MPEG-7 and Multimedia Database Systems

Harald Kosch
Institute of Information Technology
University Klagenfurt, Austria
harald.kosch@itec.uni-klu.ac.at

ABSTRACT

The Multimedia Description Standard MPEG-7 is an International Standard since February 2002. It defines a huge set of description classes for multimedia content, for its creation and its communication. This article investigates what MPEG-7 means to Multimedia Database Systems (MMDBSs) and vice versa. We argue that MPEG-7 has to be considered complementary to, rather than competing with, data models employed in MMDBSs. Finally we show by an example scenario how these technologies can reasonably complement one another.

1. INTRODUCTION

Multimedia Database Systems (MMDBSs) organize and structure multimedia information for content retrieval [1]. In this context, multimedia data modeling is one of the key issues. The term is used to determine which information should be stored in a database and to reflect the relationships between the data items. Many good models have been proposed in the past which reflect the needs of the database developer as well as that of the database user [2]. Most of these models have been implemented in extensible database systems and recently object-relational (OR) systems have been used. Using OR technology for this purpose is supported by the SQL Multimedia and Application Packages (SQL/MM) standardized in May 2001 by the ISO subcommittee SC32 Working Group (WG) 4. SQL/MM introduces object types and associated methods for full-text, spatial data, and images. Data-mining applications are also under consideration. For images, SQL/MM provides structured object types, notably *SI_StillImage*, and methods to store, manipulate and represent image data by content (e.g., color histogram). An introduction to all parts of SQL/MM can be found in Jim Melton and Andrew Eisenberg's overview, appeared in Sigmod Records 30(4), December 2001 [3].

In this context, the ISO subcommittee SC29, WG11, MPEG (Moving Picture Experts Group), published in February 2002 another standard called "Multimedia Content Description Interface" (in short 'MPEG-7') which provides a set of descriptors (D), i.e., quantitative measures of audio-visual features, and description scheme (DS), i.e., structures of descriptors and their relationship. An overview is provided in [4] (see also the 'further reading' paragraph at the end of this paper). Audio-Visual (A/V) material that has MPEG-7 data associated with it can be indexed and searched for. This 'material' may include: images, graphics, 3D models, audio, video, and information about how these elements are combined in a multimedia presentation ('scenarios', composition information). It is currently the most complete description standard of multimedia.

In this scope, we like to raise the following question:
Is MPEG-7 a data model for MMDBSs?

This article investigates the answer to this question and will show that MPEG-7 is not a competing technology to data modeling in MMDBSs and to SQL/MM, but may provide interoperability to an MMDBS in its distributed environment.

2. MPEG-7: THE MULTIMEDIA DESCRIPTION STANDARD

MPEG-7 [4, 5] specifies a standard set of DSs and Ds that can be used to describe various types of multimedia information. The standard is organized in eight parts: systems issues (reference framework for storage and transport of documents), Data Definition Language, low-level visual description tools (Ds for image and video content), low-level audio description tools, *Multimedia Description Schemes* (*MDS*, high-level DSs and Ds independent of the media types), reference software, conformance issues (guideline for testing the validity of the MPEG-7 decoder), and finally extraction and use (guidelines and examples of the extraction and use of descriptions). An *MPEG-7* instance must be structured according to the rules of the *Data Definition Language* (DDL). The DDL is based on *XML Schema* and extends it by new data types, like histograms.

For an example of usage of MPEG-7, we will concentrate on the *VideoSegment DS*. This useful description scheme describes temporal intervals or segments of video data which can correspond to an arbitrary sequence of frames, a single frame, or even the full video sequence. It is part of the MDS [6] which model the semantic content of multimedia data.

How is an MPEG-7 description written?
An MPEG-7 description is written using the types defined in the DDL. The *VideoSegmentType* defines the structure of a *VideoSegment DS*. Let us consider the following extract:

```
<complexType name="VideoSegmentType">
 <complexContent> <extension base="mpeg7:SegmentType">
   <sequence>
    <element name="MediaTime"
       type="mpeg7:MediaTimeType" minOccurs="0"/>
     ...
    <choice minOccurs="0" maxOccurs="unbounded">
     <element name="SpatialDecomposition"
         type="mpeg7:VideoSegmentSpatialDecompositionType"/>
     <element name="TemporalDecomposition"
         type="mpeg7:VideoSegmentTemporalDecompositionType"/>
     <element name="SpatioTemporalDecomposition"
       ...
    </choice>
     ...
   </sequence> </extension> </complexContent> </complexType>
```

At first, *VideoSegmentType* extends the *SegmentType*, i.e., *VideoSegmentType* inherits all characteristics of the *SegmentType*. For instance, it inherits the optional child element *TextAnnotation* from the *SegmentType*. A VideoSegment is then a *sequence* of child element declarations. The *sequence* group element forces that the child elements of *VideoSegmentType* appear in the instance document exactly in the order they are declared in the DDL. The first one is *MediaTime* which describes the temporal localization of the video segment. This element is optional, i.e., the value of the attribute *minOccurs* is "0". It is then followed by a list of alternative child elements, indicated by a *choice* group element. *choice* allows exactly one of its children to appear in an instance, i.e., either a *VideoSegmentSpatialDecompositionType*, or a *VideoSegmentTemporalDecompositionType*.

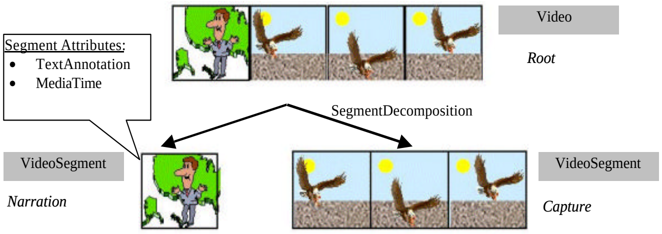

Figure 1: ***Example segment tree.***

Let us now introduce an instance of the *VideoSegmentType* describing the segment tree hierarchy shown in Fig. 1. In this example, the video *Root* represents a video sequence with two scenes, *Narration* and *Capture*. The video `RootV` is decomposed into two video segments, `NarrationVS` and `CaptureVS`, which represent the Narration and Capture scenes. The segment decomposition has neither gaps nor overlaps. All the video segments are temporally connected. Text annotations are associated with the video segments using the *TextAnnnotation D*.

We specify the example scene in an MPEG-7 *complete description*, which usually contains a "semantically-complete" description. The complete video is encoded with a *VideoType*, whereas the segments *Narration* and *Capture* base on the *VideoSegmentType* described above.

```
<Mpeg7>
<Description xsi:type="ContentEntityType">
 <MultimediaContent xsi:type="VideoType">
  <Video id="RootV">
    <TextAnnotation> <FreeTextAnnotation>
      RootV </FreeTextAnnotation> </TextAnnotation>
    <!--  MediaTime description:
      It follows the temporal component of the RootV Video:
      start = 0:0:00, duration = 1 minutes and 30 seconds -->
    <MediaTime>
     <MediaTimePoint>T00:00:00</MediaTimePoint>
     <MediaDuration>PT1M30S</MediaDuration>
    </MediaTime>
    <!--  TemporalDecomposition description:
    RootV is decomposed into two segments with no gaps:
    * NarrationVS:start=0:0:00,duration =0minutes,15seconds
    * CaptureVS:start=0:0:15,duration =1minute,15seconds -->
    <TemporalDecomposition gap="false" overlap="false">
     <VideoSegment id="NarrationVS">
      <TextAnnotation>
        <FreeTextAnnotation> NarrationVS </FreeTextAnnotation>
      </TextAnnotation>
      <MediaTime>
        <MediaTimePoint>T00:00:00</MediaTimePoint>
        <MediaDuration>PT0M15S</MediaDuration>
      </MediaTime>
     </VideoSegment>
     <VideoSegment id="CaptureVS">
      <TextAnnotation>
        <FreeTextAnnotation> CaptureVS </FreeTextAnnotation>
      </TextAnnotation>
      <MediaTime>
        <MediaTimePoint>T00:00:15</MediaTimePoint>
        <MediaDuration>PT1M15S</MediaDuration>
      </MediaTime>
     </VideoSegment>
    </TemporalDecomposition>
   </Video>
  </MultimediaContent> </Description>
</Mpeg7>
```

3. MMDBS DATA MODELS AND SQL/MM

The MMDBS data model deals with the issue of representing the content of all multimedia objects present in the database, that is, designing the high- and low-level abstraction model of the raw media objects and its correlations to facilitate various operations. These operations may include media objects selection, insertion, editing, indexing, browsing, querying, and retrieval. The data model thereby relies on the feature extraction vectors and its respective representations, obtained during the indexing process.

In recent years many data models have been proposed for MMDBSs. In the following we describe four representative models [7, 8, 9, 10].

The *algebraic video* data model presented in [7] has been one of the first and quite complete data models and influenced many later models. This model distinguishes between physical video streams and logical video segments and allows users to compose concurrent video presentations, supported by a *video algebra*. The video algebra is introduced as a mechanism for combining and expressing temporal relations between video expressions, for defining output characteristics of video expressions concerning the presentation of video and audio streams, as well as, for associating descriptive information with these expressions. However, it is difficult to add automatic methods for segmenting video streams.

In [8], a video data model called *Logical Hypervideo Data Model* (LHVDM) is presented. This model is capable of multilevel video abstractions (representations of video objects for which users are interested in), called *hot objects*, as well as their semantic associations with other logical video abstractions, including other hot objects. In addition to this, the proposed model supports semantic associations called *video hyperlinks* and video data dealing with such properties, called *hyper video*. Furthermore, a framework for defining, evaluating and processing video queries using temporal, spatial and semantic constraints is defined.

DISIMA [9], developed at the University of Alberta, is an image database system and proposes a model for both image and spatial applications. The DISIMA model allows the user to assign different semantics to an image component and an image representation can be changed without any effect on applications using it. An associated query language (MOQL), extending OQL, allows spatio-temporal querying as well as the definition of a presentation specification.

Finally, in a recent work we proposed a generic indexing model, *VIDEX* [10], implemented in the multimedia information system *SMOOTH* [11], which describes a narrative world as a set of semantic classes and semantic relations (including spatial and temporal relationship) among these classes and media segments. The core of the *VIDEX* model defines base classes for an indexing system, while application specific classes are added by declaring subclasses (content classes) of the base classes. Furthermore, our model introduces concepts for detailed structuring of video streams and for relating the instances of the semantic classes to the media segments. This approach combines advantages of the related methods introduced in [7, 8] and extends them by the introduction of 1) means for structuring video streams and 2) genericity in the indexing process.

The above mentioned conceptual multimedia data models are implemented as database schemas, benefiting from standard DBMS services. However, they use different database

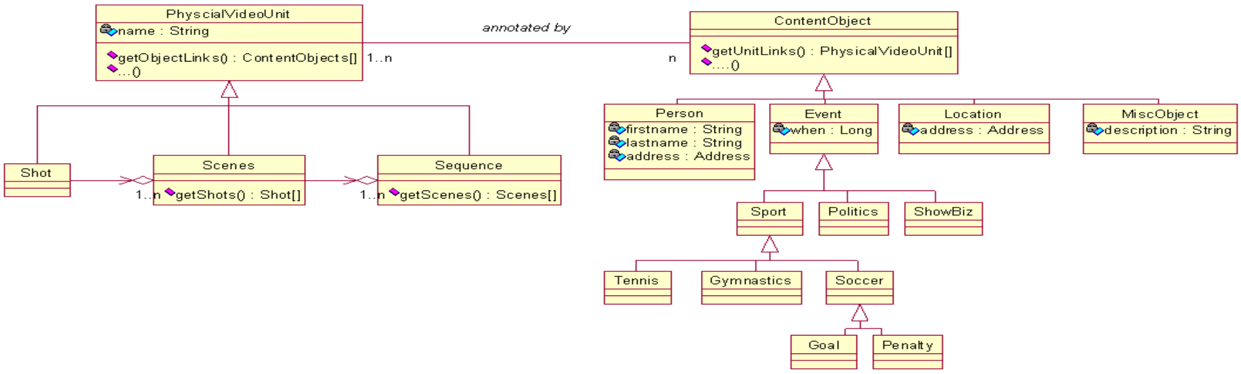

Figure 2: ***VIDEX model in UML (the inheritance hierarchy below* ContentObject *is not shown completely).***

extensions. For instance, DISIMA is implemented on top of the object-oriented ObjectStore and extends its type system to support MOQL queries. SMOOTH relies on the ORD* datatypes in Oracle to store and access multimedia data. In order to provide a higher portability for a type system which supports multimedia data, the SQL Multimedia and Application Packages (SQL/MM) standard has been developed. Please refer to [3] for an introduction and discussion of SQL/MM. It should be noted that no standard for querying video data is actually proposed. Although some features of SQL/MM Still Image apply to video too, other important features like motion as a video structuring aspect have to be modeled additionally.
Let us finally remark that different database products already provide data types and associated methods for images and videos, e.g., IBM Informix DataBlades extend the database so that images and videos can be stored and manipulated in a database (`http://www-4.ibm.com/software/data/informix/blades/index.html`).

4. MPEG-7 AND MMDBS

The following conclusions can be drawn from the discussion made in the previous Sections. The descriptive information contained in the MPEG-7 DS and Ds can be considered, to some extent, as a summary of the contents and principles introduced by the models mentioned above. However, the structural definition of the content differs, based on whether it is meant for storage in a database and further content querying or it is designed for information exchange.

This fundamental difference is revealed at several levels. The first one is in the overall organization of the content types. MPEG-7 tries to keep the hierarchy of content descriptions as flat as possible, in order to minimize the number of dereferencing before reaching a desired information. This stems from the simple requirement that some network components and the client may not have enough computing power or storage space to process complicated unfolding of descriptive information. In contrast to this, semantically rich MMDBS data models like VIDEX have complicated hierarchies in order to be as extensible as possible. The second one is in the way links among DSs are defined. For instance, the relation *SegmentSemanticBaseRelation* allows one to link the semantic DSs (e.g., *EventDS*) to the structural DSs (e.g., *VideoSegmentDS*) and vice versa. The *SegmentDS* provides links for quick navigation in the segment tree hierarchy which is useful for browsing the logical content during information exchange. However, the *SegmentDS* lacks concepts for DBMS querying. It provides no means to introduce different abstraction levels of video segmentation. Broadly used data models in MMDBSs provide at least two levels [12]: a *shot* level that consists of one or more consecutively generated and recorded frames representing a continuous action in time and space and a *scene* level which groups together shots which are related semantically in time and space. A shot may be recognized automatically by physical boundaries like camera breaks and editing points, whereas scene detection is in general a semi-automatic process and application domain dependent (e.g., Zhong et al. [13] propose scene detection for sports video). Such video structuring allows the user to specify the required abstraction level in the query.

The third one is in the coding scheme for XML-based MPEG-7 documents. MPEG-7 comes up with a Binary Format, the *BiM*. The BiM provides the functionality to delete and to modify 'access units' (the smallest unit to which timing information can be attributed) in the binary format. This is because some network components and the client may not have enough computing power as explained earlier.

Finally, the difference is revealed on how both approaches attribute importance to single descriptive elements. For example, MPEG-7 proposes a variety of tools (*MediaLocator DS*) to specify the "location" of a particular image, audio or video segment by referencing the media data. This is important for cross-referencing of meta-data and media-data. However, such an effort is not of so much interest for a multimedia data model in a DBMS, since in most cases a simple reference to the video data, e.g., as BFILE in Oracle 8i or other DBMSs, is sufficient for location.

Let us illustrate the difference with an example. In our video model VIDEX, we are able to build the following content classes for a News-on-Demand application, as shown in Fig. 2. Upon a content query we are able to quickly navigate in the schema through the direct access methods *getUnitLinks()* from *ContentObject* to *PhysicalVideoUnit* and inversely through *getObjectLink()*. The inheritance hierarchy below the class *Event*, gives us the possibility to quickly categorize events upon annotation and to answer fine-granular queries on events (e.g., give me all news items which show goals of the afternoon soccer games).

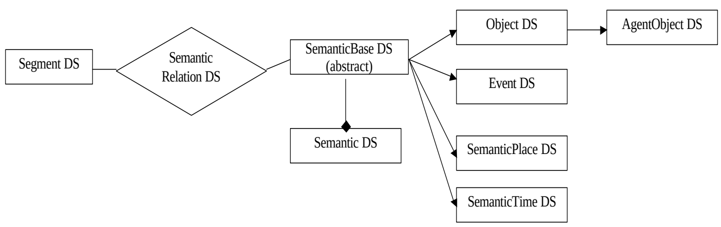

Figure 3: ***Extract of the type system for high-level video indexing in MPEG-7 (Extended-ER notation).***

MPEG-7 introduces the *SemanticDS* to describe a narrative world as a set of semantic entities and semantic relations among the semantic entities, and segments. A number of specialized *SemanticBaseDSs* are derived from the generic *SemanticBaseDS* which describe specific types of semantic entities, such as objects, agent objects, events, places, and time. Fig. 3 shows an extract of the type system for high-level video indexing in Extended-ER notation. The type hierarchy below *SemanticBaseDS* is rather flat; only one type, *AgentObjectDS*, extends another one: *ObjectDS*. This subtyping was introduced to model an object that could be a person, an organization, or a group of people. No more subtypes below *SemanticBaseDS* can be introduced for a particular application, such as News-on-Demand. Thus, a sport or a showbiz event has to be encoded as a value of a textual field (e.g., Label) in the *EventDS*.

Consequently, the above query, "give me all news items which show goals of the afternoon soccer games", must be answered through matching against the value of a textual field describing the event. In order to handle the problem of context meaning for text matching (once again in the spirit of being an information exchange standard), MPEG-7 introduces the concept of a *ControlledTermType*, by which one might control the value of a textual field using a classification scheme, e.g., a soccer expression can be controlled by a classification scheme built on the soccer terminology introduced in `http://www.decatursports.com/soccerterms.htm`. The MPEG-7 processor may be provided with a mechanism to validate the reference to a term in a classification scheme, i.e., it may perform a checking of a term during filtering or updating.

5. MPEG-7 & MMDBS MAY COMPLEMENT ONE ANOTHER

The previous Section concluded that MPEG-7 standardizes the information exchange of descriptive information. However, it is not suitable to serve as a multimedia data model for a database.

In spite of the difference in requirements of MPEG-7 and a MMDBS, they have the potential to work together to build a distributed multimedia system. This is an important issue, as most related works in MMDBSs simply ignore the existence of the media delivery and its related issues, like Quality of Service (QoS), etc.

In order to use MPEG-7 effectively in a distributed multimedia system, storage and query mechanisms for XML-based MPEG-7 documents have to used. One possibility is, for instance, to use an extensible DBMS for implementing the multimedia data model and to generate MPEG-7 documents for information exchange. Such an approach can be realized, for example, with the XML SQL Utility (XSU), a utility from Oracle (`http://technet.oracle.com`). Alternatively, one can use a so called *XML-DBMS* to store and query MPEG-7 documents directly. An XML-DBMS provides query interfaces and languages [14] with respect to the nature of XML documents. Example systems are IPSIS from the GMD Darmstadt (`http://xml.darmstadt.gmd.de/xql/`), dbXML (`http://www.dbxml.org`) which is an open source project. The broadly used query language is *XPATH*. The second version of XPATH is under development and shall simplify the manipulation of XML Schema-typed content. This is of obvious advantage for MPEG-7. For more information, refer to `http://www.w3.org/TR/xpath20req`.

Whether an XML-DBMS or an extensible ORDBMS should be used depends mainly on the application area. For instance, the M3box [15] project, currently under development by Siemens Corporate Technology,IC 2, in cooperation with our Institute, aims at the realization of an adaptive multimedia message box. One of its main components is a multimedia database for retrieving multimedia mails based on their structured content. The input to the database are MPEG-7 documents generated in the input devices (e.g., videophone, PDAs) and the output are again MPEG-7 documents to be browsed at the output devices. The system is performance-critical and, therefore, relies on an XML-DBMS, actually on dbXML, in order to avoid scheme translations of both the input and the output.

The provision of an MPEG-7 codec is another important contribution for integrating multimedia systems. The codec provides means to encode and decode MPEG-7 descriptions into and from the binary format (BiM). Therefore, it realizes conformity with respect to versions and extensions and ensures interoperability. In addition, the codec leaves enough freedom for the user to specify different ways of splitting a description into fragments to be packed into the same access units (these fragments are decoded at the same time unit). This means that the delivery and the use of the information at the destination may be controlled during the encoding process.

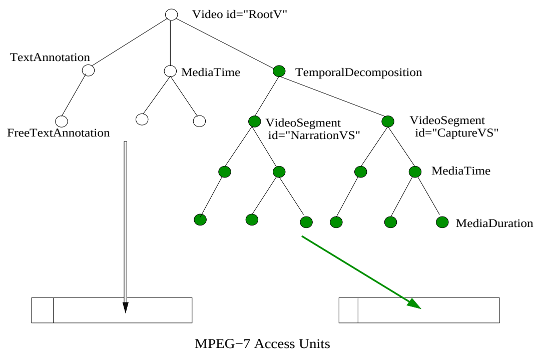

Figure 5: ***Example Coding for the video segment of Fig. 1.***

For instance, let us reconsider the scene described in Fig. 1 and assume that the scene is requested by a client. Then, the MPEG-7 description for the scene shall be sent together with the clip to the client. In this case, the database has to decide how to encode the different elements of the description. One useful approach is to encode, first, the global information of the complete clip (video `RootV`) and then the information of the temporal decomposition and the video segments `NarrationVS` and `CaptureVS`. Using two access units would lead to the coding scheme as depicted in Fig. 5. The document is here displayed as a tree, where nodes represent the elements. For a better readability, the name of some of the elements are shown.

Finally, MPEG-7 contains descriptors for quality variation of the media, i.e., they model the adaptation capability of the media. In the following Section, we will show an application which uses content and quality variation descriptors to effectively bridge different multimedia systems (MMDBS, video server and clients).

6. CASE STUDY OF A COOPERATION BETWEEN MPEG-7 AND A MMDBS

The case study for a reasonable cooperation of MPEG-7 and a MMDBS concerns the adaptation of delivered A/V (Audio/Video) streams in network components with

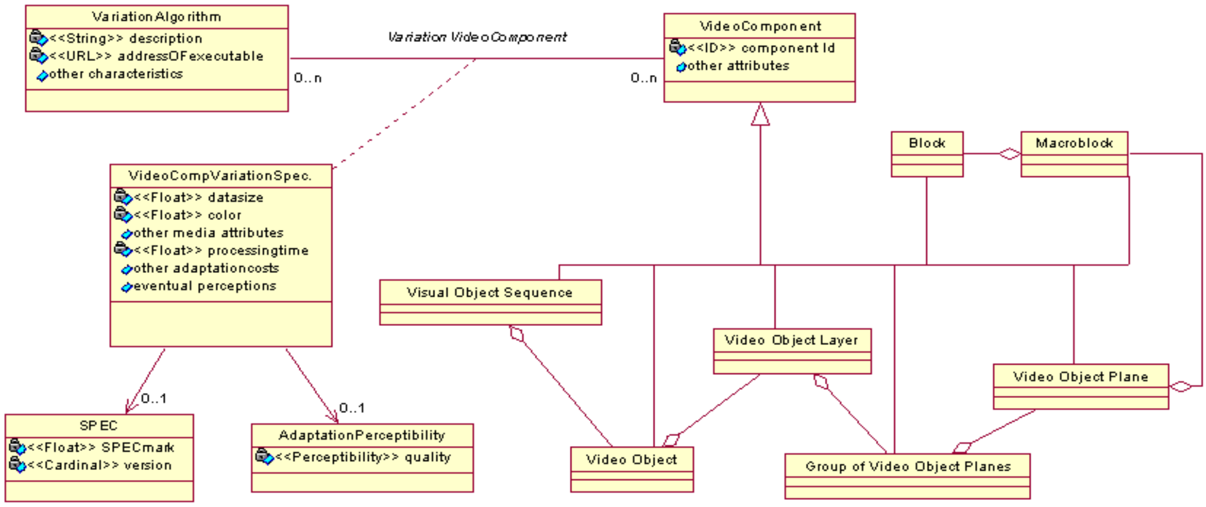

Figure 4: ***Description of the example variation in UML.***

the support of descriptive multimedia descriptions delivered from a MMDBS. This project started recently at our institute[1]. Further information can be obtained from `http://www-itec.uni-klu.ac.at/~harald/codac/`.

The MMDBS in our project is intended to store, besides content information of the media, information on their *quality adaptation capabilities*. This information may be obtained during the annotation process, on-the fly, or off-line during low access periods. The quality adaptation information may be of great interest to components on the delivery path to the client. Communicated quality adaptation description can be used in different active components for different purposes, e.g., on the proxy/cache level it can help to enhance the replacement strategy; on the network level it can be used to adapt dynamically to network congestion. Once the MPEG stream arrives at the component, one has direct and quick access to the meta-data which corresponds to the actual MPEG-4 access unit under consideration in the component. This direct access is enabled through a reference layer between the media streams and MPEG-7. This reference layer translates the synchronization information between the streams so that that media access units can be accessed from the meta-data.

Thus, upon the analysis of the meta-data, the component can quickly decide the best adaptation principle to apply (e.g., just drop the elementary stream showing the spectators in a soccer video stream and hence reduce the video size by one third).

In order to provide this descriptive information to the network components, the adaptation capability is encoded with MPEG-7. We will use the *VariationSetDS* and the *WeightDS*. The *WeightDS* is necessary to describe stream dropping and substitution of video objects by attributing priorities among elements of the different levels. The *VariationSetDS* is a more general DS and allows the specification of a set of quality variations of audio-visual data. Each quality variation specifies the variation algorithm in addition to the fidelity, priority and relationship with the source program. The *fidelity* of the variation program with respect to the source program is given by the fidelity value. The *priority* of the variation program with respect to the other Variations included in a VariationSet is given by the priority. The type of association or relationship of the Variation with respect to the source is given by the *VariationRelationship*. In addition to these, more elements are required to represent information of the meta-database, such as the variation costs (see the XML listing below).

Let us illustrate the modeling of quality variations and the mapping to MPEG-7 descriptions by an example of a color to monochrome filtering of MPEG-4 encoded videos. This variation algorithm affects the block and macroblock structure of a video. Fig. 4 shows a possible model, in UML notation, to represent the description of the variation information. The class *VariationAlgorithm* contains information about the algorithm, possibly a URL where the executable can be found, the syntactical level it concerns (here the block/macroblock level), and the quality adaptation descriptions which are independent of the video they apply to. The class *VideoComponent* is a super class for all syntactical units which an MPEG-4 video may possess (e.g., at the bottom of the layered encoding we have a block, and at the top level we may have different visual object sequences).

The association class *VideoCompVariationSpec.* contains the relevant information of the quality adaptation effects, which comprises information on: 1. variation of media attributes, like data-size reduction and color reduction; 2. adaptation cost, represented as processing time of the algorithm on a reference machine by a SPEC description (association to a class *SPEC* which contains the necessary SPECmark); 3. perceptual quality of the modified video (here simplified as a value in the range [0,1]: 1 for unchanged quality, 0 for complete loss of perceptual quality).

For instance, the following *complete* MPEG-7 description extracted from the meta-database implementation of Fig. 4, may accompany a requested *Video1.mpg* meeting the adaptation capability as described above. The *VariationSet* contains a *colorReduction* variation. A complete list of possible variation types can be found in [6]. The result file of the variation is assumed to be *Video2.mpg*. In order to describe the impact of the variation more completely, we propose to introduce two supplemental elements, *MediaAttribute* and *TranslationCost* in the *VariationSet DS*. These elements each contain a *fidelity* attribute which describes the impact of the variation on the *MediaAttribute* quantitatively. For simplification, the element value is supposed to be a number in this example. However and if required, one may introduce vector descriptions for the perception model-

[1] This project is supported by the FWF (Austrian Science Fund) under the project Number P14789.

ing, for instance. With these new elements, a more complete description of the variation impact may be obtained.

```
<Mpeg7>
 <Description xsi:type="VariationDescriptionType">
  <VariationSet>
   <Source xsi:type="VideoType">
    <Video>
     <MediaLocator> <MediaUri>file://Video1.mpg</MediaUri>
     </MediaLocator>
    </Video>
   </Source>
   <Variation priority="1">
    <VariationRelationship> colorReduction </VariationRelationship>
    <MediaAttribute> DataSize
      <VariationFidelity> 0.8 </VariationFidelity>
    </MediaAttribute>
    <MediaAttribute> Colors
      <VariationFidelity> 0.0039 </VariationFidelity>
    </MediaAttribute>
    <MediaAttribute> Perceptual Quality
      <VariationFidelity> 0.6 </VariationFidelity>
    </MediaAttribute>
    <TranslationCost>
      <ExecutionTime> 10 </ExecutionTime>
      <ExecutionTimeUnit> Seconds </ExecutionTimeUnit>
      <Reference idref="SPECMark"> </Reference>
    </TranslationCost>
    <Content xsi:type="VideoType">
     <Video>
      <MediaLocator> <MediaUri>file://Video2.mpg</MediaUri>
      </MediaLocator>
     </Video>
    </Content>
   </Variation>
  </VariationSet>
 </Description>
</Mpeg7>
```

This example assumes that three *MediaAttributes*, through which the reduction is measured, can be specified: *DataSize*, *Colors*, and *Perceptual Quality*. The data size shall be reduced by 80% and the color shall be reduced by a factor of 0.0039=(2^8 / 2^{16}) with 2^{16} originally available colors and remaining 256 gray scales. The perceptual quality has been observed as 60% of the original video. In the tag *TranslationCost* the execution time and respective SPECMark (as reference to another description to be delivered) is specified.

7. CONCLUSION

Broadly used MMDBSs now support multimedia in its different variety, i.e., audio, video, images, and also synthetic data and propose sophisticated application packages like SQL/MM. The supported multimedia objects can not, however, be considered as stand-alone objects; they are conceptually related and have to exist in conjunction with each other. Synthesizing and synchronizing the various forms of information provided by the multiple media is a very important part of multimedia applications.

In this scope we described how the new MPEG-7 Multimedia Description Standard provides us a structured metadata description for semantically rich media content. We demonstrated that MPEG-7 should *not* be considered as a competitor to broadly used MMDBS data models. Based on these considerations, we presented how MPEG-7 might facilitate the communication of a MMDBS with its environment and presented a case study.

It is up to the future MMDBS designers and realizers to make use of MPEG-7 and to speak in a commonly understandable language in a highly distributed world of multimedia data.

Further Reading:

The first six parts ($1 \leq x \leq 6$) of MPEG-7 are International Standard (ISO/IEC 15938-x:2001). They may be purchased from ISO (`mailto:sales@iso.ch`) or from a National Body. In addition, MPEG publicly posted the Final Committee Drafts at `http://mpeg.telecomitalialab.com/hot_news.htm`. Part seven and eight (conformance, extraction and use) are scheduled for IS in 05/2002 for part seven and in 09/2002 for part eight. In addition, MPEG-7 descriptions/bitstreams are posted at `http://pmedia.i2.ibm.com:8000/mpeg7/schema/`.
XML documents can be validated online against the schema at `http://m7itb.nist.gov/M7Validation.html`.
The most recent overview of MPEG-7 can be found at:
`http://mpeg.telecomitalialab.com/standards/mpeg-7/`
Additional information on ongoing projects using MPEG-7 may be found at the MPEG-7 Application Site at `http://www.mpeg-industry.com/`.

8. REFERENCES

[1] Guojun Lu. *Multimedia Database Management Systems.* Artech House, 1999.

[2] S.-C. Chen, R.L. Kashyap, and A. Ghafoor. *Semantic Models for Multimedia Database Searching and Browsing.* Kluwer Press, 2000.

[3] Jim Melton and Andrew Eisenberg. Comparative analysis of five XML query languages. *SIGMOD Records*, 30(4), 2001. `http://www.acm.org/sigmod/record/issues/0112/standards.pdf`.

[4] José M. Martínez. *Overview of the MPEG-7 Standard.* ISO/IEC JTC1/SC29/WG11 N4509, December 2001. `http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.html`.

[5] S.-F. Chang, A. Puri, T. Sikora, and H. Zhang. Special issue on MPEG-7 (12 papers). *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6), June 2001.

[6] P. van Beek, A.B. Benitez, J. Heuer, J. Martinez, P. Salembier, J. Smith, and T. Walker. *MPEG-7: Multimedia Description Schemes.* ISO/IEC FDIS 15938-5:2001, October 2001.

[7] R. Weiss, A. Duda, and D. K. Gifford. Composition and search with a video algebra. *IEEE MultiMedia*, 2(1):12–25, 1995.

[8] H. Jiang, D. Montesi, and A. K. Elmagarmid. Integrated video and text for content-based access to video databases. *Multimedia Tools and Applications*, 9(3):227 – 249, 1999.

[9] V. Oria, M.T. Özsu, B. Xu, L.I. Cheng, and P. Iglinski. VisualMOQL: The DISIMA visual query language. In *IEEE International Conference on Multimedia Computing and Systems*, volume 1, pages 536–542, Florence, Italy, June 1999.

[10] R. Tusch, H. Kosch, and L. Böszörményi. VideX: An integrated generic video indexing approach. In *Proceedings of the ACM Multimedia Conference*, pages 448–451, Los Angeles, USA, Oct.-Nov. 2000.

[11] H. Kosch, R. Tusch, L. Böszörményi, A. Bachlechner, B. Dörflinger, C. Hofbauer, C. Riedler, M. Lang, and C. Hanin. SMOOTH - A distributed multimedia database system. In *Proceedings of the International VLDB Conference*, pages 713–714, Rome, Italy, September 2001.

[12] Y. Rui, T.S. Huang, and S. Mehrotra. Exploring video structure beyond the shots. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 237–240, Austin, Texas, USA, June 1998.

[13] D. Zhong and S.-F. Chang. Structure analysis of sports video using domain models. In *Proceedings of the IEEE Conference on Multimedia and Exhibition*, Tokyo, Japan, August 2001.

[14] Angela Bonifati. Comparative analysis of five XML query languages. *SIGMOD Records*, 29(1):68–79, 2000. `http://www.acm.org/sigmod/record/issues/0003/bonifati.pdf.gz`.

[15] Jörg Heuer, Jose Luis Casas, and Andre Kaup. Adaptive multimedia messaging based on MPEG-7 - the M3 box. In *Proceedings of the Second International Symposium on Mobile Multimedia Systems & Applications*, Delft, The Netherlands, November 2000.