been made for text-based content, dealing with multimedia content is still in its infancy. One reason for this is that processing multimedia content is fundamentally different from processing text.

Let us consider a video presentation showing the various bridges in London. How to attach meaning to it, so that to allow for its seamless access for users? The idea would be to attach a description or "metadata" stating, for example, that the location is London, that the sites of interest are bridges, and possibly that the perspective is tourism. In this way, searching for tourist information about bridges in London would lead to the video presentation being returned to users.

The increasingly diverse role that multimedia sources are destined to play in our society and the growing need to have these sources accessed made it necessary to develop forms of multimedia information representation that go beyond the simple waveform or sample-based, frame-based (e.g. MPEG-1 and MPEG-2) or object-based (e.g. MPEG-4) representations. **MPEG-7**, formally called "*Multimedia Content Description Interface*", is a new standard for describing the content of multimedia data [ISO,MPE99,MPE00,MPE01,MPEG-7]. MPEG-7 is a means of attaching metadata to multimedia content. MPEG-7 specifies a standard set of description tools, which can be used to describe various types of multimedia information. These tools shall be associated with the content itself to allow efficient and effective searching of multimedia material of users' interests.

Applying such standard description tool can considerably improve the access to multimedia content; however, such representation is a necessary but not a sufficient condition to make this type of content searchable according to specific application contexts and user needs. Indeed, MPEG-7 provides a generic library of descriptions to cover almost all application domains. Nevertheless it is not designed to take into account any specific user model required for a given multimedia retrieval process. To retrieve multimedia content using its associated MPEG-7 descriptions, we need an abstract model that reflects the user retrieval needs and behaviours. A query language adapted to retrieve MPEG-7 should from one side reflect such high-level abstract model and on the other side be adequate for the effective retrieval of MPEG-7 content descriptions. In order to fulfil these two requirements we propose the *Semantic Views Query Language* (SVQL), which provides an abstract model that takes into account various users requirements and viewpoints in the process of multimedia retrieval. The development and implementation of SVQL is described in Section 4.

A second requirement to improve the access to multimedia content is to provide a relevance-based ranking of multimedia material according to user information needs. This is because the information retrieval process is an uncertain one, as it is based on estimated representation of document content and query formulation. In addition, it is often the case, and in fact mostly the case on the web, that there exists many if not too many relevant items, so the most relevant ones should be returned first. The relevance-based ranking of multimedia material should take into account characteristics that are specific to MPEG-7 descriptions, namely that they can be viewed as a mixture of content and factual knowledge; and in particular, that they display a structure: that is they are composed of elements describing parts of the multimedia content as well as multimedia content. By exploiting the structural characteristic of MPEG-7 descriptions, parts of as well as the entire multimedia content can be searched, thus allowing users to precisely access the data of interest to them. A *Retrieval Model* for MPEG-7 annotated multimedia content, which encompasses these characteristics and others in a uniform manner, is described in Section 5.

This paper is organised as follows. In Section 2, we provide background information regarding MPEG-7, including scope and definitions. In Section 3, we discuss related work. Section 4 and Section 5 present in details the development and implementation of the Semantic View Query Language (SVQL) and a Retrieval Model for MPEG-7, respectively. We conclude and discuss future work in Section 6.

2. MPEG-7

MPEG-7 has been developed by the Moving Pictures Expert Group (MPEG), a working group of ISO/IEC [ISO]. The goal of the MPEG-7 standard is to provide a rich set of standardised tools to describe multimedia and in particular audio-visual content. Unlike the preceding MPEG standards (MPEG-1, MPEG-2, MPEG-4), which have mainly addressed coded representation of multimedia content, MPEG-7 focuses on representing information about the content at different levels. The structural level (e.g. "this video consists of a sequence of segments and each segment is composed of several shots") is supported in the same way as the (visual) feature level (e.g. "this object has the form of a flower") or the semantic level (e.g. "the baby ate the biscuit"). The content itself is out of the scope of the standard and MPEG-7 states explicitly that the description tools are applicable for all kinds of multimedia content independent of its format and coding. The methods and technologies generating and using the descriptions are not part of the standard and the tools are not restricted to a specific set or class of applications. To reach this goal MPEG-7 restricts itself to few, but powerful concepts. These are:

- A set of *descriptors* (Ds), for representing features of audio-visual material, (e.g. colour histogram).
- A set of *description schemes* (DSs), which define the structure and the semantics of the relationships between elements, which include Ds and DSs. An example is the hierarchical structure of a video.
- A *description definition language* (DDL) to specify Ds and DSs.
- System tools to support efficient binary encoding multiplexing, synchronisation and transmission of the descriptions.

The DDL allows the representation of complex hierarchies as well as the definition of flexible relationships between elements [ISOb]. The DSs and Ds are platform-independent and must be validated. An existing language that fulfils most of these requirements is XML Schema [XML], which is used by MPEG-7 as the basis for its DDL.

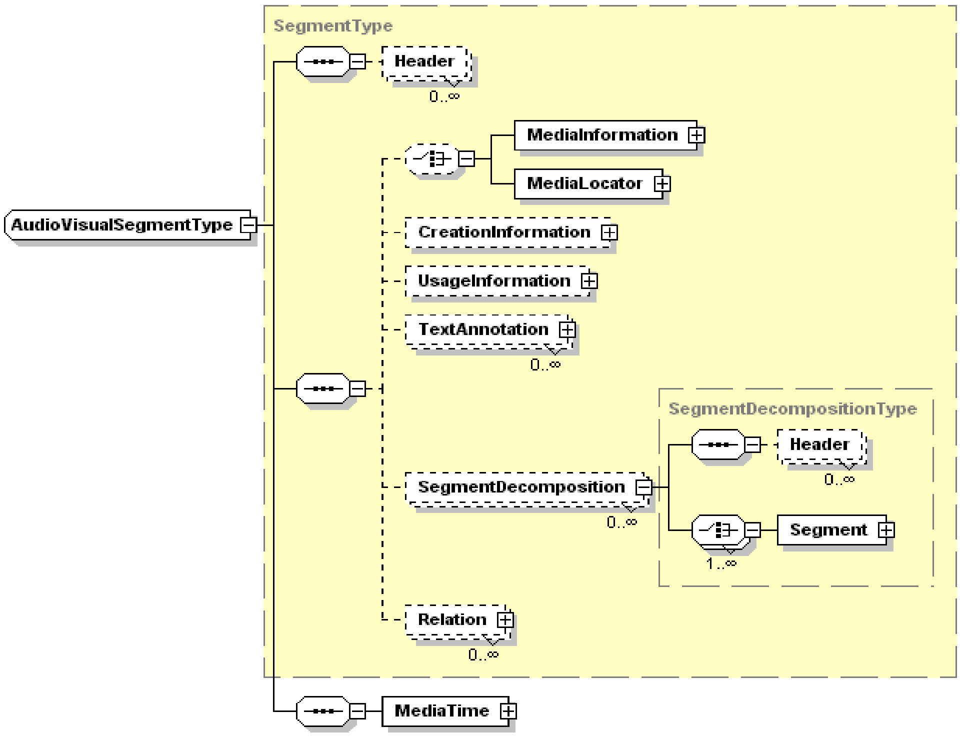

Figure 1: Audiovisual Segment Type of MPEG-7

The lower level of the DDL includes basic elements that deal with basic datatypes, mathematical structures, linking and media localisation tools as well as basic DSs, which are found as elementary components of more complex DSs. Based on this lower level, content description and management elements can be defined. These elements describe the content from five viewpoints [ISOe]:

- *Creation & Production* (describing the creation and production of the content),
- *Media* (description of the storage media),
- *Usage* (meta information related to the usage of the content),
- *Structural aspects* (description of the multimedia content from the viewpoint of its structure),
- *Conceptual aspects* (description of the multimedia content from the viewpoint of its conceptual notions).

The first three elements address primarily information related to the management of the content (*content management*) whereas the last two elements are mainly devoted to the description of perceivable information (*content description*). For instance, a segment can be decomposed into an arbitrary number of segments ("SegmentDecomposition"), which can be scenes or shots with an arbitrary number of temporal, spatial or content related relations to other segments. These segments can be described by additional elements. For instance, the "TextAnnotation" provides an unstructured or structured description of the content of the segment. An example of the structural content description scheme is given in Figure 1.

3. Related Works

There have been and are a number of MPEG-7-related projects that involve the adoption of MPEG-7 conformance [Hun99b]. The HARMONY project[1] aims at exploiting upcoming standards such as RDF, XML, Dublin Core and MPEG-7 to develop a framework allowing diverse communities to define descriptive vocabularies for annotating multimedia content. A second project is DICEMAN[2] with a broad objective in developing an end-to-end chain for indexing, storage, search and trading of audiovisual content, where MPEG-7 is used to index this content. A third project is AVIR, which aims at developing an end-to-end system for delivering of personalised TV services. AVIR has demonstrated how MPEG-7 metadata, delivered along with the content and used within a personal Electronic Programme Guide, could serve the non-IT expert user for automatic recording, later viewing, browsing and searching of broadcast video material.

The CMIP[3] developed an "Article-Based News Browser", a system that segments news, generates keyframes, and extracts additional information based on texts and icons. The system supports a browsing functionality, but not retrieval. Another project is COALA[4], which aims to design and implement a digital audiovisual library system for TV broadcasters and video archive owners with facilities to provide effective content-oriented access to internal and external end-users. The application domain is news and one of its goals is the application of MPEG-7 to news content [FA01].

The arrival of the MPEG-7 standard was an important evolution in modelling and representing multimedia content. To make possible the use of such rich content descriptions, the first important work would be the automatic generation of MPEG-7 description. This starts with the low-level feature extraction from audiovisual sequences, where the main concerns are the automatic segmentation of video sequences into shots using image processing algorithms, and information extraction using speech analysis and optical character recognition techniques, and mapping these low-level features to high-level concepts such as "sky", "sea", etc (e.g. [ABB01, Hau95, Sme00,ZTS+95]). However, the information extraction and semantic analysis to obtain these high-level concepts is still a user-centred task, since automatic extraction of semantic information is still considered as a task too complex to be exclusively carried out by computers [Sme00]. For example, user interventions have been used as a means to map low-level to high-level features in the two systems AMOS and IMKA [BZC+01].

The MPEG-7 descriptions whether created automatically or manually, are useful only once they can be correctly and easily searched for an adapted query language. Querying multimedia documents is an issue that has been the subject of many research studies. Two main categories of multimedia querying approaches can be distinguished in the literature: feature-based querying and semantic querying. The former refers to techniques that focus on the low-level multimedia features (colour, shape, etc.) such as query-by-example [Fli95,CCM+98,HGH+97] and query-by-sketch [ABL95]. The later refers to querying based on more high-level semantics that are closer to user's interpretations and the usage contexts. Various semantic query languages have been proposed. The simplest approaches use only the traditional content-oriented retrieval, i.e. they use keywords to describe the content [GB93]. This way of querying is however very limited, as it does not allow a

---

[1] See http://www.ilrt.bris.ac.uk/discovery/harmony/.
[2] See http://www.teltec.dcu.ie/diceman/.
[3] See http://www.lg-elite.com/MIGR/cmip/
[4] See http://coala.epfl.ch/

detailed specification of the content and the type of desired results. A set of semantic query languages has been proposed. In the present state-of-the-art, they are based on an extension of classical database query languages such as SQL and OQL [OT93,LYC98,HS96]. The most important problem with these languages is that they are not defined based on a study of the user's requirements. Each of these languages focuses only on a subset of the rich structure and content-based descriptions based on which users can query multimedia content.

Currently a few querying techniques are being proposed with the aim of retrieving multimedia content based on MPEG-7 descriptions. As MPEG-7 descriptions are XML documents a set of approaches [KKK03,TC02] propose to use directly XQuery as the retrieval language. These approaches have the disadvantage of considering only the XML document structure and ignoring the structure of MPEG-7 descriptions. A user formulating queries on top of MPEG-7 description would then require having an advanced knowledge of the MPEG-7 structure. Moreover, these approaches do not take into account the users' various description needs and viewpoints when retrieving the multimedia content.

A few proposals, such as [GL02,LS02], focused on the extraction of semantic information from MPEG-7 documents following the user's needs. This information could be in some cases not directly represented in MPEG-7, but deduced using for example an inference network model [GL02]. A specification of crucial issues for MPEG-7 queries is proposed in [LS02], which takes into account the implicit information to be extracted from MPEG-7 descriptions, such as spatio-temporal relations deduced from the points coordinates. Such semantic aspects are not directly expressible in XQuery, and therefore each of these approaches proposes their own specialised query languages.

An important requirement is therefore to provide a query language which from one side allows retrieving multimedia data based on a rich set of "users' descriptions" and which from the other side is adapted to search for multimedia data based on their "MPEG-7 descriptions". In order to fulfil these two requirements, we propose in Section 4 the "*Semantic Views Query Language*", an adaptation of XQuery that allows a high level description of the multimedia content based a user's semantic model, which is mapped into the MPEG-7 content descriptions.

Another critical issue in retrieving multimedia content based on MPEG-7 descriptions is the consideration of the structure of MPEG-7 descriptions and its implications such as the retrievable units, ranking of the results and so forth. As mentioned above, MPEG-7 descriptions are XML documents, therefore they display a structure; they are composed of elements. With MPEG-7 descriptions, the retrievable units should be the elements as well as the whole audiovisual sequence. That is, the retrieval process should return elements at *various levels of granularity*, for example, a video segment when only that segment is relevant, a group of segments, when all the segments in the group are relevant, or the full video itself, when the entire audio-visual sequence is relevant.

Methods to access MPEG-7 documents are, therefore, common to those for structured documents in general, and in particular XML documents. For retrieving structured documents or XML documents, the indexing process or the retrieval function has to pay attention to the structure of a document. In information retrieval, several approaches to structured document retrieval have been developed. The so-called passage retrieval (e.g. [Cal94,SAB93,Wil94]) determines retrieval weights for passages (paragraphs, sentences) to compute a final retrieval weight for a set of passages. Other approaches aim at computing a retrieval weight for each document component, so to find the *best entry points* into a structured document thus allowing for returning elements at various level of granularity (e.g. [CMF96,Fri88,LM00,LR98,MJK+98,Röl99]). Our approach for accessing multimedia data through their associated MPEG-7 description, which is described in Section 5, follows this notion of best entry points.

4. MPEG-7 querying using the Semantic Views Model

In this section, we propose an approach to retrieve MPEG-7 descriptions, which takes into account the user viewpoints and requirements via a semantic model called the Semantic Views Model. In the following we first describe the principles of the Semantic Views Model (Section 4.1). To represent formally the Semantic Views Model, we realised a set of experiments using XML Schema and RDF Schema. We briefly present the results of these experiments and the reasons for our choice of XML Schema (Section 4.2). To provide a high-level query language following the principles of the Semantic Views Model, we propose the Semantic Views Query Language, SVQL (Section 4.3). We next describe the formal syntax of SVQL as an adaptation of XQuery (Section 4.4). Finally we describe how SVQL queries are mapped into MPEG-7 descriptions via a layered retrieval architecture (Section 4.5).

4.1 Semantic Views Model

Current research approaches in the domain of audiovisual retrieval are mainly focused on the technological challenges of audiovisual automatic indexing. Most of these approaches give little attention to the users' semantic retrieval requirements. There exist a few studies on cognitive behaviours and requirements of users in the procedure of still image indexing and retrieval (e.g. see [Sha94,Jör99,OO99]). However, there is yet little research on the user's behaviours and needs in video information retrieval systems, and little design methodology proposed for creating audiovisual information retrieval systems based on users' study.

To provide an abstract model that takes into account various users requirements and viewpoints in the process of multimedia retrieval, we realised a detailed study of TSR (Television Suisse Romande) production and archiving environment. Our goal was to investigate questions such as "who are the different users of an audiovisual retrieval system?", "what do the users search for?", "how do professional users search for audiovisual content?", "what should an audiovisual retrieval system provide to the users in terms of conceptual model, retrieval strategies, tools and interactive retrieval mechanisms?".

There were two central advantages in studying the professional environment of TV news production and archiving: firstly, the variety of the users (producer, journalist, video editor, archivist, etc.) and consequently their various requirements, and secondly the expertise of the professional users in retrieving audiovisual information, achieved by an extensive practical experience in the domain. In [Fat03], we describe in details the current strategies and tools used in audiovisual production, archiving and retrieval, and we analyse the users' problems in the current system.

Analysing the different queries put forward by TV news professional and non-professional users of the TSR showed us that users adopt five different *Views* to express their requirement: *PhysicalView*, *ProductionView*, *ThematicView*, *VisualView* and *AudioView*. The following example (Figure 2) shows one such query and analyses how the user expresses the query via different *Views*.

Find A news item in the context of Euro 2000 football games containing a shot of at least 5 seconds

showing a French football supporter saying « que le meilleur gagne »

Figure 2: A multimedia query example

Figure 3 shows our analysis of the above query based on the different viewpoints that the user adopts in expressing the query. In the Semantic Views Model, each View is described using five elements: *BasicViewEntity*, *ViewDescriptions*, *IntraViewRelations*, *InterViewRelations* and *ViewOperators*. BasicViewEntity is the atomic unit of description in each View. ViewDescriptions express different characteristics of the BasicViewEntity in each View. InterViewRelations express the correspondence relation between the BasicViewEntities that belong to different Views. IntraViewRelations express the relations between the BasicViewEntities that belong to the same View.

A detailed description of the Semantic Views Model and its formal definition is described in [Fat03].

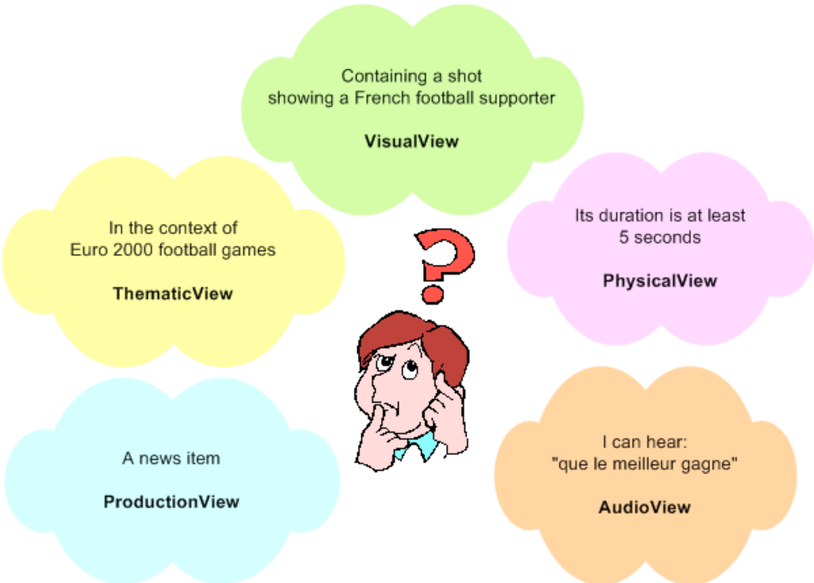

Figure 3: Analysis of the different views that users adopt in expressing a multi-media query

4.2 Representation of the Semantic Views Model

To provide a formal representation of the Semantic Views Model, we examined two different approaches using W3C XML Schema [XML] and RDF Schema [RDF]. In the approach based on RDF Schema, we were biased towards the conceptual graphs (CGs) semantics [Sow84]. RDF and CGs have very similar semantics. According to Tim Berners-Lee [Ber01] there is a huge overlap between these two technologies, making them very comparable and interworkable. Another advantage of using CGs semantics is the querying and inference capabilities enabled by the CG formalism. In the approach based on XML Schema our motivation was driven by the fact that MPEG-7 is modelled in XML Schema and therefore it should be more convenient to map an XML based model on top of MPEG-7.

Our experiments (described in detail in [Fat03]) show that RDF allows an accurate representation of the Semantic Views Model. It is well adapted to represent relational models such as a CGs. However, during the design of the Semantic Views Model with RDF, we noticed an important shortcoming: many descriptions that are needed in Semantic Views Model already exist in MPEG-7. Ideally, we should use these description tools as they are and use the RDF Schema to represent the rest of the descriptions that are purely related to Semantic Views Model and which regroup and relate MPEG-7 descriptions in our desired structure. Unfortunately, we observed that RDF Schema is not capable of reusing XML Schema datatypes; i.e. it is not possible to refer to an XML Schema data types via the URI values of the domain and range in an RDF property. This problem represents one of the weaknesses of the RDF Schema for our purpose. To create the RDF Schema of the Semantic Views Model we had to rewrite the whole set of required descriptions in RDF. Another shortcoming of RDF Schema in modelling the Semantic Views Model concerns the definition of the hierarchical structure of production. This problem also noted in [Hun99b] is because RDF Schema does not allow a property to have multiple ranges or domains.

The representation of the Semantic Views Model using XML Schema has a central advantage in that it allows the reuse of the description tools that are already defined in MPEG-7. However, when reusing MPEG-7 descriptions a few problems occur that are related to the weakness of MPEG-7. For example, one of the problems is that in MPEG-7 some datatypes are defined locally, i.e. inside another type definition. It is thus impossible to reuse such datatypes. Nevertheless, XML Schema has many other advantages besides allowing the reuse of MPEG-7 description tools: the representation of a hierarchical structure, such as production structure is much easier and natural using XML Schema; also a set of structural constraints, such as cardinality and range, can be represented using XML Schema, and are useful in defining a well-structured model. Based on these observations, we chose XML Schema to represent the Semantic Views Model. A complete description of the model is provided in [Fat03].

4.3 Semantic Views Query Language

SVQL is a high-level query language that allows users to express their requirements following the Semantic Views Model in a concise, abstract and precise way. Based on SVQL, users can retrieve multimedia information described by MPEG-7 standard, without getting involved into the implementation details. The language can be used both by the end-users and by application programmers. The former allows users with a high-level query language that is relatively easy-to-use and abstract. The latter presents a useful tool that can facilitate and speed-up the development of applications based on the Semantic Views Model.

The principal features required in the expression of a query in SVQL are:

- Identification of basic units of descriptions in different Views, i.e. the *BasicViewEntities*.
- Description of the characteristics of the *BasicViewEntities* using a set of *ViewDescriptions.*
- Expression of different relationships between the *BasicViewEntities* using the *IntraViewRelations* and the *InterViewRelations*.

Figure 4 shows an example of how the above principles are expressed in SVQL. We refer to the same example as the one given in Figure 2. As it can be observed, the query is based on a “**LET-WHERE-RETURN**” structure, very close to the FLWR structure of XQuery (“FOR-LET-WHERE-RETURN”), with a small difference that we do not use the FOR keyword. This type of query syntax, refereed to as “keyword oriented syntax” is used in the most well-known query languages, such as SQL, OQL, and XQuery, and is a familiar mode of query expression for specialized end-users and application programmers.

LET	$semanticViews := *semanticViews* (“D:/News/news12-06-2001.xml”),
	$newsItem **:=** *newsItem* ($semanticViews),
	$fact **:=** *fact* ($semanticViews),
	$shot **:=** *shot* ($semanticViews),
	$videoSegment **:=** *videoSegment* ($semanticViews),
	$speech **:=** *speech* ($semanticViews),
WHERE	*match* (getDescription ($fact, Event), *event* (“EURO 2000 football games”)) AND
	match (getDescription ($shot, Person), *person*(„“French football supporter”)) AND
	greaterThan (getDescription ($videoSegment, Duration), *duration* (“5s”)) AND
	match (getDescription ($speech, SpeechTranscription), *speechTranscription* («Que le meilleur gagne ») AND
	corresponds ($videoSegment, $newsItem, $fact, $shot, $speech)
RETURN	$videoSegment

Figure 4: A query formulated using SVQL

As can be seen in the above query, the **LET** clause contains two types of expressions:

- In the first expression, the *SemanticViews* () function is called with the name of the file containing the MPEG-7 instances to be retrieved. This function creates the *Semantic Views Document* corresponding to the MPEG-7 file on the fly.
- In the next series of expressions, a set of functions are called to get different required *BasicViewEntities* of the *Semantic Views Document* and to assign them into a set of variables. Each function name indicates the type of the *BasicViewEntity*, e.g. *NewsItem*, Fact, Shot, *VideoSegment*, and *Speech*.

The **WHERE** clause also contains two different types of expressions:

- The first four expressions represent a set of conditions that should be held on the *BasicViewEntities* of different Views. These conditions are expressed via a set of *ViewOperators*: here *match*, and *greaterThan* are used.
- The last expression represents the condition concerning the *InterViewRelation* of *BasicViewEntities* via *corresponds* operator. It determines which of the cited *BasicViewEntites* correspond to each other. This expression is used if more than one View is used in the query.

Finally, the **RETURN** clause identifies a variable containing a *BasicViewEntity* to be returned to the user. In the above query the variable containing the *VideoSegment* is returned.

This mode of query formulation facilitates the description of the user's requirement. The user only needs to specify the MediaSegment he/she is looking for by characterizing it in different Views via the **LET** and **WHERE** clauses. He/she then asks for the MediaSegment viewed in one of the introduced Views in the **RETURN** clause.

4.4 SVQL Syntax

The syntax of SVQL is a specialization of the syntax of XQuery. This design feature of SVQL has two main advantages. Firstly, as we mentioned above, this type of syntax allows a familiar mode of query expression for specialized end-users and application programmers. Secondly, it allows a straightforward implementation of SVQL based on XQuery. We will describe the implementation details of SVQL in Section 4.5. In the following we present the summarized syntax of SVQL in Figure 5.

The syntax shows the expressiveness and the abstract level of SVQL by focusing on the essential features of the conceptual Semantic Views Model. In the following paragraphs we describe the details of the SVQL syntax.

In the **LET** clause, there are five types of expressions:

- The *SemanticViewVar* consists of an XQuery variable [CR01] i.e. in the form of "$" *QName*, where the *QName* specifies the name of the variable.
- *SemanticViewsCreation* corresponds to an XQuery function call [XQuery]. Specifically, the *semanticViews* () function is called to create *a Semantic Views Document* on-the-fly. The *Semantic Views Document* is created based on the MPEG-7 file whose name is passed to the *semanticViews* () function.
- The *BasicViewEntityVar* is also an XQuery variable in the form of "$" *QName*. This variable takes as values any path corresponding to the *BasicViewEntities* returned by *BasicViewEntity-Location*.
- The *BasicViewEntityLocation* corresponds to an XQuery function call [XQuery]. The function name here is the name of the *BasicViewEntity* that the user wants to select, such as *NewsProgram*, *Shot*, *Fact*, etc. The function's argument is the *SemanticViewVar* which corresponds to *the Semantic Views Document* created-on-the fly.
- The *SVQExpression* corresponds to a complete Semantic Views Query in the form "LET-WHERE-RETURN". As a result SVQL queries can be nested.

```
SVQExpression :=
                    "LET"    SemanticViewVar ":=" SemanticViewsCreation

          (BasicViewEntityVar":="(BasicViewEntityLocation    |SVQExpression))+

                    "WHERE"   Condition (("AND" |"OR") Condition)*

                    "RETURN"  (BasicViewEntityVar)+

Condition                    :=     ViewDescriptionCondition   | IntraViewRelationCondition |
                                           InterViewRelationCondition
SemanticViewsCreation              :=      "semanticView(" SourceFile ")"
BasicViewEntityVariable            :=     "$" QName
Sourcefile                         :=     QName
BasicViewEntityLocation            :=     BasicViewEntity "(" SourceFile ")"
BasicViewEntity          :=    "NewsProgram" | "NewsItem" | "Summary"  |"Presentation" | "Report" | "Interview"
                                |"AudiovisualSegment" | "AudioSegment" |"VideoSegment" | "Fact"
                                |"EditedVideo"|"Shot" |"Rush"| "EditedAudio" | "Speech"
|"AmbinaceSound" |"Jingle"
ViewDescriptionCondition     :=    ViewDescriptionOperator
IntraViewRelationCondition   :=    IntraViewRelationOperator
InterViewRelationCondition   :=    InterViewRelationOperator
```

Figure 5: SVQL Syntax

The **WHERE** clause is a Boolean expression composed of a set of *Condition*s. There are three different types of Conditions:

- *ViewDescriptionCondition*, which uses a *ViewDescriptionOperator* to compare two *ViewDescriptions*. The *ViewDescriptionOperatrors* are defined in details in [Fat03]. There are two main types of *ViewDescriptionOperators*: textual operators, such as *match*, which compares two textual *ViewDescriptions*, and arithmetic operators, such as *equal*, *greaterThan* and *lessThan*, which compare the numeric value of two *ViewDescriptions*.
- *IntraViewRelationCondition*, which uses an *IntraViewRelationOperator* to verify the relationship between two *BasicViewEntities* inside the same *View*. The *IntraViewRelationOperator* are defined in details in [Fat03]. Examples of such operators are *contains*, which determines if a *BasicViewEntity* has *IntraViewRelation* with another *BasicViewEntity.*
- *InterViewRelationCondition*, is represented via *corresponds* operator. This operator takes a set of *BasicViewEntityVars* as its arguments and determines if the corresponding *BasicViewEntities* have *InterViewRelation*.

In the **RETURN** expression the *BasicViewEntityVars,* determine a set of *BasicViewEntities* to be returned.

4.5 Processing of Semantic Views Queries

SVQL allows the retrieval of multimedia data described in MPEG-7 format. In order to map an SVQL query into MPEG-7 descriptions, we propose a layered architecture depicted in Figure 6. This figure shows different layers between the Semantic Views, which is the way the user interprets the content, and the MPEG-7 instances, which is the standard description of the multimedia data.

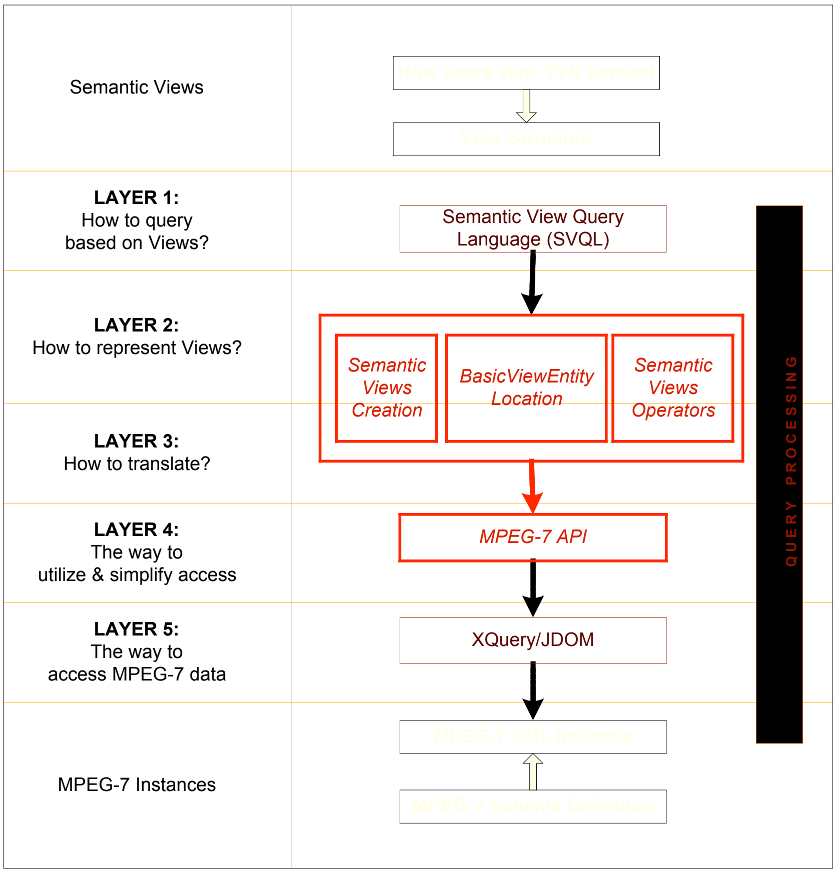

Figure 6: Layered retrieval architecture of SVQL processing

The user's query is first formulated following SVQL. As described before, the syntax of *SVQL* is a specialization of the syntax of XQuery. This specialization is realized by defining a set of functions and operators that represent the basic functionalities of Semantic Views Model at an abstract level.

The processing of SVQL queries is then realized by an XQuery processor taking care of the SVQL functions and operators. These are represented as XQuery internal functions calls. Three different types of functions are applied in an SVQL query:

- **Semantic Views Document Creation**: the on-the-fly creation of *Semantic Views Document* via *SemanticViews()* function,
- **BasicViewEntity Location**: a set of functions (such as *NewsProgram*, *Shot*, etc.) which locate *BasicViewEntities* inside the *Semantic Views Document* whose path is passed in the parameter of the function.
- **Semantic Views Operators**: operators, such as *match*, and *corresponds*, which are used inside *Conditions*.

Eventually, in order to facilitate and standardize the access to MPEG instances, we have defined a set of functions, named **MPEG-7 API**, which are called by the three sets of functions mentioned above. The advantage of MPEG-7 API is that they can be reused in any application that needs to manipulate MPEG-7 descriptions. The detailed description of MPEG-7 API is given in [Fat03].

5. MPEG-7 retrieval model

The previous section described a query language, SVQL, based on semantic views that users may adopt to express their information need. The processing of an SVQL query does not allow for ranking of MPEG-7 descriptions according to how well they satisfy user information needs and considering the uncertainty inherent in the representation of documents (here the MPEG-7 descriptions) and the formulation of the information need. In this section, we add a new function to the MPEG-7 API described in Figure 6, with the aim to provide a query processing that captures the uncertainty and provides ranking. This added functionality is done through the development of a *Retrieval Model* for searching multimedia material based on their associated MPEG-7 descriptions. First, we examine the requirements of a *Retrieval Model* for MPEG-7 (Section 5.1). Second, we present HySpirit, the software development kit that we used to develop and implement the retrieval model (Section 5.2). Third, we describe in details the actual design and implementation of the retrieval model using HySpirit (Section 5.3).

5.1 Requirements of a Retrieval Model for MPEG-7

MPEG-7 DSs define the schemes for representing structure, content and relationships of multimedia data; MPEG-7 DSs are specified as XML schemes. An MPEG-7 description is an instance of a DS, so we can consider an MPEG-7 description as an XML document. An XML document is a structured document, in the sense that the XML format is one way to capture the structure of a document. With this view in mind, the requirements for a model for structured document retrieval, and in particular, XML document retrieval, apply to MPEG-7 retrieval.

The first requirement applies to any retrieval model. We need a "*relevance-based" ranking function*, so that weights (e.g. probability values) are assigned to elements (e.g. segments) forming a retrieval result, reflecting the extent to which the information contained in an element is relevant to the query. This is particularly important when searching large repositories of multimedia data because it captures the uncertainty inherent to the retrieval process so that best matches (i.e. the most relevant elements) are displayed *first* to users.

A crucial requirement for structured document retrieval is that the *most specific element(s)* of a document should be retrieved (i.e. returned to the user). For MPEG-7 retrieval, this means that not only an entire video but also video parts can constitute a retrieval result, depending on how they match the query. Therefore, a retrieval model for MPEG-7 must determine the *best entry points* into the MPEG-7 structure. For example, suppose that a news broadcast (e.g. "AudioVisual" DSs) is structured into several news clips (e.g. "Segment" DSs). For a generic query, the entire news broadcast (i.e. the "AudioVisual" segment) would be an appropriate retrieval result, whereas for specific query particular news clip (one "Segment") would constitute a better retrieval result.

A third requirement relates to the *relationships between elements*, such as spatial and temporal relationships. In classical retrieval, we deal with independent documents and simple propositions in documents (e.g. terms occurring in a document). With structured documents, the relationships between the elements (here MPEG-7 DSs and Ds) must be considered, in particular for determining the most relevant document elements. Besides spatial and temporal relationships, relationships such as links (e.g. pointing to additional material such as an HTML page), order (sequence) and others should also be captured.

With respect to XML documents, the use of *attributes* leads to a further requirement. Standard information retrieval, which has one of its aims the representation of the content of documents, will treat the attributes of an XML document as its content, and hence it will not explicitly model attributes. Attributes are used in databases to characterise entities (e.g. Name, Address, and Age of Person entity). In standard database approaches, content is often considered as an attribute, and again, there is no conceptual support that distinguishes between content and attributes (e.g. [ACC+97]). For accessing XML documents, and hence MPEG-7 descriptions, more refined retrieval methods are necessary to distinguish between attributes, which constitute mostly factual knowledge, and content of XML documents.

The next requirement arises from one of the goals of MPEG-7, which is to describe multimedia data in terms of the *objects* (persons, places, actions, etc.) that occur in them. The representation of objects in a scene is part of an MPEG-7 instance (i.e. descriptors “Who”, “Where”, “WhatAction”, etc.). Those objects add a new dimension to a retrieval model, when we can distinguish between content-bearing entities (retrievable document units) such as videos and video segments, and “semantic” entities such as persons, actions, etc.

The next requirement refers to the *data propagation* of MPEG-7 descriptions. That is, some attributes and elements (e.g. descriptors) defined at upper levels may be valid for elements at lower levels. For example, a “FreeTextAnnotation” specified for a video description root is the description of all contained video elements, if not specified at the video segment level. A retrieval model for MPEG-7 should be able to specify which elements, if any, are propagated up or down an MPEG-7 structure.

As users come with varying background, experience and interest, a last requirement of a model for MPEG-7 retrieval is the conceptual integration of user profiles into the retrieval model, so that an element is not only retrieved with respect to its information content, but also according to user preferences.

5.2 HySpirit

HySpirit is a software development kit [RLK01], which provides support for representing complex documents and describing retrieval functions. HySpirit is based on generic and well-established approaches to data and information management such as relational database model, logic, and object-orientation. HySpirit therefore provides the necessary expressiveness and flexibility for capturing content (i.e. “being about”) and fact (e.g. “author”, “date”) information in MPEG-7 descriptions. As such it is particular suited to deal with both textual queries (as it is done in classical information retrieval) and more sophisticated queries such as SVQL queries. In addition, HySpirit supports the design and implementation of retrieval models for data with an underlying structure, a key characteristic of MPEG-7 documents, thus allowing the retrieval of elements at various level of granularity. Finally, HySpirit provides means to represent the uncertainty inherent to the information retrieval process, which leads to a relevance-based ranking of the retrieval results.

HySpirit represents knowledge (content, fact, and structure) by a *probabilistic object-oriented four-valued predicate logic* (POOL). The object-oriented nature of POOL was motivated by F-Logic [KLW95], which combines object-oriented principles (e.g. classification and attribute values) with logical rules (e.g. Datalog rules). The semantics of POOL is based on the semantic structure of modal logics [HM92]. This allows for a context-dependent interpretation of knowledge representation, which is necessary for modelling the nested structure of MPEG7 descriptions. The uncertainty of knowledge is modelled with a probabilistic-extended semantics [FH94]. The retrieval functions are implemented as *inference* process based on the logical approach to information retrieval (see [Rij86]), which computes the probability $P(d \rightarrow q)$ that a document d implies a query q. For implementation issues and the integration of relational database management system, POOL is mapped onto a *probabilistic relational algebra* (PRA) [FR97].

5.3 Design and Implementation of the Retrieval Model with HySpirit

This section describes the procedure followed to develop and implement the retrieval model using HySpirit. POOL is first used to provide a probabilistic representation of MPEG-7 data and MPEG-7 based queries (Section 5.3.1). The POOL representation is then mapped to a PRA representation (Section 5.3.2). The PRA representation is then interpreted by an inference engine (retrieval functions) that produces a ranked list of results (Section 5.3.3).

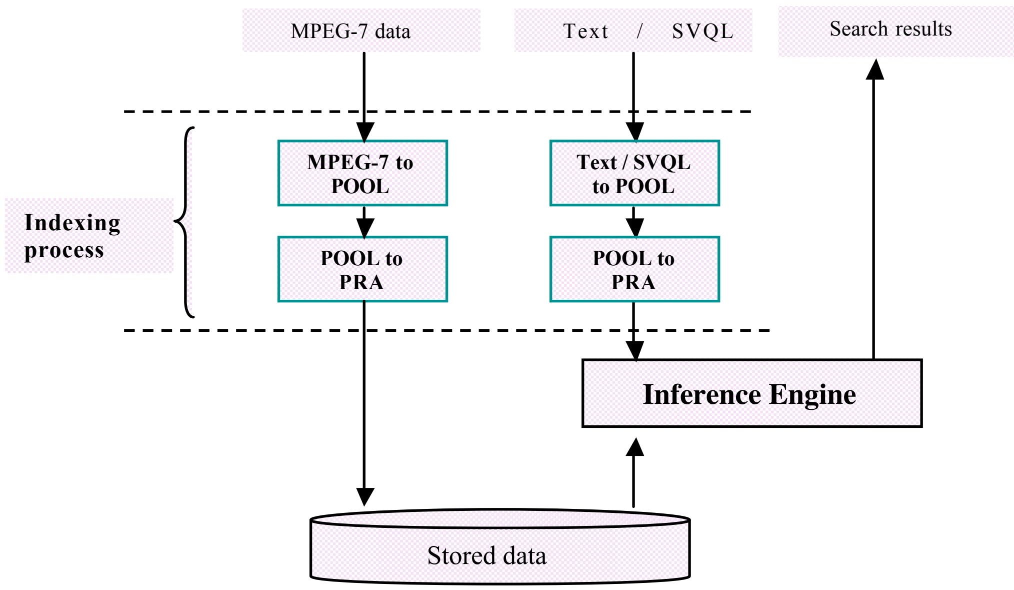

Figure 7: The architecture of the system

The overall architecture of the system that implements the model is shown in Figure 7. MPEG-7 data (DS instances) are indexed (MPEG-7 to POOL and POOL to PRA as explained above). The indexed data is then stored. Queries can be formulated as stand-alone keywords (text queries) or as structured queries formulated with SVQL. The formulated queries are mapped into POOL and then PRA representations. The inference engine compares the indexed queries to the indexed MPEG-7 data, which results in a ranked list of entry points (i.e. segments at various granularity levels of the hierarchical structure) in the MPEG-7 data. The entry points constitute the search results.

```
<AudioVisual xsi:type="AudioVisualSegmentType">
  <CreationInformation>
   <Creation>
     <Title>Spain vs Sweden</Title>
     <Abstract><FreeTextAnnotation>Spain scores a goal quickly in this World  Cup soccer game against
         Sweden. The scoring player is Morientes. </FreeTextAnnotation></Abstract>
     <Creator>BBC</Creator>
   </Creation>
   <Classification>
     <Genre type="main">Sports</Genre>
     <Language type="original">English</Language>
   </Classification>
  </CreationInformation>
  <TextAnnotation><FreeTextAnnotation>Soccer game between Spain and Sweden.
     </FreeTextAnnotation></TextAnnotation>
  <SegmentDecomposition decompositionType="temporal" id="shots" >
   &lt;Segment xsi:type="VideoSegmentType" id="ID84"&gt;
       <MediaLocator> (?) </MediaLocator>
     <TextAnnotation><FreeTextAnnotation>Introduction.</FreeTextAnnotation></TextAnnotation>
  &lt;/Segment&gt;
  &lt;Segment xsi:type="VideoSegmentType" id="ID88"&gt;
    <MediaLocator> (?) </MediaLocator>
    <TextAnnotation><FreeTextAnnotation>Game.</FreeTextAnnotation></TextAnnotation>
  &lt;/Segment&gt;
</SegmentDecomposition>
</AudioVisualContent>
```

Figure 8: Extract of a MPEG7 Description

Throughout this section, we illustrate the procedure using the extract of a sample MPEG-7 description of a soccer game[5] (shown in Figure 8). The extract consists of an audiovisual segment ("AudioVisualSegmentType"), composed of two sub-segments ("SegmentDecomposition"). Creation information is provided for the audiovisual segment, such as a "Title", an "Abstract", the "Creator", the "Genre" and "Language" (the content management part of MPEG-7). The segment has also a free text annotation. The sub-segments ("VideoSegmentType") correspond to video shots. Each sub-segment has a free text annotation component.

5.3.1. From MPEG-7 Description to POOL Representation

POOL is a probabilistic object-oriented logic, which enables the integration of content-based and fact-based querying, as well as the structure of documents. The *knowledge of* (content) or *knowledge about* (fact) multimedia data is expressed in terms of POOL programs. These combine:

- object-oriented modelling concepts like aggregation, classification, and attributes,
- classical information retrieval concepts such as weighted terms,
- probabilistic aggregation of knowledge necessary for structured document retrieval.

The retrieval units are modelled as *contexts*, where a retrievable unit can be any "Segment" DS of the multimedia sequence at any level of granularity. This includes the root segment, corresponding to the complete video sequence, or a segment corresponding to a particular scene or shot in the video (see Figure 8).

Since a video sequence is decomposed into segments (e.g. scenes), which can themselves be decomposed into (sub-)segments (e.g. shots), etc. (as represented by the "SegmentDecomposition" element modelling the hierarchical structure of the video), contexts are nested into each other. The retrieval process should therefore return the most relevant level(s) of granularity. For example, for a long video sequence with only one relevant scene, the scene (i.e. sub-segment level) should be retrieved instead of the whole video (i.e. segment root level).

Representation of MPEG-7 Descriptions An example of a POOL program that illustrates the MPEG-7 description of Figure 8 is given in Figure 9. The POOL program consists of a set of *clauses*, where each clause is either a context or a *proposition*. The *nested* contexts represent the structure, whereas the propositions represent the content (terms) and the attributes at the respective context level.

```
audiovisualsegment(audiovisualsegment_1) % classification of object audiovisualsegment_1
audiovisualsegment_1.title("Spain vs. Sweden")
% optionally created in addition to the title_1 context
audiovisualsegment_1.creator("BBC")
audiovisualsegment_1[
   title_1["Spain vs. Sweden" 0.8 spain 0.8 sweden]
   % probabilities express uncertain contribution of terms to title content
   % title is modelled both as content and as attribute
   abstract_1[spain scores a goal  …]
   soccer game between spain and sweden

   segment_1.medialocator("(?)")
   segment_1[introduction]

   segment_2.medialocator("(?)")
   segment_2[game]
]
audiovisualsegment_1.genre("Sports")
audiovisualsegment_1.language("English")
```

Figure 9: Example of the POOL representation of the MPEG-7 Description of Figure 8

[5] This extract is based on the Monster Description of the MPEG soccer game video, 2000.

The first clause classifies the context "audiovisualsegment_1" to be an instance of the class "audiovisualsegment". The second clause states that "Spain vs. Sweden" is the title of the context "audiovisualsegment_1". The third clause means that "BBC" is the creator of the context "audiovisualsegment_1". These three clauses express facts, i.e. knowledge about the context "audiovisualsegment_1" (its type, its title and its creator). The fourth clause reflects the structure of "audiovisualsegment_1", which is composed of four sub-contexts: "title_1", "abstract_1", "segment_1" and "segment_2". The context "audiovisualsegment_1" is called the super-context of these four sub-contexts. The content of "audiovisualsegment_1" is given by the terms "soccer game between spain and sweden". The content of "segment_1" is the term "introduction"; that of "segment_2" is the term "game".

Fact vs. Content Contexts have a unique identifier, a content and a set of attributes. This is driven by a conceptual distinction between content-oriented and factual querying. In the context of MPEG-7, this would mean that, from the descriptors of a segment that can be exploited for searching, some are considered to represent the content (knowledge) of a segment, and others are considered to represent facts (knowledge) about these segments. For example, a query seeking videos about dinosaurs is considered a content-oriented query, whereas a query seeking videos in English produced by BBC is a factual query. In this sense, seeking a BBC documentary about dinosaurs corresponds to both a content-oriented and factual query. In our representation, we consider the "Title" descriptor to contribute to the content of a segment because it is composed of keywords that can be used to rank the corresponding segments with respect to a given query. We also consider "Title" to be a fact about the segment because users may give an exact title as their query. Therefore, the "Title" descriptor is translated to both knowledge of (content) and knowledge about (fact) the context "audiovisualsegment_1" in Figure 9.

Note that not all available MPEG-7 descriptors contribute to deriving the content of a video (for searching purposes). For example, "Media Time" or "Media URI" provides technical information such as links etc. Therefore, the retrieval is not based upon such descriptors.

The "Text Annotation" DS and its elements are considered to describe the content of a segment, whereas elements of the "Creation" DS and Classification DS are considered to describe facts about the content, with the exception of the "Title" and the "Abstract" descriptors. "Abstract" was thought to express the content, whereas "Title" was considered to express both some content as well as the fact of a segment being entitled as such. Therefore, when modelling the MPEG-7 data in POOL, we classify the above descriptors as content or as attributes of the segment context, respectively. We also include the "Media Time" and the "Media Locator" elements to be attributes of the segment context, since they provide information necessary for locating and presenting the retrieved segments.

Querying MPEG-7 Descriptions An information need such as "I am looking for every instance of a goal" can be described as querying for all contexts (segments) where the logical formula (the proposition) "goal" is true. For example, the POOL query

```
?- D[goal]
```

searches for all contexts D where the formula "goal" is true (the result here consists of the two contexts "audiovisualsegment_1" and "segment_1"; this is explained later in this section). An example of a combined content and factual query is

```
?- D[goal] & D.title("Spain vs. Sweden")
```

This query combines a typical information retrieval criterion referring to the content (all contexts about "goal") with a typical database selection referring to the attribute values (all contexts with title 'Spain vs. Sweden"). The result consists of one context, "audiovisualsegment_1". The query corresponds to a conjunction (AND combination). A disjunction (OR combination) query is expressed via *rules*. For instance, the following query (which is also a combined content and factual query)

```
retrieve(D) :- D[goal]
```

retrieve(D) :- D.title("Spain vs. Sweden")

searches for all contexts (segments) about (showing a) "goal" *or* with title "Spain vs. Sweden". A disjunctive query will retrieve a higher number of contexts; however, the ranking function will assign higher retrieval weights to the contexts fulfilling both rules (showing a "goal" *and* with title "Spain vs. Sweden").

Structure A major concern in an MPEG-7 retrieval model is to capture the hierarchical structure of the MPEG-7 descriptions for determining the entry points. Consider the following modified extract from Figure 9:

```
audiovisualsegment_1[ segment_1 [introduction game]
                      segment_2 [goal game] ]
```

This program expresses that ""audiovisualsegment_1" is composed of two segments, "segment_1" and "segment_2", and the content of these segments is given by the terms "introduction game" and the terms "goal game", respectively. The query

?- D[introduction]

retrieves both "segment_1" and "audiovisualsegment_1"; they both constitute entry points. The context "segment_1" is retrieved, since the term "introduction" occurs in "segment_1", whereas the context "audiovisualsegment_1" is retrieved, since "segment_1" contains the term "introduction" and it is part of "audiovisualsegment_1." Consider the following query:

?- D[introduction & goal]

The conjunction "introduction & goal" is true in the context "audiovisualsegment_1(segment_1, segment_2)", i.e., the context that consists of both sub-contexts "segment_1" and "segment_2". The term "goal" is true in "audiovisualsegment_1(segment_1, segment_2)" since it is true in "segment_2", and the term "introduction" is true in "audiovisualsegment_1(segment_1, segment_2)" since it is true in "segment_1". Neither sub-context on its own ("segment_1" or "segment_2") satisfies the query; only the context "audiovisualsegment_1(segment_1, segment_2)", i.e. "audiovisualsegment_1", satisfies the query. In other words, only "audiovisualsegment_1" is an entry point for that query. We call "audiovisualsegment_1(segment_1, segment_2)" an *augmented* context since its knowledge is augmented by the knowledge of the sub-contexts. An augmented context *accesses* its sub-contexts.

Uncertainty A major task of the information retrieval process (e.g. in the indexing phase) is the incorporation of the intrinsic uncertainty in representing documents and queries. Unlike XML, POOL provides probabilities that can be used to reflect this intrinsic uncertainty. POOL programs address two dimensions of uncertainty: the uncertainty of the content representation, and the uncertainty that a super-context accesses its sub-contexts.

For the uncertain content representation, probabilities can be defined. For instance, in Figure 9, a probability value of 0.8 is assigned to the terms "spain" and "sweden", which means that the probability that the terms "spain" and "sweden" is true (false) is 0.8 (1.0 – 0.8 = 0.2) in the context "title_1", respectively. This could also be read as follows: 0.8 (0.2) is the probability that the term "spain" is (is not) a good indicator of the content of the context "title_1".

MPEG-7 provides tools that address the organisation of content (*content organisation*) [MPE00] that may be used to estimate these probabilities. For instance, the "Probability Model" DS provides a way to describe statistical functions for representing samples and classes of audiovisual data and descriptors using statistical approximation. A second DS, the "Analytic Model" DS, provides a way to describe properties of groups of objects, groups of descriptors and classifiers that assign semantic concepts based on descriptor characteristics, training examples and probabilities models. In our current implementation of the search component, such data was not available. Therefore, we use standard statistic-based techniques from information retrieval. These are based on term frequency information (i.e. how often a term occurs in an element) and inverse document frequency (how many elements contain a particular term) [BR99].

The uncertain access reflects the effect of a sub-context on the knowledge of an augmented context. A weight can precede the opening of a sub-context. Consider the following modified extract of Figure 9:

```
audiovisualsegment_1[ 0.5 segment_1 [0.8 goal]
                      0.5 segment_2 [0.6 goal]]
```

In context “segment_1”, “goal” is true with a probability of 0.8. In “segment_2”, “goal” is true with a probability of 0.6. These probability values reflect the uncertain indexing of the two sub-contexts as described above. The two sub-contexts “segment_1” and “segment_2” are accessed by “audiovisualsegment_1” with a probability of 0.5. This probability reflects the effect of the knowledge of “segment_1” and “segment_2” on the augmented knowledge of “audiovisualsegment_1”. The query

```
?- D[goal]
```

retrieves three contexts (i.e. identifies three entry points) with the following probabilities:

```
0.8 segment_1
0.6 segment_2
0.58 audiovisualsegment_1(segment_1, segment_2)
```

The sub-contexts are retrieved with the probabilities of “goal” being true in them. The augmented context “audiovisualsegment_1(segment_1, segment_2)” is retrieved with a probability of 0.58 which is the summation of three probabilities[6]: the probability that goal is true if both sub-contexts are accessed (0.5□0.5□0.8□0.6+0.5□0.5□0.8□(1-0.6)+0.5□0.5□(1-0.8)□0.6=0.23) plus the probability that goal is true if only “segment_1” is accessed (0.5□0.8□(1-0.5)=0.2) plus the probability that goal is true if only “segment_2” is accessed ((1-0.5)□0.6□0.5=0.15). The use of probabilities provide the “relevance-based” ranking of the segments forming the video sequence, which corresponds to determining entry points to the MPEG-7 structure.

Assigning access probabilities to sub-segments makes it possible to differentiate between the sub-segments of a segment. For instance, in a news program, the first shot of a news item often contains a summary of that item. The content of that sub-segment could then be given higher priority than other segments in contributing to the content of the augmented segment, and the news program itself. The probabilities need however to be estimated either automatically (e.g. see [RLK+02] for a general methodology for deriving the estimates) or manually (e.g. the content producer) via, for instance, the instantiations of the “Probability Model” and “Analytic Model” DSs discussed above.

Data Propagation One requirement for a model for MPEG-7 retrieval is the capture the propagation of MPEG-7 descriptions. Propagation is expressed in POOL via rules. For example, the rule

```
S.title(T) :- segment(X) & X[segment(S)] & X.title(T)
```

expresses that the title T is assigned to each segment S if S is a segment within the context of segment X and T is the title of X. In this way, we can model the decomposition of segments and the propagation of attributes downwards in the hierarchy [CMF96]. The above rule defined the title relationship in the root context, whereas the rule

```
X[S.title(T)] :- segment(X) & X[segment(S)] & X.title(T)
```

---

[6] The semantics of the probability computation is beyond the scope of this paper, and readers should refer to [Röl99].

assigns the title to segments in the context of a decomposed segment X only. Some initial investigation can be found in [Zog02].

5.3.2. From POOL to PRA Representation

For execution, POOL programs are translated into PRA (probabilistic relational algebra) programs. The translation of POOL into PRA follows the so-called object-relational approach; PRA programs consist of probabilistic relations that model aggregation, classification, and attributes as well as the terms and structure. The relations necessary for modelling MPEG-7 descriptions include:

- *term*: represents the terms occurring in the MPEG-7 descriptions.
- *attribute*: represents the relationships between MPEG-7 elements.
- *acc*: represents the structure of MPEG-7 descriptions.

As an example, Figure 10 shows an extract of the PRA representation of the MPEG-7 data of Figure 8, based on the POOL representation of Figure 9. The *term* relation models the occurrence of a term in a document; a high weight (probability) corresponds to a high term frequency. The *acc* relation reflects the hierarchical structure of the documents. Here, "segment_1" is a sub-context of "audiovisualsegment_1". The higher the probability, the more impact has the content of the sub-context in describing the content of the super-context. In our example, no differentiation is made between the sub-segments, so the impact is full (the probability is 1.0). Relations reflecting spatial and temporal relationships between segments could also be used. The same criterion can be used in quantifying the impact of segments with respect to the content of spatially and temporally related segments. The *attribute* relation models relationships between elements. The last parameter of the attribute relation gives the context in which the relationship holds; for instance "audiovisualsegment_1" and "web" (the up-most context, i.e. the web).

```
1.0 attribute(title,audiovisualsegment_1,"Spain vs. Sweden", web).
% The attribute "title" of audiovisualsegment_1 having value
% "Spain vs Sweden" in the context of the database web
1.0 attribute(genre,audiovisualsegment_1,"Sports", web).
1.0 attribute(language,audiovisualsegment_1,"English", web).

1.0 term(soccer, audiovisualsegment_1).
1.0 term(game, audiovisualsegment_1).
0.8 term(spain, audiovisualsegment_1).
0.8 term(sweden, audiovisualsegment_1).
% The term tuples represent the "word" content of audiovisualsegment_1.

1.0 acc(audiovisualsegment_1, segment_1). % representation of the logical structure

1.0 attribute(medialocator,segment_1,"(?)",audiovisualsegment_1).
1.0 term(introduction,segment_1)
```

Figure 10: Example of the (simplified) PRA representation of the MPEG-7 Description of Figure 9

One can see that PRA programs are "assembler-like". The assembler nature of PRA programs was the motivation for defining POOL, thus having a more abstract and object-oriented view of the data than that provided by PRA.

5.3.3. The Retrieval Function

The retrieval function is performed through a probabilistic interpretation of standard relational algebra operations (e.g., UNION, JOIN, PROJECT, etc), where the relations are those obtained from the translation from POOL to PRA (e.g. see Figure 10). The retrieval process is implemented through PRA programs.

As described in the previous section, probability values are attached to tuples (e.g. *term* tuples, *acc* tuples, *attribute* tuples) and capture the uncertainty of the representation. The retrieval process accesses these tuples, and the probabilities are combined to infer the ranking. For instance, in the

PROJECT operation, when independence is assumed, the probabilities of duplicate tuples are added. The complete details of the calculation of the probabilities are beyond the scope of this paper, and interested readers should refer to [FR97]. In this section, we describe the retrieval process through examples of PRA retrieval functions.

A simple retrieval function that considers only the terms present in segments and queries would be implemented as follows ($n refers to the columns of the relations). For instance, a query about "goal" leads to the execution of the following PRA program[7]:

```
qterm(goal)
segment_index = term                    % renaming of relation for later use
retrieved_segments = PROJECT[$3](JOIN[$1=$1](qterm,segment_index))
```

The *qterm* relation represents the actual query (the terms composing the query, here the term "goal"). The first equation (*segment_index*) computes the segment index. The second equation matches (JOIN) the query terms with the document terms, and then returns (PROJECT) the matched segments. Applying this PRA program to our example, the two contexts "audiovisual-segment_1" and "segment_1" are retrieved. The retrieval status value (RSV) of "segment_1", which is used to rank segments in the retrieval result is computed as follows[8]:

```
RSV(segment_1) =  qterm(goal) □ segment_index(goal, segment_1)
```

More sophisticated indexing schemes can be used. For instance, inverse document frequency can be modelled by a relation *termspace*:

```
0.6 termspace(goal)
0.2 termspace(game)
…
```

This relation corresponds to the probabilistic interpretation of the inverse document (here segment) frequency of terms. The retrieval function (the segment indexing equation) is then expressed as follows:

```
segment_index = JOIN[$1,$1](term,termspace)
```

User preferences with respect to topicality can be incorporated in a similar way. For instance, suppose that a user has a preference for goals, and in particular those being scored by the French player Zidane. These preferences can be modelled by the relation *userspace* as follows:

```
0.7 userspace(goal)
1.0 userspace(zidane)
```

An additional equation would be inserted before the *retrieve_segments* equation joining the *segment_index* relation with the *userspace* relation. This would retrieve at higher rank any segments showing goals scored by Zidane, then segments showing goals scored by other players.

With the *acc* relation, the representation of super-contexts (e.g. "audiovisualsegment_1") is specified in terms of the representation of its sub-contexts (e.g. "segment_1"). The content of the super-context is augmented by that of its sub-contexts. This is modelled by the following PRA program:

```
term_augmented = PROJECT[$1,$3](JOIN[$2=$2](term,acc))
```

7 In POOL, such query was formulated as "?-D[goal]", see Section 5.3.1.

8 The computation requires the specification of the so-called disjointness key of the termspace relation. The clause _disjointness_key(termspace, "") tells HySpirit about the disjointness of the termspace tuples. Details can be found in [FR97] and [FR98].

Applied to our example, augmentation produces “term_augmented(goal, audiovisualsegment_1)”, i.e. we have augmented the description of “audiovisualsegment_1” by that of “segment_1”. The probability of the augmented term in the context “audiovisualsegment_1” is:

P(term_augmented(goal, audiovisualsegment_1))
= P(term(goal,segment_1)) × P(acc(audiovisualsegment_1,segment_1))

The *term* probability is multiplied with the *acc* probability, resulting in a smaller or at most equal probability in the super-contexts. The probability of an *augmented term* should be smaller since the super-contexts are larger contexts than the sub-contexts, and the probability should be normalised with respect to the document size and the *acc* value.

Retrieval is now performed with the *term_augmented* relation, yielding the super-contexts. In our example, “audiovisualsegment_1” and “segment_1” are retrieved. The *acc* relation allows the retrieval of entry points instead of components of fixed granularity levels.

We finish with an example of a combined content-based and factual query. The query on videos about “goal” or produced by the “BBC” would be expressed as the following PRA program (a simple indexing schema is used):

```
qterm(goal)
segment_index = term
retrieved_segments_1 = PROJECT[$3](JOIN[$1=$1](qterm,segment_index))
retrieved_segments_2 = PROJECT[$2](SELECT[creator =~ /BBC/](attribute))
retrieved_segments = UNION(retrieved_segments_1,retrieved_segments_2)
```

The first three expressions of the retrieval function have already been explained. The fourth expression (the third equation) retrieves all segments with creator similar to “BBC”[9]. The last expression (fourth equation) performs a probabilistic union over the two query criteria.

6. Conclusions and future work

A common distinction made between successive generations of web content is that the first generation is about hand-coded HTML pages, and the second generation is characterized by HTML pages generated on demand (i.e. the deep web). The third generation is envisaged to make use of rich and sophisticated mark-up languages to describe the semantic content of the document in a way that enables computer programmes (e.g. search engines) to process it for search purpose. The great vision underlying the third generation of web content is commonly referred to as the semantic web, which aims at enhancing the functionality of the current web to bring “meaning” to the content of web pages. Reaching this aim is a challenge for text content, and the more so for multimedia (non-text) content. Describing the semantic content of multimedia content is the aim of the MPEG-7 standard. Using MPEG-7 can considerably improve the access to multimedia data. However, simply annotating multimedia content with MPEG-7, whether manually or automatically, is useless if there are no means to exploit the associated MPEG-7 descriptions for the effective access to multimedia content.

A first requirement is to make the multimedia data searchable according to specific application contexts and user needs. For this, we need an abstract model that reflects the user’s retrieval needs and behaviours. To fulfil this, we propose the *Semantic Views Query Language* (SVQL). *SVQL* provides a high-level query language that allows users to express their retrieval requirements based on the Semantic Views Model in an abstract and concise style. SVQL queries can retrieve multimedia content described with the MPEG-7 standard. As was shown via a few examples, the important advantage of this language is that the users can retrieve multimedia content by focusing on their professional requirement and ignoring the frustrating details of MPEG-7 or the Semantic Views Model implementation issues. Moreover, the definition of SVQL as a specialization of XQuery has different advantages. Firstly, the similarity of SVQL syntax to the XQuery syntax is

[9] PRA provides regular expression matching.

an advantage for the users who already know the XQuery syntax (or the similar ones, such as SQL and OQL). Secondly, the implementation of the SVQL based on XQuery as shown in Section 4.5 is quite straightforward. Finally, if it is required, SVQL can be mixed with XQuery to provide more sophisticated query possibilities.

A second requirement is the development of a *Retrieval Model* that given a query, whether expressed in natural language or using sophisticated but user- or application- adaptable query languages such as SVQL, return a ranked list of relevant multimedia segments back to the user. One important characteristic of MPEG-7 descriptions is that they display a *structure*; that is they are composed of elements describing parts of the multimedia segment as well as the entire multimedia segment. By exploiting the structural characteristic of MPEG-7 descriptions, parts of as well as the whole multimedia content can be searched, thus allowing users to precisely access the data of interest to them. Since describing content involves uncertainty, returned segments should be ranked according to how they satisfy a query based on this uncertain description. In addition, due to the increasingly high amount of relevant information (i.e. for broad queries), ranking allows displaying highly relevant documents before less highly ones. This paper described in details the development and implementation of a retrieval model for MPEG-7 annotated multimedia content. The model was based on the HySpirit platform, which enables, among others, the uniform representation of content, fact and structure, all primary characteristics of MPEG-7 descriptions.

The SVQL query language and the retrieval model described in this paper (in Sections 4 and 5, respectively) can be seen as complementary work. As a matter of fact, their development was done separately and in parallel. Our next step is to combine the two approaches, so that to obtain integrated means for a relevance-based access multimedia content according to specific application contexts and user needs, and to evaluate the combined model on realistic scenarios (in terms of data set and user needs). It will be relatively straightforward for the HySpirit platform to take a query expressed using SVQL and to translate it into POOL. An investigation onto what constitutes content querying and factual querying will have to be carried out. An other issue is that of target elements as with SVQL it is possible to specify which type of element should be returned (through the RETURN clause in Figure 4). Although not described in this paper, HySpirit can process target elements and has been used in the context of MPEG-7 (see [SAM00]) and XML retrieval (see [LR04]) to deal with content-only queries (i.e. no constraint with respect to the structure, as discussed in this paper), and content-and-structure queries (i.e. there are constraints with respect to the structure).

After combining the two approaches, it will be necessary to integrate the combined approach (query language + retrieval model) with the semantic web so that to offer seamless access to both text and non-textual web content. There are two issues here. The first one is to integrate our work into existing tools, platforms, and standards (e.g. OWL [OWL]) that are currently being used and tested in order to deliver a true semantic web. The second issue is that the integration should be done taking into consideration user behaviours and expectations in the context of a semantic web that contains multimedia content as user behaviours may be different when dealing with multimedia content to that when confronted with text only.

Although the work presented in this paper still requires further development and investigation before it can be truly said to provide effective access to multimedia content for the semantic web, we believe that the two complementary approaches and their future integration correspond to a step forward towards this ultimate aim.

7. References

[ABB01] J. Assfalg, M. Bertini and A. Del Bimbo. Information Extraction from Video Streams for Retrieval by Content, *23rd European Colloquium on Information Retrieval Research*, Darmstadt, Germany, 2001.

[ACC+97] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte and J. Simeon. Querying documents in object databases, *International Journal on Digital Libraries*, 1:1-9, 1997.

[ABL95] G. Ahanger, D. Benson, and T.D.C. Little. Video query formulation, *Storage and Retrieval for Images and Video Databases II*, San Jose, 1995.
[Ber01] T. Berners-Lee. Conceptual Graphs and the Semantic Web,
http://www.w3.org/DesignIssues/CG.html, 2001.
[BR99] R Baeza-Yates and B Ribeiro-Neto. *Modern Information Retrieval*, Addison Wesley, 1999.
[BZC+01] A.B. Benitez, D. Zhong, S-F. Chang, and J.R. Smith. MPEG-7 MDS Content Description Tools and Applications, *International Conference on Computer Analysis of Images and Patterns (CAIP-2001)*, Warsaw, Poland, 2001.
[Cal94] J. Callan. Passage-Level Evidence in Document Retrieval. *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp 302-310, Dublin, Ireland, 1994.
[CCM+98] S.-F.Chang, W. Chen, H.-J. Meng, H. Sundaram and D. Zhong. A fully automated content based video search engine supporting spatio-temporal queries. *IEEE Transaction on Circuits and Systems for Video Technology*, 8:602-615, 1998.
[Chi97] Y. Chiaramella. Browsing and querying: two complementary approaches for multimedia information retrieval, *Hypermedia - Information Retrieval – Multimedia*, pp 9-26, Dortmund, Germany, 1997.
[CMF96] Y. Chiaramella, P. Mulhem and .F Fourel. A model for multimedia information retrieval. Technical Report Fermi ESPRIT BRA 8134, University of Glasgow, 1996.
[CR01] P. Cotton and J. Robie. *The W3C XML Query Working Group*, 2001.
[FA01] N. Fatemi and O. Abou-Khaled. Indexing and Retrieval of TV News Programs based on MPEG-7, *IEEE International Conference on Consumer Electronics (ICCE2001)*, Los Angeles, CA, June 2001.
[Fat03] N. Fatemi. *A Semantic Views Model for Multimedia Indexing and Retrieval*, PhD thesis, N° 2738, Swiss Federal Institute of technology of Lausanne, Lausanne, 2003.
[FH94] R. Fagin and J. Halpern. *Reasoning about Knowledge*, MIT Press, Cambridge, Massachusetts, 1995.
[Fli95] M. Flinker et al., Query by Image and Video Content: The QBIC system, *IEEE Computer*, 28:23-32, 1995.
[Fri88] M. Frisse. Searching for information in a hypertext medical handbook, *Communications of the ACM*, 31(7):880-886, 1988.
[FR97] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 14(1):32-66, 1997.
[FR98] N. Fuhr and T. Rölleke. HySpirit - A probabilistic inference engine for hypermedia retrieval in large databases, *International Conference on Extending Database Technology (EDBT),* pp 24-38, Valencia, Spain, 1998.
[GB93] S. Gibbs and C. Breitender. Audio/video database: an object-oriented approach, *International Conference on Data Engineering*, 1993.
[GL02] A. Graves and M. Lalmas. Video retrieval using an MPEG-7 based inference network, *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp 339-346, Tampere, Finland, 2002.
[GBC+99] D. Gibbon, A. Basso, R. Civanlar, Q. Huang, E. Levin and R. Pieraccini. Browsing and Retrieval of Full Broadcast-Quality Video, *10th International Workshop on Packet Video New York*, USA, 1999.
[HGH+97] A. Hampapur, A. Gupta, B. Horowitz, C-F. Shu, C. Fuller, J. Bach, M. Gorkani and R. Jain. Virage video engine, *SPIES vol 3022, Storage and Retrieval for Image and Video Databases*, pp. 188-198, 1997.
[HS96] E. Hwang and V.S. Subrahmanian, Querying video libraries, *Journal of Visual Communications and Image Representation*, 7(1), 1996
[Hau95] A.G. Hautmann. Speech Recognition in the Informedia Digital Video Library: Uses and Limitations, *IEEE International Conference on Tools with Artificial Intelligence*, Washington DC, USA, 1995.
[HI98] J. Hunter and R. Iannella. The application of Metadata Standards to Video Indexing, *Second European Conference on Research and Advanced Technology for Digital Libraries*, Crete, Greece, 1998.
[HK99] O. Hori and T. Kaneko. Results of Spatio-Temporal Region DS Core/Validation Experiment. *ISO/IEC JTC1/SC29/WG11/MPEG99/M5414*, Maui, December 1999.
[HLM+00] P.G.T. Healey, M. Lalmas, E. Moutogianni, Y. Paker and A. Pearmain. Integrating internet and digital video broadcast data. *4th world Multiconference on Systemics, Cybernetics and Informatics (SCI 2000), Information Systems*, Vol I, pp 624-627, Orlando, Florida, U.S.A, 2000.
[HM92] J. Halpern and Y. Moses. A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief, *Artificial Intelligence*, 54:319-379, 1992.
[HML+01] P.G.T. Healey, E. Moutogianni, M. Lalmas, Y. Paker, D. Papworth and A. Pearmain. Requirements for Broadcast and Internet Integration, *Media Future*, Florence, Italy, 2001.
[Hun99a] J Hunter. MPEG-7 Behind the Scenes, *D-Lib Magazine*, 5(9), 1999.

[Hun99b] J. Hunter and L. Armstrong. A comparison of schemas for video metadata representation, *8th WWW Conference*, Toronto, 1999.
[ILR01] D. Ileperuma, M. Lalmas and. T Rölleke. MPEG-7 for an integrated access to broadcast and Internet data, *Media Future*, Florence, Italy, 2001.
[ISOa] ISO MPEG-7. Text of ISO/IEC FDIS 15938-1 Information Technology - Multimedia Content Description Interface - *Part 1 Systems*, ISO/IEC JTC 1/SC 29/WG 11 N4285, 2001.
[ISOb] ISO MPEG-7. Text of ISO/IEC FDIS 15938-2 Information Technology - Multimedia Content Description Interface - *Part 2 Description Definition Language*, ISO/IEC JTC 1/SC 29/WG 11 N4288, 2001
[ISOc] ISO MPEG-7. Text of ISO/IEC FDIS 15938-3 Information Technology - Multimedia Content Description Interface - *Part 3 Visual*, ISO/IEC JTC 1/SC 29/WG 11 N4358, 2001.
[ISOd] ISO MPEG-7. Text of ISO/IEC FDIS 15938-4 Information Technology - Multimedia Content Description Interface - *Part 4 Audio*, ISO/IEC JTC 1/SC 29/WG 11 N4224, 2001.
[ISOe] ISO MPEG-7. Text of ISO/IEC FDIS15938-5 Information Technology - Multimedia Content Description Interface - *Part 5 Multimedia Description Schemes*, ISO/IEC JTC 1/SC 29/WG 11 N4242, 2001.
[Jör99] C.L. Jörgensen. Theory and Practice in the Organization of Images and Other Visuo-Spatial Data for Retrieval, *Bulletin of the American Society for Information Science*, 25(6):13, 1999.
[KKK03] J.H.Kang, C.-S. Kim and E.-J. Ko, An XQuery engine for digital library systems, *(JCDL) Joint Conference on Digital Libraries*, Houston, Texas, 2003.
[KLW95] M. Kifer, G. Lausen and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages, *Journal of the Associations for Computing Machinery*, 42(4):741-843, 1995.
[LM00] M. Lalmas and E. Moutogianni. A Dempster-Shafer indexing for the focussed retrieval of hierarchically structured documents: Implementation and experiments on a web museum collection, *6th RIAO Conference, Content-Based Multimedia Information Access*, Paris, France, 2000.
[LR98] M. Lalmas and I. Ruthven. Representing and retrieving structured documents with Dempster-Shafer's theory of evidence: Modelling and evaluation, *Journal of Documentation*, 54(5): 529-565. 1998.
[LR04] M. Lalmas and T. Rölleke. Modelling vague content and structure querying in XML retrieval with a probabilistic object-relational framework, *FQAS, 6th International Conference On Flexible Query Answering Systems*, Lyon, France, 2004.
[LS99] J. Llach and P. Salembier. Analysis of video sequences: Table of contents and index creation, *VLDB International Workshop on Very Low Bit-rate Video Coding,* pp 52-56, Kyoto, Japan, 1999.
[LS02] P. Liu and L.H. Hsu. Queries of digital content descriptions in MPEG-7 and MPEG-21 XML documents, *XML Europe 2002*, Barcelona, Spain, 2002.
[LST+00] H. Lee, A.F. Smeaton, C. O'Toole, N. Murphy, S. Marlow and N.E. O'Connor. Recording, Analysing and Browsing System, *6th RIAO Conference, Content-Based Multimedia Information Access*, Paris, France, 2000.
[LYC98] Q. Li Y. Yang and W.K. Chung. *CAROL:* Towards a declarative video data retrieval language, *Proceedings of SPIE's Photonics China: Electronic Imaging and Multimedia Systems*, pp 69-78, Beijing, China, 1998.
[MJK+98] S. Myaeng, D.H. Jang, M.S. Kim and Z.C. Zhoo. A flexible model for retrieval of SGML documents, *ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp 138-145, Melbourne, Australia, 1998.
[MPEG-7] MPEG-7 Home Page: http://ipsi.fraunhofer.de/delite/Projects/MPEG7/
[MPE99] MPEG Requirements Group. *MPEG-7 Requirements Document v.16*, ISO/IEC JTC 1/SC 29/WG 11 N4510 , MPEG Pattaya Meeting, 2001.
[MPE00] MPEG Requirements Group. *Overview of the MPEG-7 Standard (v.5.0)*, ISO/IEC JTC 1/SC 29/WG 11 N4031, MPEG Singapure Meeting, 2001.
[MPE01] MPEG Requirements Group. *MPEG-7 Applications Document v.10*, ISO/IEC JTC 1/SC 29/WG 11 N3934, MPEG Pisa Meeting, 2001.
[OO99] B.C. O'Connor and M.K. O'Connor. Categories, Photographs & Predicaments: Exploratory Research on Representing Pictures for Access, *Bulletin of the American Society for Information Science* 25(6):17-20, 1999.
[OT93] E. Oomoto. and K. Tanaka. OVID: Design and implementation of a video-object database system, *IEEE Transaction on Knowledge and data Engineering*, 5(4):629-643, 1993.
[OWL] Web Ontology Language (OWL) home page: http://www.w3.org/2004/OWL/.
[PLM+01] A. Pearmain, M. Lalmas, E. Moutogianni, D. Papworth, P. Healy and T. Rölleke. Using MPEG-7 at the Consumer Terminal in Broadcasting, *EURASIP (European Association for Signal, Speech and Image Processing) Journal on Applied Signal Processing*, 2002.
[RDF] RDF Vocabulary Description Language 1.0: RDF SchmaW3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-schema/

[Rij86] C.J. van Risjbergen. A Non-Classical Logic for Information Retrieval, *The Computer Journal*, 29(6):481-485, 1986.

[RLK01] T. Rölleke, R. Lübeck and G. Kazai. The HySpirit Retrieval Platform. *ACM SIGIR Conference on Research and Development in Information Retrieval,* pp 454, Demonstration, New Orleans, USA, 2001.

[RLK+02] T. Rölleke, M. Lalmas, G. Kazai, I. Ruthven and S Quicker. The Accessibility Dimension for Structured Document Retrieval, *24th European Conference on Information Retrieval Research*, pp 284-302, Glasgow, 2002.

[RLS98] J. Robie, J. Lapp and D. Schach. XML Query Language (XQL), *Query Language Workshop, W3C*, December 1998, http://www.w3.org/TandS/QL/QL98/.

[Röl99] T. Roelleke. *POOL: Probabilistic Object-Oriented Logical Representation and Retrieval of Complex Objects - A Model for Hypermedia Retrieval*, PhD thesis, University of Dortmund, Germany, 1999.

[SAB93] G. Salton, J. Allan and C. Buckley. Approaches to passage retrieval in full text information systems, *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 49-58, Pittsburgh, 1993.

[SAM00] *System for Advanced Multimedia Broadcast and IT Services*, IST-2000-12605, http://www.irt.de/sambits/, 2000.

[SC00] B. Smyth and P. Cotter. A personalised Television Listings Service*, Communication of the ACM*, 43(8):107-111, 2000.

[Sha94] S. Shatford-Layne. Some issues in the indexing of images, *Journal of the American Society of Information Science*, 45(8):583-588, 1994.

[SLG00] P. Salembier, J. Llach and L. Garrido. Visual segment tree creation for MPEG-7 description schemes, *International Conference on Multimedia and Expo*, *ICME'2000*, Vol 2, pp 907-910, New York City, USA, 2000.

[Sme00] A.F. Smeaton. Indexing, Browsing and Searching of Digital Video and Digital Audio Information, Tutorial Notes, *3rd European Summer School in Information Retrieval* (ESSIR), Varenna, Lago di Como, Italy, 2000 (also in Lecture Notes on Information Retrieval, pp 93-110, Springer-Verlag, 2001)

[Sow84] J.F. Sowa. *Conceptual structures: Information processing in mind and machines*, Addison-Wesley Publishing Company, 1984.

[TC02] D. Tjondronegoro, D. and Y.-P.P Chen. Content-based indexing and retrieval using MPEG-7 and XQuery in video data management systems, *World Wide Web Journal*, 5(3):207-227, 2002.

[Wil94] R Wilkinson. Effective Retrieval of Structured Documents. *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp 311-317, Dublin, Ireland, 1994.

[XML] XML Schema, Parts 0, 1, and 2. http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/, 2001.

[XQuery] XQuery 1.0: An XML Query Language, WTC Working Draft, http://www.w3.org/TR/2002/WD-xquery-20021115/, 2002.

[Zog02] E. Zografou. *Data propagation in MPEG-7*, MSc in Advanced Methods of Computer Science. Queen Mary University of London, 2002.

[ZTS+95] H. Zhang, S.Y. Tan, S.W. Smoliar and G. Yihong. Automatic parsing and indexing of news video, *Multimedia Systems*, 2:156-166, 1995.