# Video Database Management System

A Mini thesis submitted for transfer of registration from M-Phil to PhD.

May 1999,

University of Southampton U.K.

# Contents

# 1.0 Introduction

Advances in multimedia computing technologies have offered new opportunities to store, manage, and present video data in databases. Object oriented technology on the other hand provides novel ways to organise the multimedia items, by supporting both the temporal as well as spatial properties. Many researchers come up with data models by implementing object oriented concepts into multimedia technology. The aim of this work is to make a video data model, called VDbMS (Video Data Base Management System) by incorporating and integrating the above mentioned technologies.

This report summarises the current progress of this research in the development of VDbMS. The intention of this database system is to provide an effective way for storing video data, optimal query parsing and an ideal interface for user interactivity.

This report is separated into six sections. Section 2 briefly reviews the previous twelve months. Section 3 depicts the work carried out in the next six months in developing the video model. Section 4 tells us about the experiments on the video data. Section 5 is about the conclusions and the line of action for future work and finally section 6 is the appendixes for some video models and MPEG4

# 2.0 Initial work carried out

Following is the summary of the initial six months, which can be termed as the foundation of my research work:

## *2.1 Literature Review*

A lot of work is going on in the field of video data modelling and its applications, due to its high commercial value. Since video itself is a very complex object with both spatial and temporal properties, researchers are still unable to get a proper and comprehensive data model. A summary of literature review covering various video data models (annotation dependent, hierarchical, object-oriented, algebraic, etc.), video content-based understanding, indexing, segmentation of scenes and objects, visual learning, query processing and different video formats is discussed below:

### *2.1.1 Data Modelling*

The functionality of a database is measured by data manipulation and query processing. A query for video data is very complex in nature. It is also possible that a database user can make a content based query (objects inside the video), as well as feature based (technical specifications of the video i.e. frame rate, segment change etc.). Various models have been suggested by different researchers, based on relational, hierarchical and object oriented concepts. The two major modelling schemas are:

**Annotation Dependent Models:**

These models heavily depend on hierarchical or relational databases. In these type of models, some textual data (also known as signatures or metadata) about the contents of video are embedded into the video data. Usually an annotator is required to write the captions, but in some models dynamic text can be generated from the audio. VidIO[1], VANE[2], Video-STAR[3] and Media Object [4]are the models of this category.

**Object Oriented Models:**

Object Oriented Database Management Systems are considered to be good for defining multimedia models, as they can entertain both spatial as well as temporal features of any media. Another property of a media object is that it can define its own data rate, abstraction and other attributes. Again an AV (audio video) object can be accessed concurrently. OVID[5], Video Widgets[6], MPIL[7] are examples of object oriented video modelling.

Apart from the above stated models, researchers have also tried to solve the problem of video modelling by using complex algebraic methods[8], frame based techniques[9], or making a hybrid system[10], comprising all or some of the above stated models. These models are further discussed in Appendix A.

### 2.1.2 *Query Processing*

A browser enables a user to access all information related to a specific piece of video, querying makes it possible to formulate some conditions and then retrieve only the video material that have desired properties. In VDbMS, a query can give the output defined as exact match, in-exact match or similar match, where the major data indexing is based on annotations, video technical data such as frame rate, colour histogram, etc and audio data[11].

### 2.1.3 *Compression Techniques*

Compression is the key to low data rate for digital video and a number of studies have undertaken to devices suitable compression algorithms and techniques. These methods can be categorised into lossy or loss-less compression methods. Some of the industry compression standards are DVI (Digital Video Interface), MPEG (Moving Pictures Expert Group), M-JPEG (Motion Joint Pictures Enhanced Group), Apple QuickTime, and Microsoft AVI (Audio Video Interleaved). It was found that MPEG-4 is the best for multimedia applications, at the moment as it provides content based interactivity, improved compression techniques and support for working at a very low bit rate (< 1Mbits/sec)[12,13]. Keeping the advanced functionality of MPEG-4 in view, VDbMS is designed to be totally compatible to this standard. MPEG-4 is discussed in detail in Appendix B.

## 2.2 *Initial Experimentation*

Trial experimentation was started by making a video database of some videos from the Lord Mountbatten archives. It was decided that the research direction would be in the area of digital video archives and their databases.

The major task for next six months was to design and develop a comprehensive video data model, called VDbMS (Video Database Management System), which can incorporate temporal as well as spatial entities of a digital video.

## 3.0 The current state of Progress

During the last twelve months, I have carried out the task of understanding the video data formats and then designing an object oriented video model for VDbMS. At the same time trial experimentation was continued on the Lord Mountbatten documentaries. First, using a basic scene detection algorithm, the video was broken into different scenes. These scenes were then given some captions and stored in a simple database. Still no impressive results were found. Then it was decided that a comprehensive data model should be developed to support all the video-related functions.

Section 3.1 discusses the basic types of information available in the video data, 3.2 describes the VDbMS model, the object entities used in the model and a note on thematic indexing, 3.2.3 discusses video object tree and finally section 3.3 describes data query and retrieval.

### *3.1 Video Data*

A video database will store different types of information, which is divided into streams, sequences, scenes, shots and frames. A video database can be defined as a collection of sets where the elements of each set are elements of the same type. Here the object – oriented approached is used in VDbMS, so that each element is identified by a unique object identity. Objects may be complex with attribute that are objects on their own.

Following is a short description of the different object types, which are relevant for the data model of VDbMS:

- **Stored Media Segments:** Audio & Video data which are generated during recording.

- **Video Streams:** Video documents are composed from pieces of stored media segments. A video document represents a logical stream of video data that may be explicitly stored.

- **Media Streams:** Media streams are a generalisation of stored media segments and video streams.

- **Stream Interval:** A stream interval represents a contiguous sequence from a media stream that is explicitly identified.

- **Video Content Indexes / Annotations:** For browsing and querying purposes there is a need to relate entities from a real-world model to pieces of video. This relation is made by annotating an video object that identifies the stream interval of interest and that is linked to an element of a real-world model.

- **Video Document Structure:** Video documents can have a certain structure. This structure can be represented by a set of structural components where each component identifies a stream interval.

Video data is stored as contiguous groups of frames called stored video segments. A video document is represented by a video stream which is mapped to one or more stored video segments. An important and flexible video information unit is frame sequence which is an interval of subsequent frames from a video document. One single frame sequence can represent any sequence of video frames ranging from one single video frame to an entire video document. The frame sequence is also responsible for connecting structural units and thematic annotations to the video material. A frame sequence can also be defined as a 'part of' relationship to a video stream, as shown in fig (1).

The structure of a video document is represented by a hierarchy of structural components. Each structural component identifies a frame sequence which consists of the frames that belong to the component. The entire video document or a single frame can also become a structural component, but a whole document is too coarse as a level of abstraction and a single frame is rarely a unit of interest[9].

From experiments, it has been learnt that abstractions such as scenes or events make it easier for a user to make references to video information and easier to comprehend its contents. Therefore more emphasis in VDbMS is given to metadata which are categorised as Event, Location, Person, Object_of_Interest metadata.

Since a huge metadata is created while annotating different video scenes, it is very important that some standards for metadata should also be followed. While doing literature review, it was found that WWW consortium (W3C) is coming up with Resource Description Framework (RDF), which provides a more general treatment of metadata. RDF is a declarative language and provides a standard way for using XML to represent metadata in the form of properties and relationships of items on the Web. Such items, known as *resources*, can be almost anything, provided it has a Web address.

Along with ISO also introduced MPEG-7 standard, which will specify a standard set of descriptors that can be used to describe various types of multimedia information. MPEG-7 is discussed further in Appendix 2.

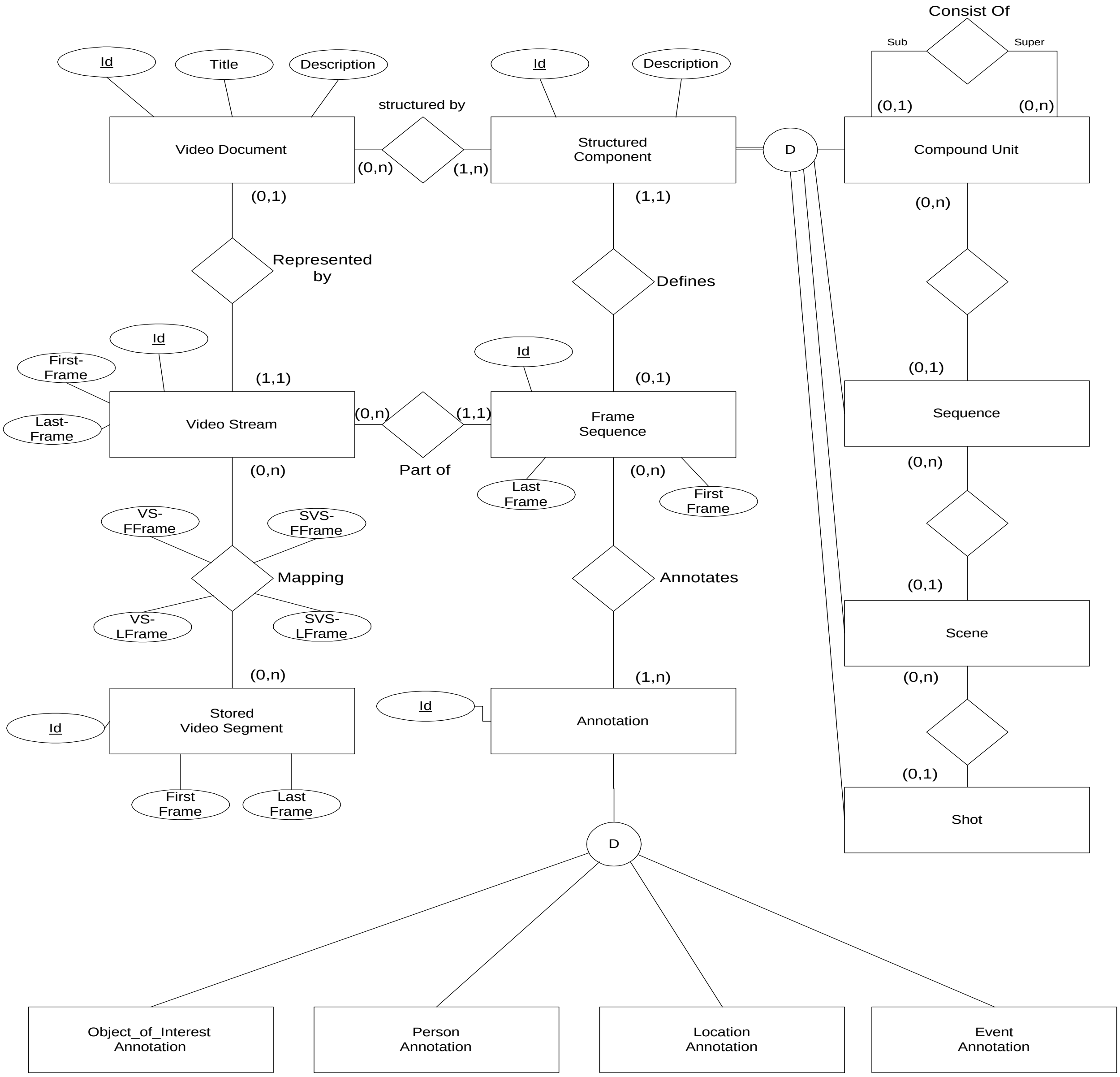


***Fig (1) A common data model for video information sharing using an ER-notation, suggested by Hjesvold[27]. Here the video document is catogorised into Frame Sequence, which is further divided into different types of annotations***

## 3.2 Video Data Model

The work started with the idea that a large or long video should be divided into very small fragments or clips of a few seconds, depending only on simple scene detection algorithm. Soon it was found out that this technique is useless due to the following drawbacks:

- It was impossible to track the main story of the video, as one small video clip was unable to show the main theme.

- Query search  was not possible, as a query can be of a concept, story or an event, whereas our division was based on a simple scene change technique, which was just the histogram difference of the two shots.

- This technique might be suitable for videos such as news clips or a catalogue of commercials, but when a full hour documentary programme is processed (as in our case), event sequence, frame sequence and hierarchy of scenes cannot be established.

It was also found that in documentary movies, like that  of Lord Mountbatten, video modelling was heavily dependent on annotations or metadata of a video. Since algorithms of computer vision are still not able to get the whole data of a scene or a sequence, especially concepts, themes and ideas, so special importance to the metadata is given in VDbMS model.

Audio data is the other domain providing a lot of information about the video. Applying any speech-to-text converter, one can generate streams of data which can be used to extract metadata.

Apart from metadata, data regarding frame size, frame rate, colour, texture and other features are also of important value. To create a comprehensive data model, all the above mentioned entities should be modelled to such a way that issues such as data storage, query processing and data indexing should be optimally resolved.

During the literature review phase, it was found that  Hjesvold [14] had developed a very precise model for annotating a segment. Metadata is generated from audio components as well as video components. Textual information about  an event, person, location or an object is also inserted, as shown in figure (2). Hjesvold's  idea appeared very relevant and was made the part of VDbMS model, with slight modifications such as adding an entity of concept in the metadata and placing the event entity top of the  segment entity in the hierarchy.

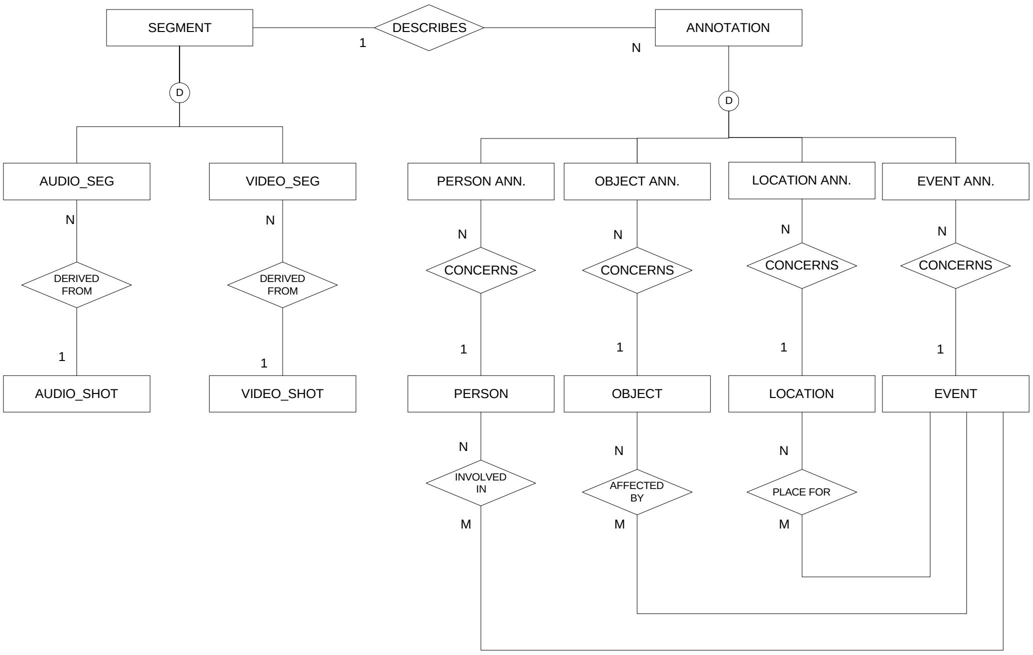


***Figure (2) Data components of a generalised video segment***

For VDbMS, considerable modifications are made in the data model shown in figure (2). Since in VDbMS model the entity is considered as event, and a segment is a subset of an event, so the event has been made the pivotal item, which comprise of segments, as shown is figure (3). Further the segment describes the annotations for the objects inside the scene, such as person, location, etc. Each object has its own metadata, which creates a many-many relationship with the segment metadata. This is then normalised to create a relational database for accessing any object present in the video scene.

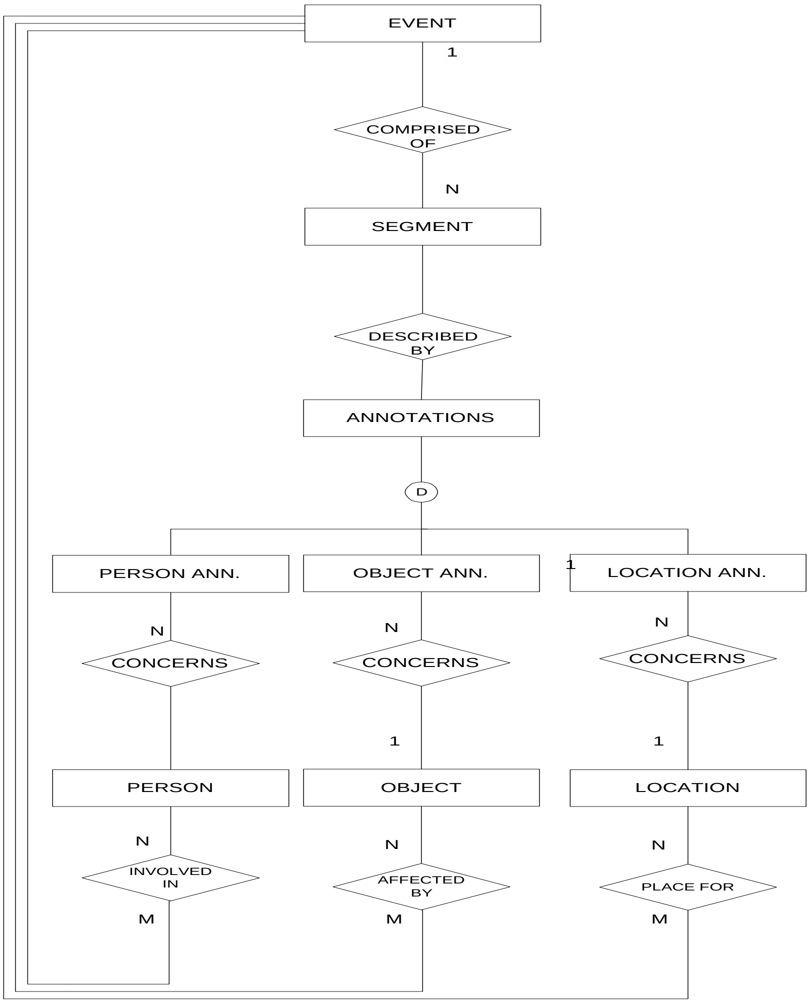


***Figure (3) Suggested Annotation model for VDbMS. Here 'event' entity is added and been assigned a top level in hierarchy, as opposed of Hjesvold's model[27].***

### *3.2.1 Object Entities*

In this section, all the operations and attributes applicable to a composite entity are discussed. The attributes of a composite entity are:

| | |
|---|---|
| Object Identity: | The unique identity of the component |
| Symbolic Name: | The text string that the author may optionally assign to the component to have a user generated name for identifying the component. |
| Synopsis: | Textual description of the component. |

| | |
|---|---|
| Architecture Operations: | This group includes operations for creating, modifying or deleting components in the component architecture and for manipulating the synchronisation of components in a narrative presentation. |

An event is defined as the set of scenes, constituting a part of a story. In the VDbMS data model the major object entity is  an event.  It should be noted that even humans' memory circle around events. When people recall something, they often say "*I played a movie of that event in my mind"*. It tells us that an event plays a very important role in our memory [2,6]. An event comprising of one or more segments, can be generated by another event or can generate many sub events, and many events can generate a single sub event, as shown in figure (4).

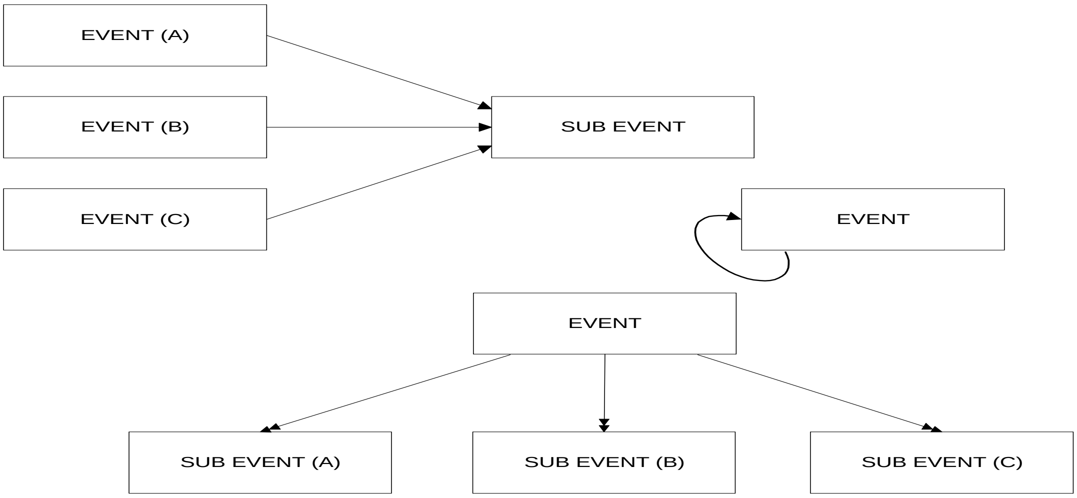


***Figure (4)  Inheritance properties of an Event***

Hence the event entity is proposed over segment entity, as shown in figure (4), as compared to Hjesvold's generalised video model in figure (2). The main objective in this modification is to maintain the sense of a story. Again the other entities, such as person, location, object_of_interest, etc. are the sub group of the segment. This ER diagram shown in figure (4) can be normalised accordingly and can be adjusted in the main data model.

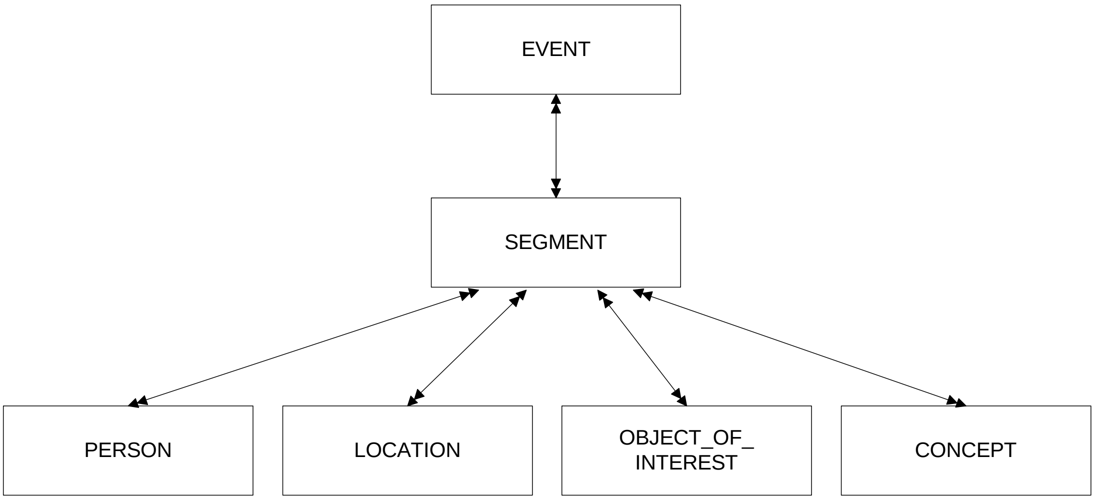


*Figure (5)  ER Diagram of Annotation Model*

It should be noted in figure(5) that another entity of concept is introduced in the model. While watching the Lord Mountbatten documentary, it was found that some scenes require additional information. Again some scenes were found which were impossible to define using physical objects only. So another entity of 'concept' is introduced in the data model, where a user can provide some textual information about  the scene. The idea of concept as a textual study is also used in Vane [2], where the entities are classified as tangible and conceptual entities.

### *3.2.2 Thematic Indexing*

A theme is defined as a topic of discourse or discussion [38]. While watching a movie, one can find that many themes are interwoven with in and around a story. For the documentary of Lord Mountbatten, it was realised that two main themes substantially form the whole story. One theme is of Lord Mountbatten himself, his life, his career as a commander in the armed forces and as a dignitary in Burma and India. The second theme is about the war between the Allied Forces and the Japanese. Since a war is fought between nations, soldiers are involved, weapons are used and injuries and fatalities occur. Here the two themes combine as Lord Mountbatten himself is a soldier and a soldier fights a war.

Figure (6) shows the object model of the above mentioned themes. This model can be very useful in indexing and cataloguing the data, as the sense of the story is not broken and structured data is obtained as well. Since this model is object oriented, real time modification is also possible.

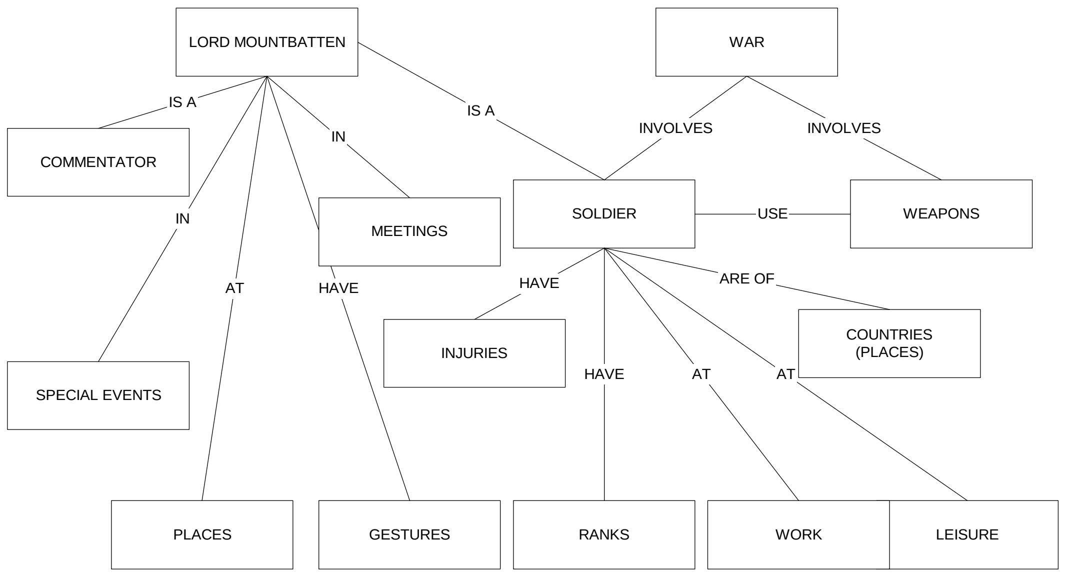


***Figure (6) Thematic Indexing of Lord Mountbatten's documentary***

It should be noted that both the themes are implicitly related to each other, so indexing and query processing can easily be done at various levels in the hierarchy of entities. For example if a user is querying about *Air Bombing*, this will come under weapons entity, which is a sub-part of the war entity. Another benefit of hierarchical modelling is that, at a certain level, if the search engine is unable to obtain some match, it can always go back to the parent level and fetch some related data.

### 3.2.3 Common Video Tree Model

Li, Goralwala, Ozsu and Szafron [39,40], gave the idea of a Common Video Object Tree (CVOT), where objects contained in a video frame can be accessed. To understand this tree model, following axioms should be determined:

Let a video clip "C" be associated with a time interval [$t_s$, $t_f$]

Where $t_s$ → starting time of clip

$t_f$ → ending time of clip

Here $t_s$ and $t_f$ are relative (discrete) time instants and $t_s \leq t_f$

Since all clips have a start and finish time, a partial order could be defined over clips,
i.e. Ci = [$t_{si}$, $t_{fi}$]

Then the following axioms hold true in a digital video,

***Partial Ordered Clips:***

Let $C_i$ = [$t_{si}$, $t_{fi}$]

$C_j$ = [$t_{sj}$, $t_{fj}$]

Then $\prec$ is defined as the partial order over clips with Ci$\prec$ Cj iff:

$T_{si} \leq t_{sj}$ and

$T_{fi} \leq t_{fj}$

***Ordered Clips***

$C_i$ is said to be ordered iff C is finite i.e. C = {$C_1$, $C_2$,..……,$C_m$} and there exists a time order, such that $C_1 \leq C_2 \leq C_3$ …..…. $\leq C_m$

***Perfectly Ordered Clips***

A set of clips C = { $C_1$, $C_2$, …., $C_m$} is said to be perfectly ordered iff C is ordered and for some two neighbouring clips i.e. $C_i$ = [$t_{si}$, $t_{fi}$] and $C_{i+1}$ = [$t_{si+1}$, $t_{fi+1}$], we have $t_{si+1} = t_{fi+1}$

***Strongly Ordered Clips***

A set of clips C = {$C_1$, $C_2$, ….,$C_m$} is said to be strongly ordered iff $C_1 < C_2 < C_3$ ..< $C_m$

***Weakly Ordered Clips***

A set of Clips C = {$C_1$, $C_2$,….,$C_m$} is said to be weakly ordered iff C is ordered for two neighbouring clips, $C_i$ = [$t_{si}$,$t_{fi}$] and $C_{i+1}$ = [$t_{si+1}$, $t_{fi+1}$] and we have $t_{fi}$>= $t_{si+1}$ (∃i = 1,2,….,m-1)

Hjelsvold and Midtstraum [3] also define some temporal operations on video clips. These operations are annotation dependent and can be expressed in a traditional query language. Some of these operations are

**$C_i$ EQUALS $C_j$:** Returns true if $C_i$ and $C_j$ are identical

**$C_i$ BEFORE $C_j$:** Returns true if $C_i$ happens before $C_j$

**$C_i$ MEETS $C_j$:** Returns true if $C_j$ starts with the next frame after $C_i$ has ended.

**$C_i$ OVERLAPS $C_j$:** Returns true if $C_j$ starts while $C_i$ is still active

**$C_i$ CONTAINS $C_j$:** Returns true if $C_j$ starts after $C_i$ and ends before $C_i$

**$C_i$ STARTS $C_j$:** Returns true if $C_i$ and $C_j$ start with the same frame and $C_i$ ends before $C_j$

**$C_i$ FINISHES $C_j$**: Returns true if $C_j$ starts before $C_i$ and $C_i$ and $C_j$ end with the same frame.

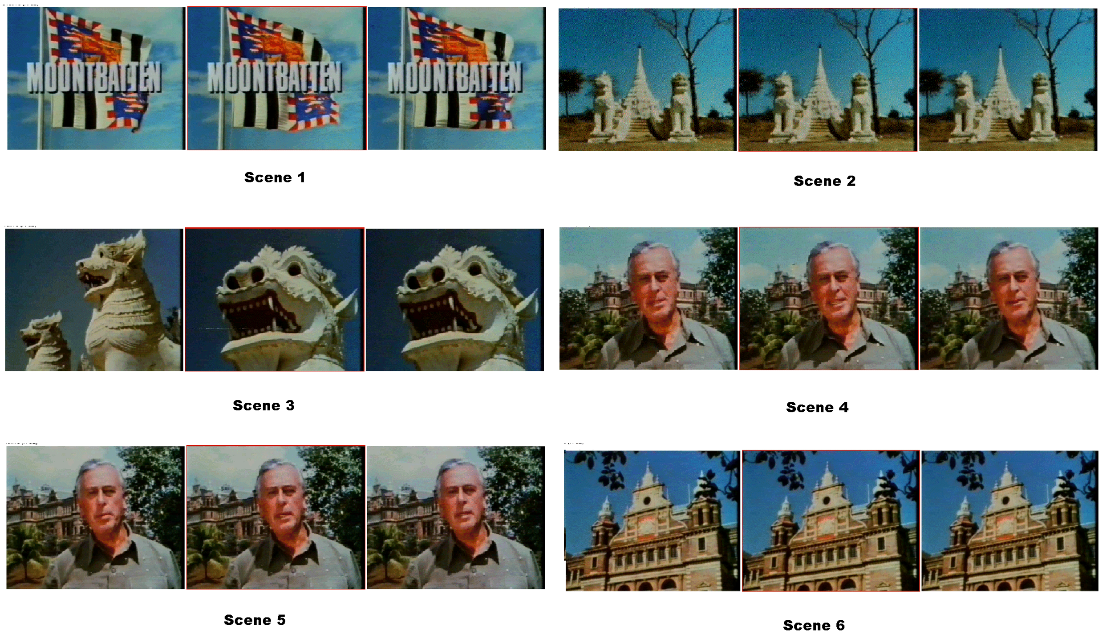
If we look closely to these operations, they are more likely same, however quite important for the foundation of VBDMS thematic indexing model. Further these axioms provide all the operations required to query a video.

### *3.2.4 Objects in Video Clips:*

A video frame has many noticeable real finite number of objects like persons, house, cars, etc. Its assumed that there is always a finite set of objects, for a given video. Let "O" be the collection of all the objects and "C" be the collection of all the video shots. Then the common objects for a given set of video shots are the objects which appear in every clip within the set. Then the intersection of m video shots can be defined as

$$\mathbf{(C_1,C_2,\ldots..,C_m) = \{ \Upsilon(C_1) \cap \Upsilon(C_2) \cap \ldots..\cap \Upsilon(C_m)\}}$$

Where $\Upsilon(C_i) \rightarrow$ set of objects in Video shot $C_i$



Scene 1

Scene 2

Scene 3

Scene 4

Scene 5

Scene 6

*Figure (7) Noticeable objects in different video scenes*

For Example in Figure (7), we have six video shots, from Lord Mountbatten's video. Clip 1 contains a flag, Clip 2 contains titles, a tower and Burmese lions statues in the background. The other clips contain Lord Mountbatten, a palace in Burma. If the video is segmented, then

$$C = \{C_1,C_2,C_3,C_4,C_5,C_6,C_7\}$$

Furthermore,

$C_1$ = Flag

$C_2$ = Tower, Burmese lions statues

$C_3$ = Burmese lions statues

$C_4$ = Mountbatten, palace

$C_5$ = Mountbatten, palace

$C_6$ = palace

With the object point of view

ϒ(Flag) = $C_1$

ϒ(Tower) = $C_2$

ϒ(Lion statues) = $C_2$,$C_3$

ϒ Mountbatten) = $C_4$,$C_5$

ϒ(palace) = $C_6$

Hence now a relational tree can be developed by using any of the axioms stated above. For example if we use the intersection axiom, then we have,

$C_1 \cap C_2 =$ Null

$C_2 \cap C_3 =$ lion statues

$C_4 \cap C_5 =$ Mountbatten, palace

$C_5 \cap C_6 =$ palace

Similarly we can build a hierarchical affinity like

$C_4 \cap C_5 \cap C_6 =$ palace

In this way we construct an object tree, shown in figure (8).

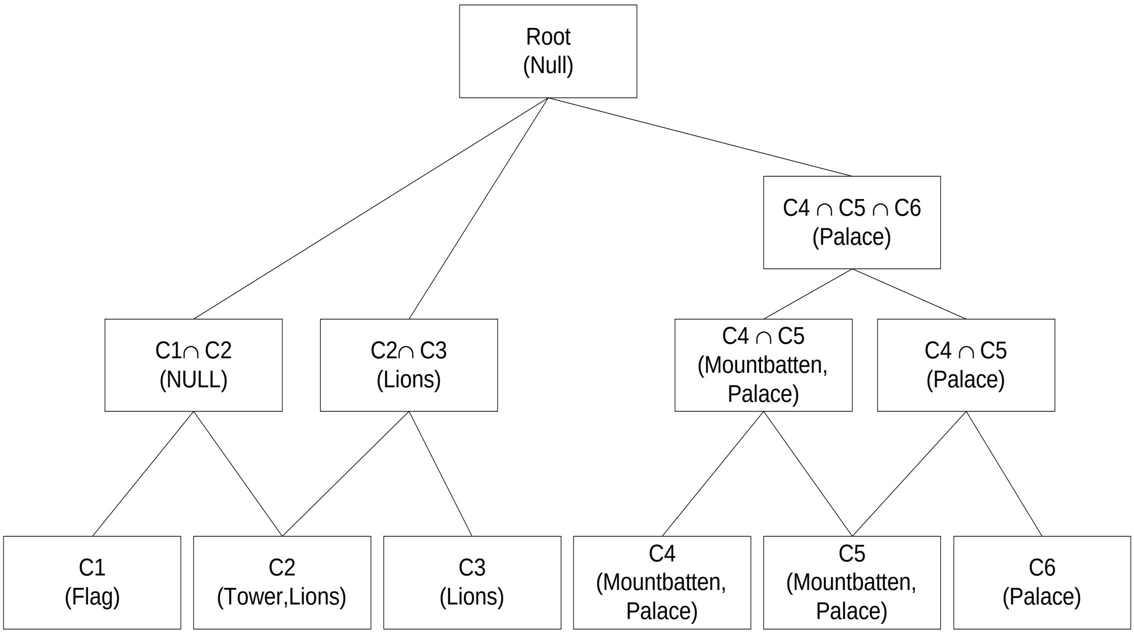


***Figure (8). A Video Object Tree***

Once the tree is established, a query can be created, by using any of the axioms. It should be noted that this sort of tree can be developed at the frame, segment or scene level. Once this tree is ready, the next step is to create generic links between these objects. For example Mountbatten is a Soldier. Here *"is a"* relationship has been created between two objects Mountbatten and Soldier. These generic relationship links will also provide themes of the story. In this way one can also follow the story line of the video with thematic indexing.

## 3.3 Data Query & Retrieval

The efficiency of a database is evaluated by the nature and complexities of the queries, that will be made about data. In terms of video the query and retrieval process becomes more complicated by the numerous demands placed on the system. Elmagarmid [11] describes the video data retrieval system in the following simple steps. First, the user specifies a query using a facility provided by the user interface. The query is then processed and evaluated. The value or feature obtained is used to match and retrieve the video data stored in the database. At the end, the resulting video data is displayed on the user interface in suitable form.

### *3.3.1 Query types*

Since the video data is spatial and temporal, the queries are heavily dependant on their data content. Along with the architecture of the video data model and intended applications are also many other factors that modify a query. A query can be divided into:

**Query by Content:** These queries are further categorised as semantic information query (content information in the scene) , meta information query (scene description information) and audio-visual query (audio and visual feature of a scene)

**Query by Nature:** These queries depend on the nature of the video content and can be further categorised in spatial or temporal aspects of the video

### *3.3.2 Query Certainty*

The certainty of a query can be specified in terms of the type of matching operator used to satisfy the query. A query can fall into Exact match, Inexact match, or similarity matched queries.

Hjelsvold [15], in his Video-STAR data model, defined a video query algebra, that allows the user to define complex queries based on temporal relationships between video stream intervals. These operations include normal Boolean set operations *(AND, OR)*, temporal set operations *(i.e. stream A equals stream B, A is before B, A meets B, A overlaps B, A contains*

*B, A starts B and A finishes B),* annotations operations that are used to retrieve all annotations of a given type and have non empty intersections with a given input set and mapping operations that map the elements in a given set onto different contexts that can be basic, primary or video stream.

### *3.3.3 Query Processing*

This process usually  involves query parsing, query evaluation, database index search and the returning of results. In the query parsing phase, the query condition or assertion is usually decomposed into the basic unit and then evaluated. Along with text based search for annotations, feature based search is also applied to extract spatial contents like colour, motion, texture of the video data. Here thematic indexing will also aid to retrieve data for a query. Now the index structure of database is searched and checked. The video data are retrieved, if the assertion is satisfied or if the similarity measured is maximum and  finally the resulting  a  video data are usually displayed by a GUI, developed by the user.

## 4.0 Experimentation

The major requirement for our experiments was a need of a video player, which allow us to annotate, insert metadata and overall allow us to insert a link into a digital video. While having a look on the current experimental and commercial video players, we decided to develop our own video player, which can allow us to do the above mentioned task, as well as to match the following considerations:

- It should be capable enough to run any sort of video format, e.g. avi, mpeg, etc
- It allows to connect the video file with a database, which can provide metadata for the video
- Once starting and ending frames are selected, the player should be able to run the selected part of video

Soon a player was developed in the Microsoft windows environment, which was capable to fulfil the above mentioned requirements. Then a database of annotations, based on the methods stated in section 3 was developed in Microsoft Access 97, and was then connected to the video player. Figure (9) shows the front-end of the VDbMS video player

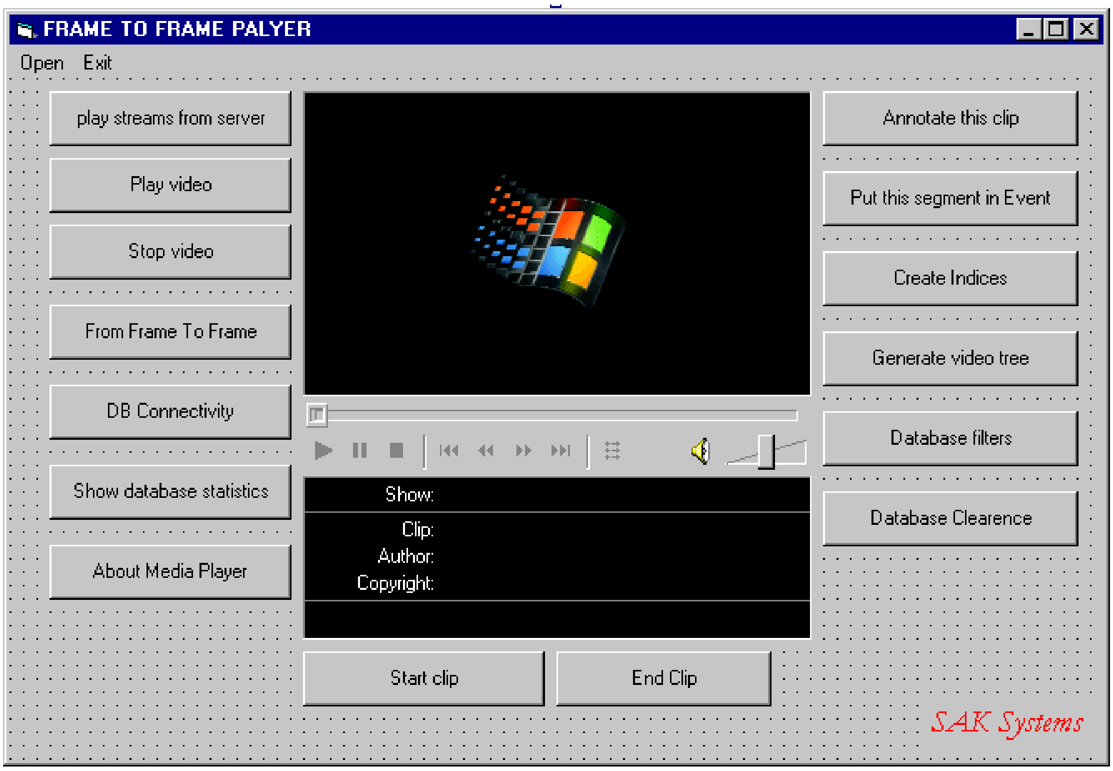


***Figure (9) Front end of VDbMS Video Player***

A number of options have been incorporated into this video player. On the left hand side of the video player, video options and on the right hand side database options are provided. For example this player is directly connected to the "Godzilla" machine, which is currently working as the temporary video server for the MMRG group. Along with pressing the "Frame to Frame" button, the user can run a video for a desired no of frames. The "DB-Connectivity" button connects the player with the VDbMS database. On the other side, options for annotating a video segment, attributing a video segment to an event, creating and re-establishing indices, database filters and database garbage control buttons are also provided.

The second task was to construct a query output window, so that the user can see up to 12 nearest selected portions of video for his/her query. Figure (10) shows the output for a query about "people" in a news clip.

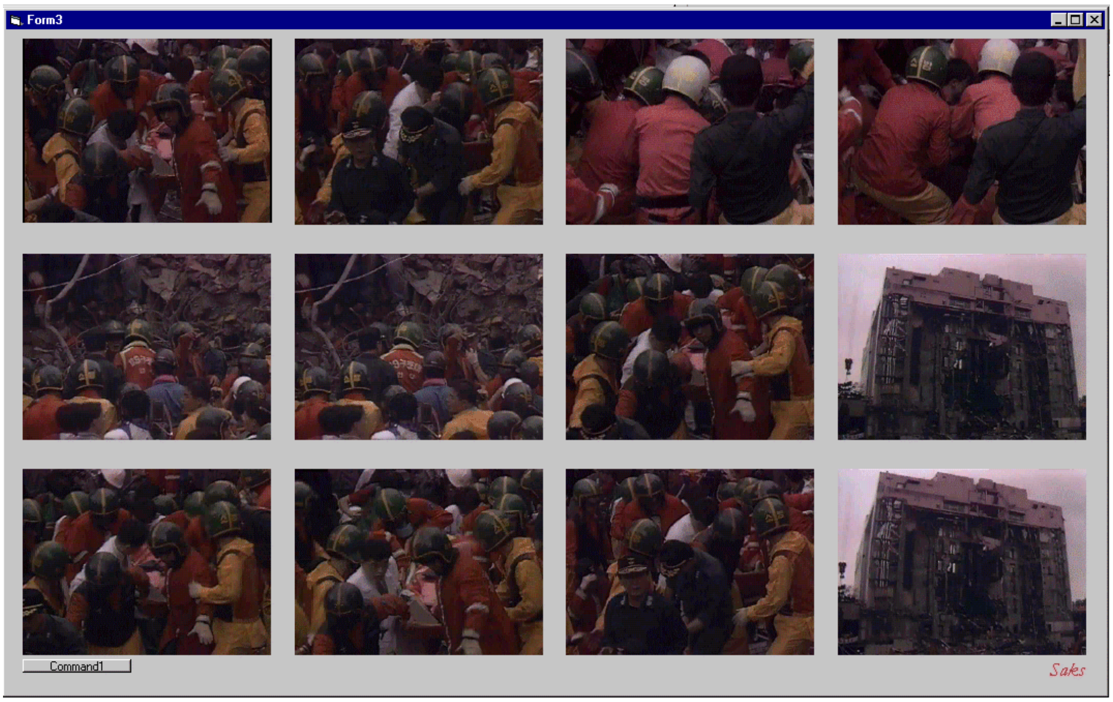


*Fig (10) Query output box for VDbMS*

## 5.0 Summary & Future Tasks

The next task is to increase the efficiency of the VDbMS query module, by inserting more data to thematic indexing modules and video object trees. It should be noted that that the model will be modified throughout the research work, so the development of the model will be altered accordingly. At the moment  the following tasks are suggested for further research:

- Designing and developing new techniques to incorporate thematic indexing model
- MPEG-4 standard incorporation with VDbMS
- Query complexity
- Incorporating the technical video specifications in the data model
- Working of complex issues such as sharing and re-using  digital video

There are many projects that deal with video databases, developed in recent years, with special properties, ranging from editing, segmentation, indexing, searching, browsing, sharing and re-using and synchronising video presentation. VDbMS on the other hand is novel in such a way that it demonstrates functionality for supporting thematic indexing and query parsing. Again the model also entertains a wide range of metadata, providing a wide range of attributes for a user to perform query. Still a lot of work is required to make this model effective up to optimal.

## Proposed Timetable for research work up to October 2000

| | |
|---|---|
| *July 1999 - September 1999* | Exhaustive experimentation on Lord Mountbatten archives, by providing metadata to the long videos and using VDbMS data model and thematic indexing. |
| *October 1999 - December 1999* | Further research on query processing and query complexity. Along with, some improvements in VDbMS browsers will also be done to provide further accessibility to the VDbMS users. |
| *January 2000- February 2000* | Implementation of MPEG -4 and -7 protocols in VDbMS. |
| *March 2000* | Evaluation of my work and comparing it with other available models<br>Some research on video content based searching, sharing and the re-use of digital video. Tracking of the state of the art technology at that time. |
| *April 2000* | Evaluation of my work and comparing it with other available models<br>Submission of one research paper in some international conference / research journal. |
| *May 2000 - August 2000* | Writing up of PhD thesis and final touches to the experimental work. |
| *September 2000 - October 2000* | Submission of thesis. |

# APPENDIX 1

## A1 Video Data Modelling

Data modelling is essentially required to present video data based on its characters, information extent and the applications it is intended for. It plays a very important role in the system since all the functions, procedures and classes are dependent on it. Again all the features, outputs, storage spaces, processes are also described at this stage.

Looking at the complexity of a video stream, the contents attribution and semantic structure of the video need to be described in a data model. A generic video model should also support the following features:

- Multi-level video structure abstraction
- Spatial and temporal relationship support
- Video annotation support

The conventional relational database model suggested by Codd [35] in the early 70's was not able to represent multidimensional video data as all sort of rich-information-file-format data are considered as simple binary large object files (blob), which is very insufficient to incorporate temporal and spatial features. In the past few years, a lot of data models based on different techniques have been proposed, which can be categorised as annotation dependent, annotation oriented, object oriented, and others. Following is a discussion on some of these models:

### *A1.1 Annotation Dependent Models*

In these type of models, some textual data (also known as signatures or metadata) about the contents of video are embedded into the video data. For example the annotations for a typical television news clip will be news date and time, news headlines text, news captions, scene captions etc. Usually an annotator is required to write the captions, but in some models dynamic text is also generated, depending on scene detection algorithms and other semantics, generated from the spatial characteristics of the video. Examples include histogram colours, basic shapes, textures and other information generated from audio contents. Following are some of the data models, which depend heavily on annotation data:

### *A1.1.1 VidIO*

In the Multimedia Research Group, at the University of Southampton, work on video databases was started by Salam and Hall [3]. They developed the system, VidIO (Video Information Organiser), that takes account of the importance of different perspectives of different users in video retrieval.

The main feature of this system was the provision of support for the creation and maintenance of personalised video materials (i.e. personal collections, video segments, video documents). Other features support the user for easy access to personalised video material. This is achieved by maintaining both, the original and the personalised video.

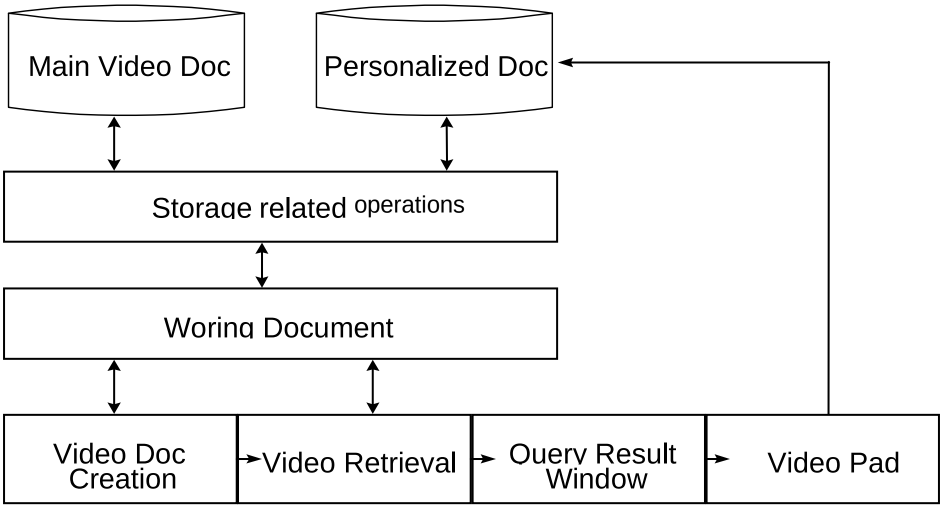


*Fig. A1: Block Diagram of VidIO Interface and Storage*

.

To provide more user access, tools such as Query Result Window (made for displaying user query results and other related information), Video Pad (used in creation, maintenance and re-indexing of personalised video segment, also supports segment browsing, file append and segment addition / deletion) and Document Creation tools (supports construction and maintenance of original video database, and aids in organising personalised video segments) were also added to the system, as shown in figure A1. [3]

VidIO's data model is hierarchical and revolves around the three data elements, i.e. structural components, contents in text form and the data catalogue. The video segment is divided into four main components, which are frames, shots, scenes and program. The hierarchy for a particular video is also resolved with these components [4]. The software's front end was developed in Visual C++(16 bit) and the back end database is Microsoft Access ver 2.0. The videos are stored in AVI format *(frame size =240x180, frame rate/sec = 15.0, kilobits/sec data rate =190, compression mode = Microsoft Indeo32)*. The data used for testing this model was the recorded news clips of CNN International.

During the creation of personalised documents, users are required to add their own descriptions, which are stored in the hierarchical manner, as shown in fig 1 above, maintaining linear story structures of original video materials, as they have contextual information that should be preserved to help users to interpret the data during video retrieval. This can be seen for a part of news clip data, stored in hierarchical form, in figure 2. At level 1 is the main review section. Then at level 2 are the sub-sections like Arts & Culture, Political, Crime, etc. Further in level 3 are the subsections of Arts & Culture column like WW2 paintings, Ballet Dispute, Movies of the week, etc. In this way each frame of the news video is captioned with its inheritance and can be very easily retrieved from the main video file. Hence according to figure 2, the link for the *councillor's speech* part in the main video document will be :

*news_reviews . arts&culture . ww2_booty_paintings . councillor's_speech*

The major problem with VidIO is about its video description input. No tools were developed or incorporated for auto indexing of contents. Again no provision is provided for the recognition of a particular object in a video scene or frame. Thirdly major attention was given only to the video part of the data and no consideration was given to the audio support, which was emphasised by Merlino [5], who suggests that audio information can play a major part for content indexing especially in videos such as news bulletins, archives or documentaries, where there is always an anchor person giving some important information in the background. The author further suggests that some textual footings and the presenter's voice can become excellent keys for content indexing.

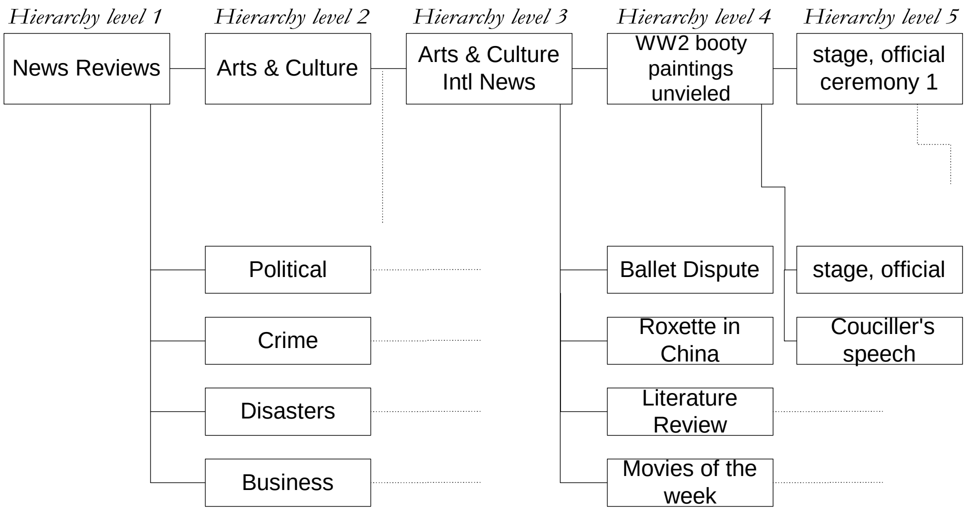


*Fig 2.VidIO hierarchical description structure for an original video document*

Along with this project, the MMRG at Southampton is actively involved in MAVIS. The MAVIS project is a programme of research to develop Multimedia Architectures for Video, Image and Sound. Modular approach is used and different modules are responsible for all the processing associated with a particular media-based feature type and, as new feature types are introduced, associated matching techniques are developed and added to the main engine. For example, to make use of the added richness which digital video presents, modules are being developed which understand the temporal nature of the video and which can extract combined spatial and temporal features. This will be further used in multimedia thesaurus (MMT) and intelligent agent support for content-based retrieval and navigation. The earlier MAVIS 1 project was concerned with enhanced handling of images and digital video sequences in multimedia information systems. The project will extend the Microcosm architecture to support the MMT, in which representations of objects in different media are maintained together with their inter-relationships. Intelligent agents will be developed to classify and cluster information from the MMT, as well as additional knowledge extracted during information authoring and indexing, to seek out discriminators between object classes and also naturally-occurring groupings of media-based features and to accelerate media-based navigation. [6]

### *A1.1.2 Vane*

The Multimedia Group at University of Boston has developed video database Vane [7] with the property to encompass multiple applications or video document domains. Using SGML[1] and TCL/TK[2] they developed a semi-automatic annotation process which extracts the content and semantic value of a raw video stream. However this system still require a human annotator who expedites the canonical information as metadata within the application domain.

The data model for Vane supports three types of indexing structural, content based , and bibliographic. The structural metadata is again divided into media specific data and cinematography data. The media specific data indexes information about recording rate, compression format and resolution of frame, whereas the cinematic index is about the creation of video, specific information, title, date and camera motion. The other end the content based information is again divided into two main categories i.e. information about tangible (physical shapes appearing in video) objects and information about conceptual (events, concepts, actions etc.) entities.

The developers of Vane are of the view that three levels of hierarchy i.e. sequences, scenes and shots are enough for most straight forward generic decomposition. Additional layers will yield excessive fragmentation and excessive computation but do not provide significant knowledge [7,8].

In the model the user can also have multiple annotations of the same video segment, by going a step ahead in hierarchy but the implementation of this idea makes the model very complex and it is not clearly stated that how these multiple annotations are stored in the data model.

In order to compensate for this problem they have provided additional dynamic metadata definitions, as attributes to the unique identifier DTD (data type definition). These are category definitions, attribute order and attribute definitions.

Overall this tool is designed to construct large and useful video documents and was tested on some news archives and was found to be quite successful. Still this tool is in its evolution stage and its usability on complex videos like documentaries or commercial programmes is not known. Secondly an annotator with prior domain knowledge is also required to identify the start and end of a segment, which is quite cumbersome. Regarding its model the additional metadata definitions seem useless if human annotator does all the segmentation.

[1] Standard General Mark up Language

[2] Tool Command Language / Toolkit for X-windows

### *A1.1.3 A. Caetano & N. Guimaraes (1998)*

Arthur Caetano and Nuno Guimaraes defines shot as a sequence of one or more frames that were contiguously recorded and if sequenced represent a continuous action in space and time.

They presented a modelling language for video, which enables the automatic segmentation, labelling and clustering of video data. These clusters of data generate a hierarchy of groups that can represent scenes as well as other types of semantic or logic groups to felicitate browsing and navigation in the contents of the video. N-dimensional vectors represent low level features of the video shots. These vectors contain data of frame rate, frame size, frame priority (for frames), colour, texture, shape (for image / shot) and other related camera operations like tilting, zooming and panning. These vectors (data sets) are stored in R-tree and its variants (R*-tree, -tree) for adequate indexing.

The scenes, frames and shots are stored as multimedia objects (MO) which is in fact the basic unit of information in their data model. Again the media object is defined as

$$MO= (O_{id}, O_{type}\ F, R)$$

where $O_{id}$ is the unique object identifier,

$O_{type}$ shows the type of object and

F is the feature set associated with the ojbect i.e $F= \{fi\}$ and $fi = \{r_{i1}, \ldots\ldots., r_{in}\} \supset R$.

Each feature f*i* is composed of one or more representation, for example, the colors in an object can be representated by histogram, moments, color vectors etc. These features aids in creating hierarchies of data at different levels. The Query is then processed by retrieving the k-most similar ojbects to a set of feature vectors that designate the query’s content. Work is also in progress on developing tools for creating algorithms on MPEG and MJPEG compressed video clips.

## *A1.2 Object Oriented Data Models*

Since multimedia objects possess both temporal and spatial features, OODBMS (Object Oriented Data Base Management System) are considered to be the best option for defining multimedia models [10,11].

Gibbs defines an AV (audio-video ) object as

> "An AV value, v is a finite sequence, $v_i$, of digital audio or digital video data elements" [12].

Further he suggests that,

> "Each AV value has a media data type governing the encoding and interpretation of its elements. The type of v determines $r_v$, the data rate of v."

These definitions clearly state that the media objects define themselves, their own data rate and other attributes, which is a likely behaviour of objects defined in object-oriented environments. Oomato and Tanaka [22] have also worked on this idea. They developed the Object Oriented Video Information Database (OVID), and introduced the term 'video object'. Traditional databases are unable to support video data models using multimedia objects for the following three reasons. First, AV values often have a very long duration (i.e. of minutes or hours). Secondly, it may not be possible to allow concurrent use of AV objects (as is the case of relational models) and finally system resources (buffers, processor cycles, bus bandwidth etc ) can only be allotted in a limited quantity to a process to handle a tuple.

Another major problem with conventional databases is the concurrent access to AV data, as it requires explicit scheduling of different processes in the system, by different clients. Since AV sequences are also temporal in nature, the database system should also be responsible for co-ordinating the presentation of these sources to different clients at different time. Traditional databases usually solve this problem by providing a temporary copy of the particular tuple to each client and then save the last modified copy to the master database. As the AV object is of a very large size (in our case, a single object can be greater than 500 Mega bytes !!), the idea of providing individual copy to all the clients becomes impossible.

The above points convinced the researchers about the usefulness of object-oriented data models, particularly for AV data models. A lot of work has been done in this regard, which as presented below:

### *A1.2.1 S. Gibbs & C. Breiteneder (1995)*

Gibbs and Breiteneder define several abstractions that provide higher-level descriptions of AV values. The first is *Quality factor,* which takes care of compression ratio and frame size. Second abstraction is of *Temporal Entity Type,* which is specified in the form of **N[**T, {Q}, {D}**]** , where N is the name of entity type, T is underlying data type, Q shows the quality factor (additional) and D is its discrete time co-ordinate system[13]. For example in the expression

Video Entity[JPEG video Type, 480*640*8, $D_{30}$]

JPEG shows the compression format, 480*640 is for frame rate, 8 is for 8 bits per pixel, and $D_{30}$ shows the data rate of 30 frames per second.

Further a number of relationships can be derived and instantiated with the above mentioned entity, solving the problem of concurrency. Useful derivation relations can be of the type like Translation (the start times of a value are uniformly displaced), Concatenation (a value is added to the end of another value), Selection (a sub-value is extracted) and Conversion (a value is re-sampled and used at some other location).

The Object-oriented approach is also useful in request scheduling. For instance, read and write requests are scheduled by the concurrency control subsystem and disk accesses are scheduled by the storage subsystem. With videos, these areas are particularly important and should be visible at the database interface and under application control of the user. For example, it should be possible to request ‘play video X now’ or ‘play video X after Y seconds’ [14]. This facility will not only aid in attractive features but will also help is scheduling resources and  thedatabase itself, thus providing concurrent access to media data.

Gibbs provides a class for categorising video objects as,

```
class VideoValue subclass-of Media Value
{       int width
                int height
                int depth
                int numFrame
                Image Value frame[numFrame]
```

```
            }
    class AudioValue subclass-of Media Value
            {       int numChannel
                    int depth
                    int numSample
                    int sample[numChannel][numSample]
            }
```

The above classes provide very little information about the media objects. i.e. image is considered as a binary object, however more information about image is required. Further annotation and signatures for images, scenes and sequences are also missing. Since these classes provide the basic structure of a video and were produced in early 90's, a lot of improvements in these classes were made by other researchers in later years.

### *A1.2.2 L. Huang & W. Xiong (1995)*

Xiong and Huang divide AV databases into two categories, depending on their functionality. The first category consists of those databases which concentrate on retrieving an image/frame according to one or more specific features. The second category of systems work mainly on retrieving requested information from a single image. In some systems, these two functionalities are combined together [15].

They further state that the main problems with the conventional OODB modelling is that a class has to be defined statically and objects of a class must be homogeneous in nature, which cannot be applied to AV objects which are inter-related, adhoc, tentative and evolving [16].

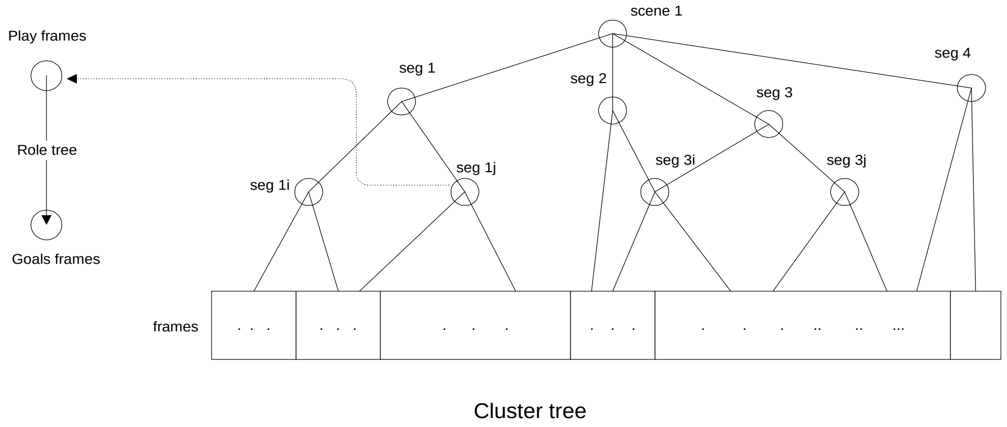


Fig 3 Example of a video programe

A Conceptual Clustering Mechanism (CCM) is defined as $C_t$ = <*A*,*M*,*X*>, where A is a set of cluster attributes, M is a set of cluster methods and X is the set of role-player associates. X is also the set of roles and objects that play the role within the cluster. Coming the properties of CCM, they are dynamically created and deleted and can manage adhoc dynamics, but cannot create or delete objects [16].

In the indexing mode, video classification decomposes the data into semantically meaningful segments and CCMs, as shown in fig 3. It should be noted that it is not necessary for *segXi* to inherit all the properties of *segX,* but do inherits some of the basic properties to remain in hierarchy. In this way a cluster tree is formed , which in fact is a binary tree and can be easily used for indexing any video scene.

There are many issues which are not addressed in this model for example storage problems are not discussed and parsing of arbitrary arguments (messages) from one object to another is not very flexible. Again this model is still in its infancy and has not been tested on real applications.

### *A1.2.3 MPIL*

Jain and Gupta [17] came up with their model "Multiple Perspective Interactive Video (MPIL)" which can generate three dimensional voxels (information vectors) data sets from multiple streams of video data, which is manipulated to be refined and configured.. The schema designer then chooses a set of primitives to model this data which in fact is now a specific class definition. Subsequent objects similar to this data will become the instances of this class. A set of additional features are also assigned to this object which aids in indexing the class and its instances to use later in query operations.

The special attributes of AV objects introduced in this model are name (a string composed of two vectors, the first is the identification and the second vector defines the geometric properties of the primitive i.e. it can be of the shape *square.mesh.cylinder* or *simplex.mesh.cylinder)*, parameters (define the external values set by the user at the time of data definition), create, delete and display. Along with, other attributes like cut, copy, paste, deform, compare are also added for user facilities.

The most common operations for a query in any AV model are union, difference and set-membership of point sets in feature space. More specific operations applicable in MPIL are Find Boundary (returning hyperpolyhyderal boundary of the points in image), Select by Spatial Constraint, Select by distance and k-nearest neighbour.

Jain further states that all these classes, functions and operations merge to form a Video Data Modelling Language (VDML) which is destined to support description and manipulation for AV data. In addition they are also developing an implementation of a set of shape representations, stepping ahead beyond the current paradigm of query-by-visual-example, describing the new research prototype of image-space visualisation environment [19].

## A1.3 Video Algebra

Weiss and Duda proposed an algebraic data model that defines a video stream by recursively applying a set of algebraic operations on the raw video segment[20]. The basic object of a video algebra model is presentation. A presentation is a multi-window spatial, temporal and content combination of video segments. Presentations are described by video expressions, which are constructed from raw video segments using video algebraic operations. These operations are Creation(create, delay), Composition (concatenation, union, intersection, etc.), Output (window) and description (content attributes that can be attached to video expressions).

Segments are specified using the name of the raw video and a range within the raw video. The recursive nature of the video structure is supported by creating compound video expressions from simpler ones using video algebra operations. The model also allows nested stratification, i.e. overlapping logical video segments are used to provide multiple coexisting views and annotations for the same raw video data. Users can search video collections with queries by specifying the desired attributes of video expressions. The result of the query is a set of video expressions that can be played back, reused, or manipulated by a user. In addition to the content-based access, algebraic video also allows video browsing.

This model was tested on a collection of a pre-indexed TV broadcast news, commercials and movie trailers. Along with indexing, closed captioned text (signatures) were also provided to the Video Algebra System. The video and file server was run on Sun Sparc Station, the model was run on a Silicon Graphics machine and a third different machine with HTTP server was also connected to them via Ethernet [21]. Users were then logged on through the network and edited and composed the algebraic video nodes , the performance was found quite good. This system is still under construction and tools like database support, auto indexing etc. are in the process of development.

## *A1.4 The VISION Digital Video Library*

The VISION (Video Indexing for Searching over Networks) digital video library prototype was developed at the Telecommunications an Information Sciences Laboratory of University of Kansas as a test bed for evaluating automatic and comprehensive mechanisms for library creation and content-based search and retrieval of video across networks with a wide range of bandwidths.

Gauch and Li [36] came up with the idea of an on-line digital video library, which provide content-based search. The major feature of this project is an integrated application, which provide video processing, information retrieval, speech extraction and word-spotting feature in real time.

First the full-motion video is captured in AVI format with JPEG compression, then segmented by using two-step algorithm based on video and audio contents. Then a closed caption decoder is used to extract textual information to index video clips. Finally all the information is stored textually for content-based exploration of the video library over the network.

A novel algorithm is used to segment the video. First the video is broken into different shots by the traditional image-based segmentation methods. These shots are then post-processed to merge some contiguous segments back together. This is done by analysing audio features extracted from speech signals such as endpoint detection and speaker identification.

The endpoint detection algorithm is based on measurement of the audio signal short-time energy and zero-crossing rate and end of an utterance. The short-time energy function of speech is computed by splitting the speech signal into 'N' samples and computing the total squared value of the signal in each sample. Further the energy of the speech is generally greater than that of silence or background noise. A speech threshold is then determined by taking into account the silence energy and the peak energy. Zero-crossing rate is the measure of the number of times in a given time interval that the speech signal amplitude passes through a value of zero [37].

CNN news clips (headline news footage) were used to test this prototype and a successful result was achieved, but some major problems were also incurred. The major problem was that the contiguous clips which were separated by video techniques such as zoom-in/out, fade-in/out, or pan-in/out were all separately segmented. This system also failed where there was no sound between two successive clips, as  there is no way to detect a change on the short-time energy of the audio signal.

This system is still in its prototype level and work is still carried on to modify this project.

## A1.5 Others

Hjesvold [23] has introduced an architecture to support video information sharing and reuse, specially when running video presentations, annotating video materials and video information extraction. To identify video shots, audio recordings, presentations and annotations, he incorporated two models. First a content model for basic components in a video and secondly a structure model for composite components.

The content model is responsible for scene composition. It also defines attributes, operations and content descriptions for the video basic components. The structure model is used for scene editing. It also defines attributes, operations and semantics for the composite components. The attributes include Object-Identity, Type, Object Architecture and a Textual Signature. VCR type operations such as playing, pausing, stopping, enhanced operations like creating and modifying components, and administrative operations like reading and changing the values of the attributes symbolic name, type, coding rate, quality factor, synopsis, etc. are also provided [24]. Semantics for modelling a narrative presentation are supported by five story entities: scene, sequence, descriptive story, interlaced story and concurrent story.

This model is heavily dependent on metadata i.e. data about data at every semantic node and an experienced annotator is required to provide this data, but is a very good example to share and re-use the same video among different video applications and users.

Smoliar and Zhang [25] proposed a frame based (knowledge base) model that represent a hierarchical organisation of video nodes. A frame contains textual information for classes and instances. A class frame holds important information for maintaining the hierarchy such as super class, sub class and instance, while the instance frame holds a pointer to its upper level frame and the actual content representation, description, video and other related data.

# APPENDIX 2

## A2.1 MPEG Overview

The Moving Picture Coding Experts Group (MPEG) is a working group of ISO/IEC, in charge of the development of international standards for compression, decompression, processing and coded representation of moving pictures, audio and their combination.

The purpose of MPEG is to produce standards. The first two standards produced by MPEG were:

MPEG-1, as standard for storage and retrieval of moving pictures and audio on storage media (officially designated as ISO/IEC 111172, in 5 parts)

MPEG-2, a standard for digital television (officially designated as ISO/IEC 13818, in 9 parts).

MPEG-4 Version 1, a standard for multimedia applications, that has officially reach the status of International Standard in February 1999, with the ISO number 14496.

Since July 1993 MPEG is working on its third standard, called MPEG-4. MPEG considers of vital importance to define and maintain, without slippage, a work plan. This is the MPEG-4 Version 1 work plan:

| Part | Title | Working Draft | Committee Draft | Final Committee Draft | Draft for Intl. Standard | Internat-ional Standard |
|---|---|---|---|---|---|---|
| 1 | Systems | | 11/97 | 03/98 | 10/98 | 02/99 |
| 2 | Visual | | 11/97 | 03/98 | 10/98 | 02/99 |
| 3 | Audio | | 11/97 | 03/98 | 10/98 | 02/99 |
| 4 | Conformance Testing | 10/97 | 12/98 | 03/98 | 03/99 | 05/99 |
| 5 | Reference Software | | 11/97 | 03/98 | 03/99 | 05/99 |
| 6 | Delivery MM Integration Framework | 07/97 | 11/97 | 03/98 | 10/98 | 02/99 |

MPEG-4 is building on the proven success of three fields: digital television, interactive graphics applications (synthetic content) and interactive multimedia (world wide web, distribution of access to content) and will provide the standardised technological elements enabling the integration of the production, distribution an content access paradigms in the above mentioned fields.

## A2.2 MPEG-4

MPEG 4 is an ISO/IEC standard whose formal designation will be ISO/IEC 14496, is to be released in November 1998 and will be an International standard by January 1999 [30].

In the beginning of the work, the objective of the new standard was to address very low bit rates (<1Mbits/sec) but its target was considerably modified in order to take the changes in the audio visual environment into account. To avoid the emergence of a multitude of proprietary non inter working AV formats and players, this standard is also supposed to standardise some of the elements of digital television, interactive graphics (synthetic content), and the world wide web domains. The targeted applications are, internet multimedia, interactive video games, interpersonal communications (video conferencing, video phones etc.), interactive storage media (optical disks, etc.), multimedia mailing, networked database systems (ATMs), remote emergency systems, wireless multimedia and broadcasting applications [29,30].

The MPEG-4 standard addresses the coded representation of both natural and synthetic (computer generated) audio and visual objects. MPEG-4 System was developed to provide the necessary facilities for specifying how such objects can be composed together in an MPEG-4 terminal[3] to from complete scenes, how a user can interact with the content, as well as how the data streams should be multiplexed for transmission or storage.

MPEG-4 will enable the production of content that has far greater reusability, has greater flexibility than is possible today with individual technologies such as digital television, animated graphics, World Wide Web (WWW) pages and their extensions

It will offer transparent information which will be interpreted and translated into the appropriate native signalling messages of each network with the help of relevant standard bodies

[3] The term terminal is used here in a generic sense, and includes computer programmes hosted on general purpose computers.

In all, this standard is built to accommodate both client-server as well as mass storage-based playback scenarios. In the client-server mode, the server transmits multiplexed streams containing compressed AV objects (in MPEG jargon known as Audio Visual Objects, AVOs) and the associated scene description. At the client end, these streams are de-multiplexed and the resulting objects are decompressed, composed according to the scene description. Interaction information can be processed locally, or transmitted to the sender. It is also possible to use MPEG-4 content locally i.e. using a hard disk, CD-ROM or a DVD.

### *A2.2.1 Functionality of MPEG-4*

This standard supports eight key functionalities, that can be grouped into three classes:

***Content Based Interactivity***

- Object based multimedia tools
- Object based bitstream manipulation and editing, *by providing means for editing a video object.*
- Object based random access, *by providing efficient methods to access objects at any point in the bitstream within a limited time and fine resolution*
- Hybrid natural and synthetic data coding, *by providing syntactic elements and tools to allow coding of synthetic data together with video data, and mixing and synchronisation of these streams.*

***Compression***

- Imported video compression efficiency, *by providing a subjective video quality that is better than the quality achieved by similar or emerging standard in similar conditions*

***Universal Access***

- Robustness to Information errors and loss, *by providing tools to achieve object based error protection for a variety of wired and wireless networks and storage media, with possibly severe error conditions*
- Object resolution scalability, *by providing tools and syntactic elements to achieve spatial and temporal scalability with a fine granularity in terms of content and quality.*
- Object Scalability, *by providing the ability to add or drop video objects from a coded scene.*

There are two official MPEG implementations of the video codec : one in C provided by European project ACTS-MOMUSYS, and one in C++ provided by Microsoft which are currently under testing and are evolving from integrating the changes in their descriptions[33].

### *A2.2.2 Scene Description in MPEG-4 System*

The scene description part of MPEG-4 systems specification describes a format for transmitting the spatio-temporal positioning information that describes how individual audio-visual objects are composed within a scene. It also includes information pertaining to user interaction. This information is constructed in a hierarchical manner which can be represented as a tree. The leaf nodes of such tree  are always AV objects. The other nodes perform grouping, transformations, etc. The hierarchical structure simplifies scene construction as well as content editing. The structure and scene description capabilities borrow several concepts from VRML, and includes all of the VRML nodes as well as some MPEG-4 specific nodes. In particular, MPEG-4 defines additional nodes that are suitable for purely two dimensional environments.

### *A2.2.3 Representations of AV objects*

MPEG-4 standardises a number of primitive AVOs in a scene, which are capable of representing both natural and synthetic content types, which can be either 2D or 3D in nature. For a particular scene, for example a news clip, the AVOs will be organised in an hierarchical fashion. At the leaves of the hierarchy the following primitives AVOs will be formed :

- A 2-D fixed background
- The picture of a talking person
- The voice associated with that person
- Some other background picture or other components of the background set, etc.

In addition MPEG-4 also defines the coded representation of objects such as

- Text and graphics
- Talking heads and associated text to be used at the receiver's end to synthesise the speech and animate the head
- Animated human bodies

Then these AVOs are grouped according to their object descriptions. As an example, the visual object corresponding to the news caster and his/her voice are tied together to form a new compound AVO, containing both the aural and visual components of a talking person.

Such grouping enables the construction of complex scenes, and enables users to manipulate meaningful objects. Other operations possible on AVOs are, to place AVOs anywhere in a given co-ordinate system, apply streamed data to AVOs, in order to modify their attributes and to interactively change the user's viewing and listening points anywhere in the scene.

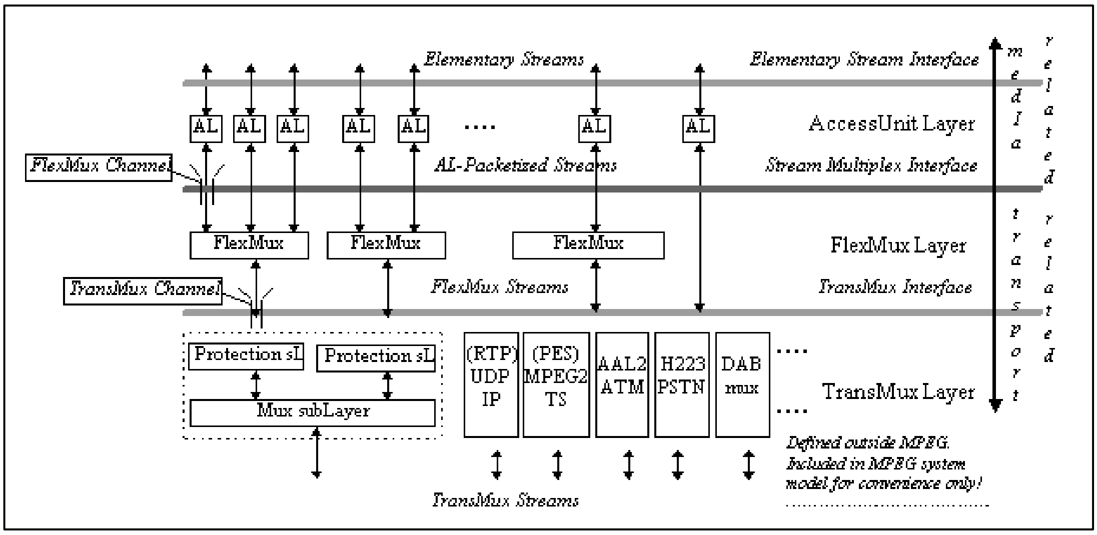


*Fig A2.1, The MPEG-4 System Layer Model*

### A2.2.4 System Layer Model

The AVOs are transferred in one or more elementary streams. These steams are then prioritised by the Quality of Service (QoS) they request for transmission (i.e. data rate), as well as other parameters, including stream type information to determine the required decoder resources and the precision for encoding timing information. This is all done through an access unit layer and a conceptual two-level multiplexer, shown in figure A2.1. The access unit layer allows identification of access units (e.g. video or audio frames, scene description) in elementary streams, recovery of the AV object's or the scene description's time base and enables synchronisation between them. The FlexMux (flexible multiplexing) groups AVOs in elementary streams and puts a low multiplexing overhead. Finally the TransMux (Transport Multiplex) unit offers transporting services matching the requested QoS [29]. It should be noted that the use of FlexMux is optional and this layer may be bypassed if the underlying TransMux instance provided equivalent functionality. Hence in this way it is possible to indicate the required QoS for each elementary stream and FlexMux stream, translate such QoS requirements into actual network resources and convey the mapping of elementary streams, associated to AVOs, to FlexMux and TransMux channels.

### *A2.2.5 Scene Description Unit*

At the decoding side (i.e. at client) streams coming from the network (or a storage device) as TransMux streams are de-multiplexed into FlexMux streams and passed to appropriate FlexMux de-multiplexers that retrieve elementary streams. The elementary streams (ES) are parsed to the appropriate decoders. Decoding recovers the data in an AV object form and reconstruct the original AV object ready for rendering on the appropriate device. The reconstructed object is made available to the composition layer for potential use during scene rendering. Decoded AVOs, along with scene description information, are used to compose the scene, as shown in figure A2.2. It should also be noted that scene descriptors are coded independently from streams related to primitive AV objects [31,32].

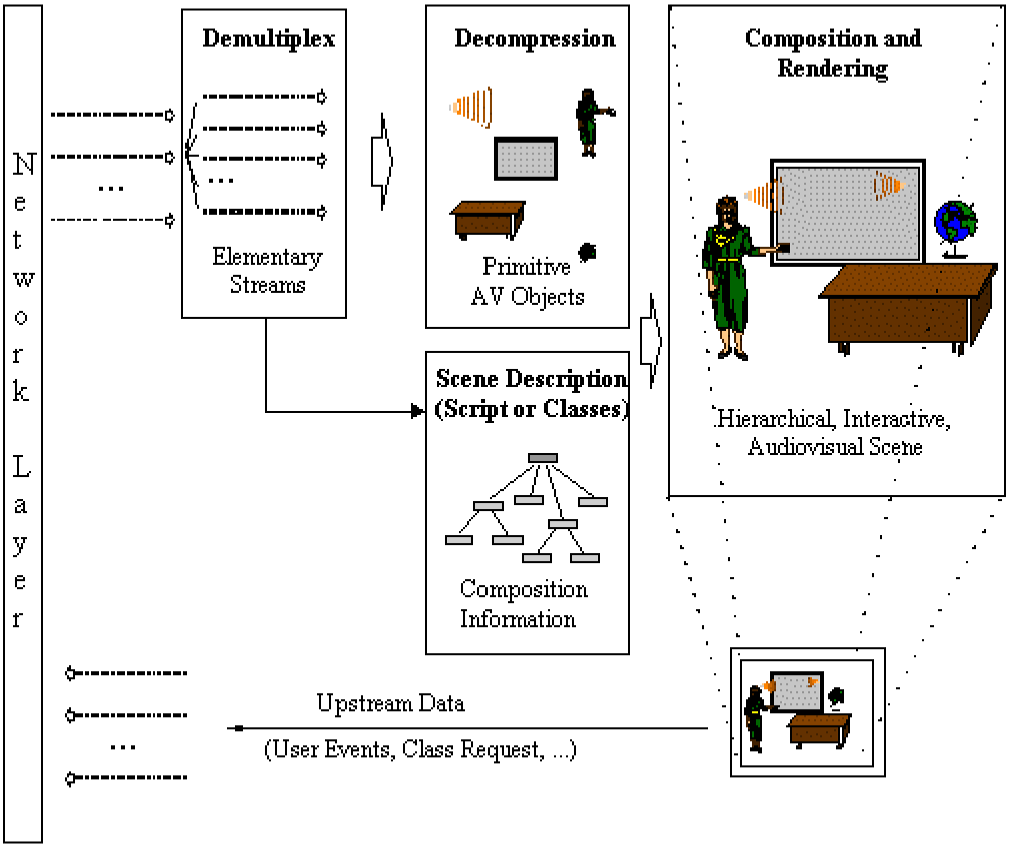

Network Layer
Demultiplex
Elementary Streams
Decompression
Primitive AV Objects
Composition and Rendering
Hierarchical, Interactive, Audiovisual Scene
Scene Description (Script or Classes)
Composition Information
Upstream Data
(User Events, Class Request, ...)

***Fig A2.2: Major Components of an MPEG-4 terminal***

As noted above, an MPEG-4 scene follows a hierarchical structure, as shown in figure A2.3. Each node of the graph is an AV object and can go up to n-hierarchies, depending upon the complexity and grouping of the object. The structure is not necessarily static and the node attributes (positioning parameters) can be changed while nodes can be added, replaced or removed.

As the AV objects have both spatial and temporal extents, each AV object has a local co-ordinate system. A local co-ordinate system for an object is one in which the object has a fixed spatio-temporal location and scale. The local co-ordinate system  serves as a handle for manipulating the AV object in space and time. AV objects are positioned in a scene by specifying a co-ordinate transformation from the object's local co-ordinate system into a global co-ordinate system defined by one parent scene description nodes in the tree[34]. Additionally individual AV objects also expose a set of parameters to the composition layer through which part of their behaviour can be controlled. Examples include pitch of a sound, the colour for a synthetic object, activation or deactivation of enhancement information for scaleable coding, etc.

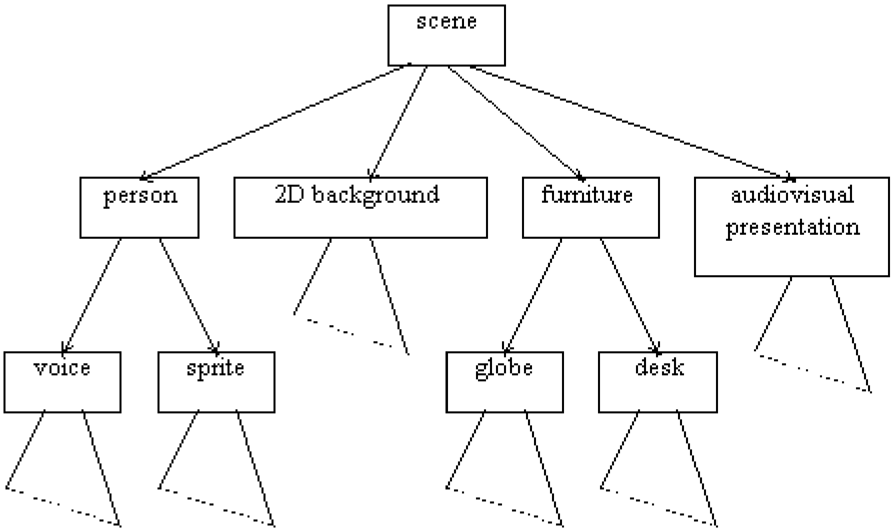


***Figure A2.3:  Logical structure of a Scene***

A variety of features are now offered by the MPEG 4 protocol. To summarise, this standard will offer

- A new kind of interactivity, with dynamic objects rather than static ones
- The integration of natural and synthetic audio and visual material
- The possibility to influence the way AV material is presented
- The simultaneous use of materials coming from different sources
- The integration of real time and non-real time (stored) information in a single presentation

### *A2.2.6 MPEG-4 File Format:*

The MP4 file format is designed to contain the media information of an MPEG-4 presentation in a flexible, extensible format which facilitates interchange, management, editing and presentation of the media. This presentation may be ‘local’ to the system containing the presentation, or may be via network or other stream delivery mechanism (a TransMux). The file format is designed to be independent of any particular TransMux while enabling efficient support for TransMuxes in general. The design is based on QuickTime format from Apple Computer Inc.

The MP4 file format is composed of object-oriented structures called ‘atoms’. Each atom is identified by a unique tag and a length. Most atoms describe a hierarchy of metadata giving information such as index points, duration and pointers to the media data. This collection of atoms is contained in an atom called the ‘movie atom’. The media data itself is located elsewhere; it can be in MP4 file, contained in one or more ‘mdat’ or media data itself is located elsewhere; it can be in the MP4 file, contained in one or more ‘mdat’ or media data atoms, or located outside the MP4 file and referenced via URLs.

The file format is a streamable format, as opposed to a streaming format. That is, the file format does not define an on-the-wire protocol, and is never actually streamed over a transmission medium. Instead, metadata in the file known as ‘hint tracks’ provide instructions, telling a server application how to deliver the media data over a particular TransMux. There can be multiple hint tracks for one presentation, describing how to deliver over various TransMuxes. In this way, the file format facilitates streaming without ever being streamed directly.

The metadata in the file, combined with the flexible storage of media data, allows the MP4 format to support streaming, editing, local playback, and interchange of content, thereby satisfying the requirements for the MPEG-4 inter media format.

### A.2.3 MPEG-7

MPEG-7, also known as "Multimedia Content Description Interface", will extend the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types. MPEG-7 will specify a standard set of descriptors that can be used to describe various types of multimedia information. MPEG-7 will also standardise ways to define other descriptors as well as structures (Description Schemes) for the descriptors and their relationships. MPEG-7 will also standardise a language to specify description schemes, i.e. a Description Definition Language (DDL). AV material that has MPEG-7 data associated with it, can be indexed and searched for. This ‘material’ may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined in a multimedia presentation (‘scenarios’, composition information). Special cases of these general data types may include facial expressions and personal characteristics.

MPEG-7, like the other members of the MPEG family, is a standard representation of audio-visual information satisfying particular requirements. The MPEG-7 standard builds on other (standard) representations such as analogue, PCM, MPEG-1, -2 and-4. One functionality of the standard is to provide references to suitable portions of them. For example, perhaps a shape descriptor used in MPEG-4 is useful in an MPEG-7 context as well, and the same may apply to motion vector fields used in MPEG-1 and MPEG-2.

MPEG-7 descriptors do, however, not depend on the ways the described content is coded or stored. It is possible to attach an MPEG-7 description to an analogue movie or to a picture that is printed on paper. Even though the MPEG-7 description does not depend on the (coded) representation of the material, the standard in a way builds on MPEG-4, which provides the means to encode audio-visual material as objects having certain relations in time (synchronisation) and space (on the screen for video, or in the room for audio). Using MPEG-4 encoding, it will be possible to attach descriptions to elements (objects) *within* the scene, such as audio and visual objects.. MPEG-7 will allow different granularity in its descriptions, offering the possibility to have different levels of discrimination.
Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications.

There are many applications and application domains which will benefit from the MPEG-7 standard. A few application examples are:

- Digital libraries (image catalogue, musical dictionary, etc.)

- Multimedia directory services (e.g. yellow pages)
- Broadcast media selection (radio channel, TV channel, etc.)
- Multimedia editing (personalised electronic news service, media authoring)

# 6.0 References

1. R Hjelsvold, (1994) **Digital Television Archives - Combining Computer Technology and Video**. Presented at the IASA/FIAT Annual Conference 1994, Bogensee, Germany, September 1994.

2. W. Hall, H. Davis, G. Huttchings (1996) **Rethinking Hypermedia: The Microcosm Approach,** Kluwer Academic Publishers, pp 4-9, 148-156.

3. S.Salam, (1996), **VidIO: A model for personalised video information management,** Ph.D. thesis, November 1996, Multimedia Research Group, University of Southampton.

4. S.Salam, W. Hall (1997), **Design and implementation of an experimental video database system for supporting video retrieval from different perspectives,** in proceedings of Storage and Retrieval for Image and Video Databases V, volume 3022, 13-14 February, San Jose California, pp 324 - 339

5. A Merlino, (1997) **Broad cast news navigation system using story segmentation,** ACM Multimedia 97.

6. P.H. Lewis, H.C. Davis, S.R. Griffiths, W.Hall, R.J. Wilkins (1996), **Media based navigation with generic links,** in proceedings of Hypertext 96, ACM.

7. M. Carrer, L. Ligresti, G. Ahanger, T.D.C. Little (1997), **An Annotation Engine for Supporting Video Database Population,** Multimedia Tools and Applications, an International Journal, November 1997, Volume 5, Number 3, Kluwer Academic Publishers

8. Gulrukh Ahanger, Thomos D.C. Little (1997), **A system for customised News Delivery from Video Archives,** MCL Technical Report no 06-06-1997, Multimedia Communications Laboratory, Dept. of Electrical and Computing Engineering, Boston University Boston, USA

9. Arthur Caetano, Nuno Guimaraes, (1998) **A Model for Content Representation of Multimedia Information.** Presented at a workshop "The Challenge of Image Retrieval" organised by the British Computer Society

10. W Klas, EJ Neuhold, M Schrefl, (1990), **Using an Object –Oriented Approach to Model Multimedia Data.** Computer Communications, Vol. 13, No. 4, pp 204-216.

11. B L Yeo, M.M Yeung, (1997), **Retrieving and visualizing Video,** ACM Communications, vol. 40, December 1997, pp 43 - 52

12. Simon Gibbs, Christian Breiteneder, Dennis Tsichritzis, (1993), **Audio Video Data Model, An Object-Oriented Approach**, 9th International Conference on Data Engineering, Proceedings, Sponsored by IEEE USA, ch-74, pp381-390.

13. Simon Gibbs, Christian Breiteneder, Dennis Tsichritzis, (1992), **Modelling of Audio/ Video Data,** Entity-Relation Approach, ER-92, ch-25, pp 323-339.

14. Simon Gibbs, Dennis Tsichritzis, (1995), **Multimedia Programming, Objects, Environments and Frameworks**, Addison Wesley / ACM Press, UK.

15. Huang Liusheng, J.C.Lee, Q.Lee, W.Xiong,(1996), **An Experimental Video Database Management System Based on Advanced Object Oriented Techniques.** ISAT/SPIE symposium on storage and retrieval for image and video databases, Feb 1996, San Jose, USA.

16. W.Xiong, C.M. Lee, M.C Ip, (1995), **Net Comparison: A fast and effective method for classifying Image Sequence.** ISAT/SPIE symposium on storage and retrieval for image and video databases, Feb 1995, San Jose, USA.

17. R Jain, A. Hampapur (1994), **Metadata in Video Databases**, SIGMOD Record, volume 23, No. 4, December 1994, pp 27 -33

18. R. Jain (1993), **NSF Workshop on Visual Information Management systems,** SIGMOD Record, Vol 22, No. 3, September 1993, pp 57-75

19. P.H. Kelly, A. Gupta, R. Jain, (1996), **Visual Computing meets data modelling: Defining objects in multicamera video databases,** in proceedings of Storage and Retrieval for Still Image and Video Database IV, volume 2670, 1-2 February 1996, San Jose, California, pp 120 -131

20. R Weiss, A Duda and D.K Gifford, (1995), **Composition and search with a Video Algebra,** IEEE Multimedia, Spring 1995, pp 13-25.

21. R. Weiss, A. Duda, D. Gifford, (1994), **Content-based access to algebraic video,** in proceedings of IEEE International Conference Multimedia Computing and Systems, Los Alamitos, CA.

22. J. Kanda, K. Wakimoto, H. Abe, S. Tanaka, (1998), **Video hypermedia authoring using automatic object tracking,** in proceedings of Storage and Retrieval for Still Image and Video Database VI, volume 3312, 38-30 January, 1998, San Jose Calefornia, pp 106 – 115.

23. R. Hjelsvold, (1994), **Video Information Contents and Architecture,** in proceedings of the 4th international Conference of Extending Database Technology, Cambridge UK, March 1994, pp 259 - 272

24. R. Hjelsvold, R. Midtstraum, O. Sandsta (1995), **Databases for Video Information Sharing,** in proceedings of SPIE on Storage and Retrieval for Image and Video Databases III, vol 2420, 9-10 February 1995, San Jose California, pp 268-279

25. S. Smoiler, H. Zhang, (1994), **Content based video indexing and retrieval,** IEEE Multimedia, Summer 1994, pp 62 – 72

26. A.K Elmagarmid, H. Jiang, A.A Helal, A. Joshi, M. Ahmed (1997), **Video Database Systems – Issues, Products and Applications**, Kluwer Academic Publishers Dordrecht, Netherlands.

27. R. Hjelsvold, R. Midtstraum (1995), **A temporal foundation of video databases,** in proceedings of the International Workshop on Temporal databases, Zurich, Switzerland, September 1995.

28. R. Hjelsvold, R. Midstraum (1994), **Modelling and Querying Video Data**, in proceedings of the 20th VLDB Conferene, Santiago, Chile, September 1994, pp 686 – 694.

29. International Standard Organisation ISO/IEC JTC1 / SC29/WG11 n1902**, Coding of Moving Pictures and Audio, Information Technology – coding of Audio visual Objects: visual ISO/ IEC 14496-2** (Committee Draft), November 21, 1997

30. International Standard Organisation ISO/IEC JTC1 / SC29/WG11 n1901**, Coding of Moving Pictures and Audio, Information Technology – coding of Audio visual Objects: system ISO/ IEC 14496-2** (Committee Draft), October 11, 1997

31. International Standard Organisation ISO/IEC JTC1 / SC29/WG11, **MPEG-4 requirements**, ISO/IEC JTC1/SC29/WG11N1886, October 1997

32. International Standard Organisation ISO/IEC JTC1 / SC29/WG11, **MPEG-4 Frequently asked Questions**, ISO/IEC JTC1/SC29/WG11N, July 1997

33. International Standard Organisation ISO/IEC JTC1 / SC29/WG11, **Overview of the MPEG-4 Version 1 Standard**, ISO/IEC JTC1/SC29/WG11N1909, October 1997

34. Thomas Sikora, (1997), **The structure of MPEG-4 video coding algorithm,** Image Processing Department, Heinrich – Hertz –Institut Berlin, November 11, 1997 (http:// wwwam.hhi.de/mpeg-video/papers/sikora/fmpeg4vm.htm)

35. E. Codd, (1970), **A relational model for large shared data banks**, Communications of ACM, vol 13, no. 6, pp. 377-387, July 1970

36. Kok Peng Pua (1993), **Prototyping the VISION Digital Video Library System**, MS Thesis submitted to the Department of Electrical Engineering & Computer Science and the Faculty of Graduate School of the University of Kansas.

37. Susan Gauch, Wei Li, John Gauch (1996), **The VISION Digital Video Library**, ACM Digital Libraries, 1996

38. **Cambridge International Dictionary of English** (1995), Cambridge University Press

39. John Z Li, Iqbal A. Goralwalla, M.Tamer Ozsu, Duane Szafron (1997), **Modelling Video temporal relationships in an object database management system,** Department of Computing Science, University of Alberta, Canada 1997

40. John Z Li, M.Tamer Ozsu (1998), **STARS: A spatial attributes retrieval system for images and videos**, Department of Computing Science, University of Alberta Canada