Modelling and Querying Video Data

Rune Hjelsvold and Roger Midtstraum
Department of Computer Systems and Telematics
Norwegian Institute of Technology
{rhj,roger}@idt.unit.no

Abstract

As video data is penetrating many information systems the need for database support for video data evolves. In this paper we present a generic data model that captures the structure of a video document and that provides a means for indexing a video stream. We also discuss query language features that can take advantage of the proposed model. We have identified basic operators that should be implemented in the query language to support content based queries. The paper also analyses how these operators can be used to provide video data queries. The model has been used as a basis for a television news archive prototype and some experimental results are presented.

1 Introduction

During the last couple of years, hardware and software tools for handling digital video have become widely available. Users can play digital video on most platforms and some platforms allow the users to produce digital video as well. Most current digital video applications do not really take advantage of a digital video data stream, they are more like digitised versions of traditional video cassette recorders - VCRs. One example of this is the Video-On-Demand service - VOD, [A+93] that gives the user on-line access to a digital library of movies. The service allows the users to select for movies by titles, genre, actors or production year. Once a title is selected, the video file is treated as an analogue video cassette and the user is only provided with traditional VCR functionality.

As the amount of digital video data increases, digital video will start penetrating information systems. The users of these systems will not be satisfied with the analogue VCR functionality. Users will require direct access to the video information, for instance to a specific scene within a movie. They also will ask for tools for manipulating, storing and retrieving video data in the same way as they manage numeric and textual data today.

Database management systems (DBMS), are among the most important tools for handling data, especially in a multiuser environment. As video become a common data type in information systems DBMS's should give support to applications of video information. Both users and vendors of DBMS's are are beginning to recognise the needs for video data support and the first commercial products are already available to the market - e.g. the Oracle Media Server [LL94].

Our research is aimed at developing functionality for video data support that can be included in DBMS's. This paper is concerned with two aspects of video data support: 1) video data modelling and 2) query language extensions. The proposed data model captures the important characteristics of video data and provides a framework that can be used by different applications of digital video. The focus for our work has been the support of video data and we have not paid special attention to problems arising when multiple media are integrated, e.g. audio/video synchronisation and multiple audio tracks. Such problems are covered in detail in, for instance, [BGT92].

Previous work on data models for video information can be found in [RD89], [DSP91], [KHT91], [Mer93], [OT93] and [Hje94]. An early proposal was to divide a video document into segments and describe every segment independently, so called "segmentation". This approach has been strongly criticised by Smith [SP91]

Proceedings of the 20th VLDB Conference
Santiago, Chile, 1994

mainly because of its inflexibility. Smith and other researchers at MIT have proposed an alternative approach, named "stratification", where they propose to segment the contextual information rather than segmenting contiguous frames. The stratification approach is discussed in [DSP91], [SP91] and [Smi92].

A similar approach has been chosen for the video object database OVID [OT93]. Video objects in OVID correspond to sets of video frame sequences. Each video object has a set of attributes and a unique identifier. OVID's video data model does not explicitly support modelling of the video document structure. OVID provides the user with the SQL-based query language *VideoSQL* which gives the user the ability to retrieve video objects by specifying some attribute values. VideoSQL does not, however, contain language expressions for specifying temporal relations between video objects.

This paper is organised as follows: Section 2 presents the data model. In Section 3 we have described how this generic model can be tailored to the television news domain and we review the user requirements querying a news archive. Section 4 discusses in what way video querying differs from traditional querying and important query language features are identified. Section 5 presents the results from a project where librarians from Norwegian Broadcasting Corporation (NRK) participated. Section 6 concludes the work and summarises areas for further work.

2 A Generic Video Data Model

A database system managing video information should provide database support for a diverse range of applications [Hje94]. Ideally, the same database should be available for several different purposes such as video production, scientific analysis of video material - for instance a study of the use of narrative techniques in early movies by Hitchcock, and video selection by end-users. One advantage of a common database for these different purposes is that information added to the database during production can be made available for scientific analysis and video selection. Thus, valuable information from the production of video documents could be available to other applications.

To facilitate sharing of the video information itself and the information describing its content and structure the database system has to provide a common data model for video. This generic model need not be an ideal model for every application. The main purpose of the model is to be "a least common denominator". Some applications may expand the common data model with specific concepts while still using the core concepts while less demanding applications may use only a subset.

To describe - e.g. index, contents which are not intimately related to structural components we have adopted ideas from the stratification approach in [SP91]. The stratification approach, whilst strong on free annotations, has ignored the need for structure as a tool to navigate and comprehend large volumes of video data. I our generic model a segmentation approach is used to define the video document structure which can provide well defined levels of abstraction.

The generic data model is given in Figure 1 using the enhanced-ER notation from [EN94]. The intention has been to make the following possible within the same framework:

1. Structuring of video material,
2. free annotations of video material and
3. sharing and reuse of video data,

Video data is stored in a video database as congiguous groups of frames called **`StoredVideoSegments`**. A **`VideoDocument`** is represented by a **`VideoStream`** which is mapped to one or more **`StoredVideo-Segments`**. Because of the latter mapping a **`Video-Document`** can be considered as a continuous stream of frames even when it physically is composed of several different **`StoredVideoSegments`**.

An important and flexible video information unit is the **`FrameSequence`** which is an interval of subsequent frames from a video document. One single **`FrameSequence`** can represent any sequence of video frames ranging from one single video frame to an entire video document. The **`FrameSequence`** plays a central role in our data model as a means to connect both structural units and thematic annotations to the video material. A **`FrameSequence`** is defined by a **`PartOf`** relationship to a **`VideoStream`** and a (first frame, last frame) pair referencing the co-ordinate system of the **`VideoStream`**.

The structure of a **`VideoDocument`** is represented by a hierarchy of **`StructuralComponents`**. Each **`StructuralComponent`** identifies a **`FrameSequence`** which consists of the frames that belong to the component. Section 2.1 presents the structure hierarchy in more details.

Thematic indexing is supported by **`Annotations`**. Each **`Annotation`** identifies a **`FrameSequences`** and gives a textual description of its contents which are searchable. Section 2.2 describes the indexing mechanisms in more details.

In the following subsections we will discuss the capabilities of this model.

2.1 Structure

In a video document there are two inherent levels of abstraction, the entire video document and the indi-

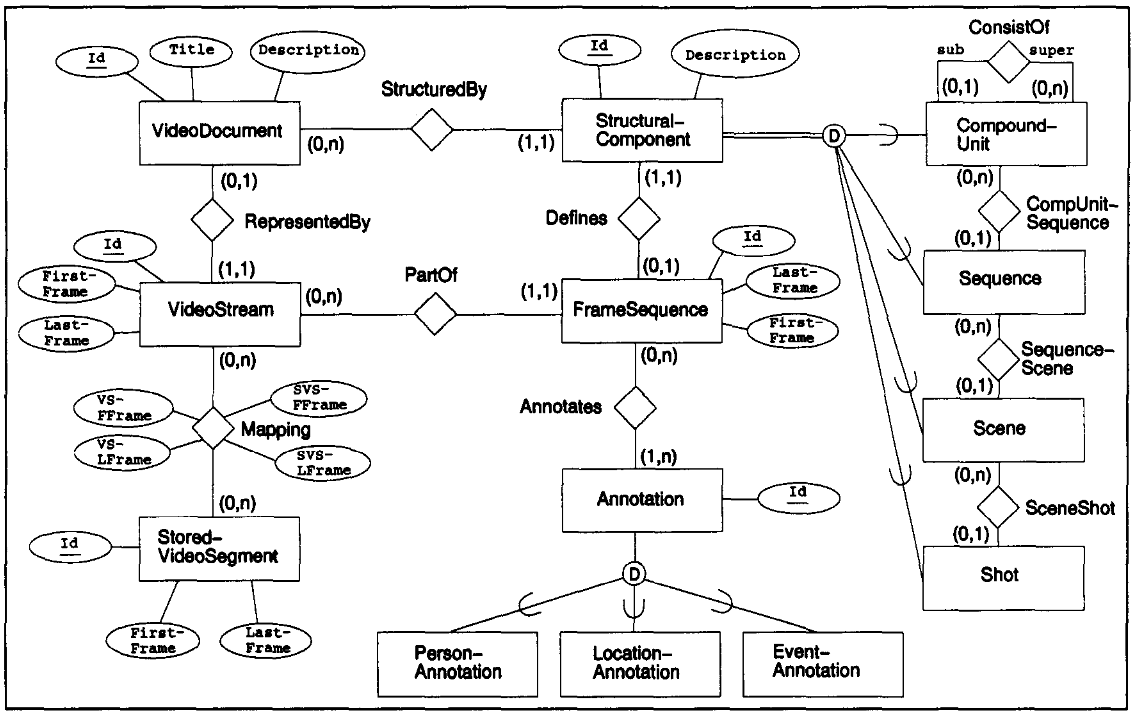

Figure 1: A common data model for video information using an enhanced-ER notation. (Only a few attributes are shown in the diagram.)

vidual frames. For most applications the entire video document is too coarse a level of abstraction. A single frame, on the other hand, is rarely the unit of interest. This is because a single frame spans a very short interval of time and because there are so many individual frames even in a quite short video document (the European video standard, PAL, results for instance in 25 frames per second). [RD89] and [KHT91] strongly emphasize the need for some sort of structuring method.

From experiments with a television news archive [Mer93] we have learned that abstractions such as *scenes* and *news items* makes it easier for the user to make references to video information and easier to comprehend its contents. Other researchers have shown that video information browsing is difficult (see [Ste91]). Our experiments show that structural abstractions give valuable support to video browsers.

The structure part of our model is inspired from film theory [Mon81] and work based on segmentation of video material. It is built around the concept of a **StructuralComponent** which has an associated **FrameSequence** of video material. The concept of **StructuralComponent** is specialised into the **CompoundUnit**, **Sequence**, **Scene** and **Shot** subclasses and a hierarchical relationship is defined between the different subclasses.

The semantics of the subclasses are best explained bottom-up starting with the **Shot** which can be considered as the basic structural unit. As defined in [DSP91] a **Shot** consists of one or more frames generated and recorded contiguously, representing a continuous action in time and space. **Shots** which are related in time and space are assembled in a **Scene** and a number of scenes which together give a meaning are put together in a **Sequence**. Related **Sequences** are assembled into a **CompoundUnit**. **CompoundUnits** can be put together to form recursive **CompoundUnits** in an arbitrary number of levels.

The proposed model is not necessarily the best one for every video document but is rather a common framework for structuring video information in video databases. Our model provides a generic framework that can be tailored to specific application domains. In Section 3.1 we show how the model can be tailored to the television news domain.

2.2 Thematic Indexing

The structure of a video document captures some aspects of the video material but is not suited as a representation of every characteristic of the material. As discussed in [Smi92] and [Hje94] it should be possible to make detailed descriptions of the content of the video material which are not necessarily directly linked to structural components but more often to arbitrary frame sequences. One simple example could be the need to identify every frame sequence within

Casablanca where Ingrid Bergman can be seen.

In our model one can annotate an arbitrary sequence of frames from a **VideoStream** by defining the **FrameSequence** of interest and establishing an **Annotates** relationship between this and the relevant **Annotation**. These annotations are independent from any structural components. Thus, the thematic indexes complement the structural description of **VideoDocuments**.

In Figure 1 we have shown a specialisation of the **Annotation** entity into the **PersonAnnotation**, **LocationAnnotation** and **EventAnnotation** subclasses. The types and semantics of annotations have to be defined in the context of the applications. A set of general annotation types, however, (see the subclasses shown) should be provided by the generic model while allowing the applications to augment the descriptive power with domain specific annotation types. Different types of annotations are more thoroughly discussed in [Hje94] and [Mer93].

2.3 Sharing and Reuse of Video Material

As recognised in [MD89] the same basic video material may be used in several different video documents. In our model we have catered for this by defining the video contents of a video document as a logical concept, the **VideoStream**, and defined the connections to (physically) stored material by a set of mappings onto **StoredVideoSegments**. This allows multiple use of the same stored video material in ways that are well defined and represented in the database.

Since different uses of stored video material is explicitly represented, the database has complete knowledge of the use of any part of the stored video material. In addition, this provides the possibility that annotations and structural information related to one **VideoStream** can be accessible in the context of another **VideoStream** which (in part) use the same stored video material. This sharing of content descriptions is further discussed in Section 4.3.

2.4 Video Data Type

To sufficiently support digital video the DBMS has to provide a video data type. If otherwise, the video information would not be a first class citizen of the database system which would lead to a number of serious problems. The database support of video data should be of the same quality as expected for applications using traditional data types.

In our framework the video data type will be used for storing the **StoredVideoSegment** entities. If this video data type is not provided by the DBMS, one has either to use a *binary large object* - BLOB, data type or to use a reference to an external file storing the video information.

Neither the BLOB nor the file solutions are satisfactory. A **StoredVideoSegment** group related frames into a manageable unit but it should not be regarded as an atomic unit. Both the DBMS and applications need the ability to identify, retrieve and use smaller components. As a capability provided by the DBMS this can not be catered for by neither a BLOB based nor a file based solution. The file based solution brings along the additional problem of managing data that is not fully under control of the DBMS. It will usually be more difficult to maintain the consistency of the system and in some cases impossible to provide necessary access restrictions.

A video data type must possess a number of desired properties:

1. A number of related frames has to be grouped together in the data type to form manageable information units, while still providing access to individual frames and continuous frame intervals within the unit. As a consequence of this, video data will not satisfy the first normal form requirements defined for conventional relational databases.

2. Format information has to be stored in addition to the video data. Examples of these are the number of pixels in each frame; aspect ratio; colour space and compression method. Without this kind of information it will be almost impossible to interpret and use the video information.

3. A video information unit has to be played back at a given rate. The DBMS should support the play operation by delivering video frames to the user application gradually to avoid immediate transfer of huge amounts of video data into the user space.

3 Television News Archives

This section is based on the experiences made during a project where Norwegian Broadcasting Corporation (NRK) participated. The goal was to investigate how digital video could provide better mechanisms for storing and retrieving video information. The librarians at NRK currently describe the video contents in a free-text database but there is no direct linkage between this text-based database and the video information. Therefore, search for specific video segments from the archive is time-consuming and often requires specific knowledge of the archive.

Even though the television archive is managing NRK's complete video production, special attention has been devoted to television news. This is because news information is the most reused information in the archive and because news reporters are the group of television reporters that use the archive and existing video information most frequently.

3.1 Television News Data Model

When the generic data model was tailored to describe television news the following interpretations were applied:

- `VideoDocument`: This top-level entity represents one complete news broadcast (e.g. the evening news a specific day).
- `CompoundComponent`: The news consists of one single, non-recursive `CompoundComponent` - i.e. there is a one-to-one relationship between the `VideoDocument` and the `CompoundComponent`.
- `Sequence`: This entity represents a complete news item - i.e. the frame sequence starting with the studio reporter introducing the news item and ending after the last report within the same item.
- `Scene`: This entity represents a frame sequence within a sequence where the video has been recorded at the same time and same place (usually a news report).
- `PersonAnnotation`: This entity attaches information about persons (e.g. name and profession) to frame sequences within the news.
- **LocationAnnotation:** This entity attaches location information to frame sequences within the news.
- **EventAnnotation:** This entity attaches keywords describing events, situations and objects to frame sequences within the news.

3.2 Types of Queries

It is beyond the scope of this paper to give a complete description of the user's requirements. Therefore, we have chosen to describe five common tasks where database support is required and will use these for further discussions on query language features.

Type 1 - Structure Browsing: The users of the television news archive request a browsing tool which allows them to browse through a "table of contents" using a top-down approach. From a list of news sorted by date and time, the users want to expand one (or more) items to have a view of all items that it (they) contains. Further expansions should be possible until the complete structure is shown. The browser should also work in the opposite direction, i.e. from a given video stream it should retrieve all components making use of this stream.

Type 2 - Clip List Generation: A clip list gives a complete description of how a video document is composed. Each item in the sequential list identifies a shot, a reference to the stored video segment it has been derived from and the start and end frames of the stored video segment.

Type 3 - Contents Report Generation: In their background research the television reporters want to make sure that they are aware of all relevant information related to video information that they intend to reuse. From a given frame sequence the generator should retrieve all annotations applicable to it.

Type 4 - Content Queries: The most important function for a television news archive is to allow the users to retrieve (and play) video from the archive by giving a specification of the contents. If, for instance, a news reporter is making a news report on Gro Harlem Brundtland - the Norwegian prime minister, going to visit John Major - the British prime minister, for discussions of the Norwegian application for joining the European Union (EU) the reporter may want to retrieve scenes from previous meetings between the two. The query processor should retrieve video components where annotations fulfilling the user request apply.

Type 5 - Complex Content Queries: The previous task describes one common situation for reuse where there is a strong thematic relation between the original use and the reuse of the video. In other cases there may not be such a strong relation. Scenes which have had a specific purpose in the original news report, may be reused because they give a good description of a more general situation. Consider the case where a historian wants to analyse if the Norwegian mass media has been using environmental crimes as an argument in the EU discussion. Could there be, for instance shots from reports on illegal gathering of eggs from protected birds used in a news report on EU?

4 What's new with Video Data Querying?

The generic model proposed in this paper is capable of representing both the structure of a video document and allow thematic indexing. A video query language should include features that can take advantage of this complex data model. The following discussion will be based on the user requirements presented in Section 3.

4.1 Query Results

The result of a relational database query is a table (relation) of tuples satisfying the query. In embedded SQL, each of the tuples can be accessed via a cursor mechanism. The result from type 1, 2 and 3 queries

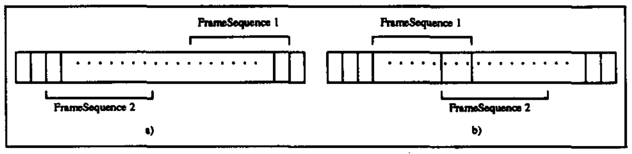

Figure 2: Frame sequences without (a) and with (b) temporal overlap

(described in Section 3.2) can be accessed via similar mechanisms and do not put forward new requirements.

Type 4 and 5 queries will retrieve a set of frame sequences. These queries should return a set or an ordered list of object identifiers to the user. The user can retrieve further information by using these identifiers. The query processor must take into consideration the case where some of the retrieved frame sequences may not have unique identifiers in advance (e.g. the overlapping part of two distinct sequences may not have a unique identifier).

4.2 Temporal Operators

The conditional part of type 1 and 2 queries can be expressed in a traditional query language like SQL. The remaining types of queries, however, cannot easily be expressed in SQL because these queries imply operations on frame sequences. For instance, to retrieve a frame sequences where two annotations are both present the query processor has to check that there is a temporal overlap between the frame sequences representing the annotations. Figure 2 shows a video stream where two frame sequences are non-overlapping (a) and where two frame sequences overlap (b).

A frame sequence is essentially a temporal interval with a temporal relation between the frames. Temporal interval operators are therefore applicable to frame sequences. In [LG93] the following temporal interval operators are discussed:

- *A Equals B*: Returns *true* if the two sequences *A* and *B* are identical.
- *A Before B*: Returns *true* if *A* happens before *B*. (The complementary operator is *After*.)
- *A Meets B*: Returns *true* if *B* starts with the next frame after *A* has ended. (The complementary operator is *MetBy*.)
- *A Overlaps B*: Returns *true* if *B* starts while *A* is still active. (The complementary operator is *OverlappedBy*.)
- *A Contains B*: Returns *true* if *B* starts after *A* and ends before *A*. (The complementary operator is *During*.)

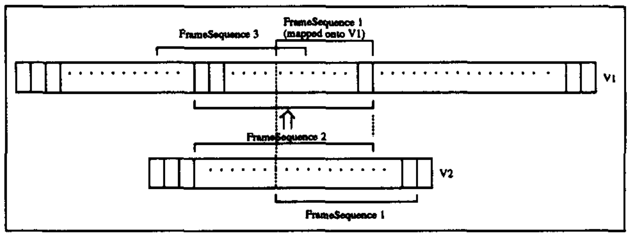

Figure 3: Frame sequence 3 overlaps frame sequence 1 because frame sequence 2 is reused

- *A Starts B*: Returns *true* if *A* and *B* start with the same frame and *A* ends before *B*. (The complementary operator is *StartedBy*.)
- *A Finishes B*: Returns *true* if *B* starts before *A* and *A* and *B* end with the same frame. (The complementary operator is *FinishedBy*.)

By using these operators the user is able to formulate queries that put constraints on the temporal relationship between frame sequences (and their corresponding Annotations).

4.3 Frame Sequence Mapping

It may seem that only frame sequences related to the same video streams can overlap. Unfortunately, the situation is more complex because one video stream can map onto another. Figure 3 shows an example of a video stream mapping. In this case video stream V2 represent a stored video segment. Frame sequence 1 is related to this video stream. Video stream V1 reuses parts of the video stream (frame sequence 2). In the figure it is shown that frame sequence 1 is mapped onto video stream V1 and thus is overlapped by frame sequence 3.

This mapping will be common in cases where some annotations are related to the contents of the stored video segments - i.e. the location for the recording and persons involved, while other annotations are related to the edited video stream - i.e. keywords describing the news item. The annotations related to the stored video segment (V2 in the figure) are also valid for all video streams derived from V2.

In the more complex case, one video stream may map indirectly onto another because they both reuse overlapping video segments from the stored video. Figure 4 shows an example where video stream 1 reuses one part from video stream 2 while video stream 3 reuses another part of the same stream. Frame sequence 2 in video stream V1 overlaps frame sequence 1 through this indirect mapping. The indirect mappings add severe complexity to a query and lots of computing are required to process such queries.

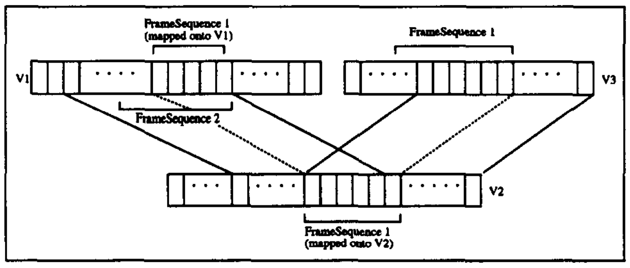

Figure 4: Frame sequence 2 overlaps frame sequence 1 via an indirect mapping

This kind of mapping will be less common than the direct mapping shown in Figure 3. Indirect mappings may not always be relevant and it should be up to the user to decide whether or not such mappings should be performed during query processing.

4.4 Query Processing Steps

In this section we will describe how the temporal operators and the mapping functionality can be used to express the query types 3, 4 and 5 given in Section 3.2. We will not define a query language syntax which incorporates this functionality but only describe the major steps involved in processing the queries:

Type 3: Contents Report Generation

```
1. Identify all FrameSequences that overlap
   the FrameSequence specified by the user.
2. Identify all the FrameSequences that can
   be directly mapped onto the
   FrameSequence specified by the user.
3. Return all Annotations that are related
   to the FrameSequences from step 1 and 2.
```

Type 4: Content Queries

```
1. Find the set of PersonAnnotations where
   name="Gro Harlem Brundtland".
2. Find the set of FrameSequences related
   to the set returned in step 1.
3. Find the set of PersonAnnotations where
   name="John Major".
4. Find the set of FrameSequences related
   to the set returned in step 3.
5. Find the set of EventAnnotations where
   keyword="European Union".
6. Find the set of FrameSequences related
   to the set returned in step 5.
7. Find all Scenes to which the
   FrameSequences in step 6 belong.
8. Delete all Scenes from step 7 which do
   not overlap with at least one
   FrameSequence from step 2 and one from
   step 4.
```

Type 5: Complex Content Queries

```
1. Find the set of EventAnnotations where
   the keywords contain "egg gathering",
   "illegal", and "protected birds".
2. Find the set of FrameSequences related
   to the set returned in step 1.
3. Find the set of EventAnnotations where
   keyword="European Union".
4. Find the set of FrameSequences related
   to the set returned in step 3.
5. Find all Scenes to which the
   FrameSequences in step 4 belong.
6. Return all Scenes from step 5 where
   there is a temporal overlap (via an
   indirect mapping) between some members
   of the sets returned in steps 2 and 4.
```

5 Experimental Results

Annotations may vary from coarse-grained - i.e. annotating a complete news item, to fine-grained annotations covering only short fragments of a shot. The annotations made by the librarians at NRK are mainly coarse - i.e. at the report level, but with some finer grained annotations -usually describing complete shots. Annotations related to only subset of shots are quite rare. Figure 5 shows a news item from the evening news Friday February 18, 1994, including some thematic indexes and a description of the structure.

Within the project we have used the datamodel from Section 2 to model the contents of the evening news. Some interesting statistics for an evening news a typical day is:

- The duration is 35 minutes.
- They consist of 15 to 20 news items on average.
- News items consist of 3 reports (scenes) on average.
- Reports consist of 4 shots on average.
- 25 percent of the shots have been used earlier. (Since these shots generally are very small, most of the time is allocated to new shots.)
- On average 2-3 shots in the edited version will typically map onto the same original recording.
- On average the contents of one news item will be described by:
  - 5 PersonAnnotations,
  - 3 LocationAnnotations and
  - 10 EventAnnotations.

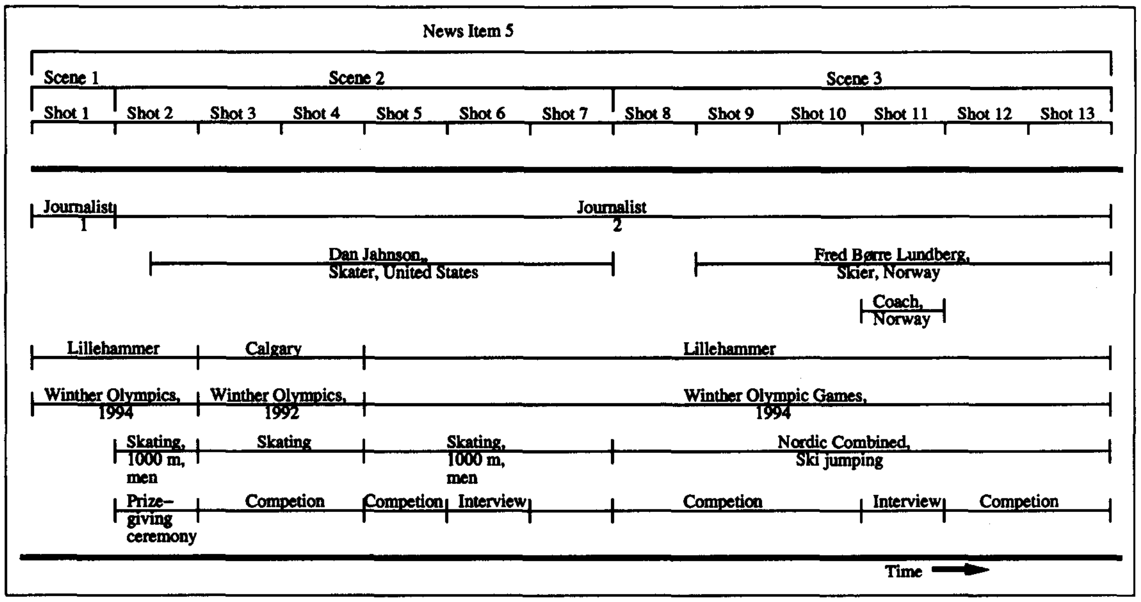

Figure 5: Annotations describing a news item from Lillehammer Olympic Games.

The librarians consider this as a good trade-off between searchability and effort needed to annotate the news items.

Using these typical values one can estimate the size of a database for evening news broadcast during one year. The most interesting figure is the number of unique frame sequences representing structural components or annotations which will be approximately 200,000.

We have implemented a small-scale prototype for a television news archive. The prototype allows the user to browse through the archive structurally (day-by-day, item-by-item, and so on) or perform a content based query. The user can activate a digital video player from the browser/query interface.

6 Conclusion and Further Work

Previous works on video data models have either focussed on video document structures - e.g. [DSP91] or thematic indexes - e.g. [SP91] but not both. In this paper we propose a generic data model which does provide a framework for modelling both the structure and the contents of a video document. The model is generic and it can be tailored to different application domains to adopt domain specific terminology or attributes. We have shown how it can be adapted to a television news domain.

The proposed model is also good for supporting reuse and sharing of video information by separating frame sequences from stored video objects and by relating thematic indexes and structures to frame sequences. The same stored video objects can be used in video documents in different domains where each domain adapts the model to its own specific requirements.

The main weakness of the proposed model is its complexity. Thematic indexes and structural components may implicit relate to each other because different frame sequences may overlap and because frame sequences may be reused. Significant processing is required to explicitly identify these relations.

OVID's video query language VideoSQL [OT93] allows the user to specify video object properties but VideoSQL does not address the temporal relations between video objects. In our work we have studied video query features that allow the user to define temporal relationships between frame sequences. We have also identified the need to map frame sequences during query processing.

Principles discussed in this paper have been tested in experiments together with NRK. The broadcasting company has appreciated the possibilities for combining structural and descriptive view of video information and for having direct linkage between the structural/descriptive data and the video itself. The users have also contributed in the work to estimate the amounts of meta data that can be expected in a television news database.

More research remains to be done in this area. The framework should be extended with more advanced audio capabilities - i.e. supporting different audio tracks to the same video stream. The work with a video query language should be completed - i.e. a formal query language syntax should be defined and tested by real end-users on a query processor prototype.

References

[A$^+$93] Allen et al. VCTV: A Video-On-Demand Market Test. *AT&T Technical Journal*, January/February 1993.

[BGT92] C. Breiteneder, S. Gibbs, and D. Tsichritzis. Modelling of Audio/Video Data. In *Proceedings of the 11th International Conference on the Entity-Relationship Approach*, Karlsruhe, Germany, October 7-9 1992.

[DSP91] G. Davenport, T.A. Smith, and N. Pincever. Cinematic Primitives for Multimedia. *IEEE Computer Graphics & Applications*, July 1991.

[EN94] R. Elmasri and S.B. Navathe. *Fundamentals of Database Systems.* The Benjamin/Cummings Publishing Company, 2nd edition, 1994.

[Hje94] R. Hjelsvold. Video Information Contents and Architecture. In *Proceedings of the 4th International Conference on Extending Database Technology*, Cambridge, UK, March 28-31 1994.

[KHT91] W. Kameyama, T. Hanamura, and H. Tominaga. A Proposal of Multimedia Document Architecture and Video Document Architecture. In *Proceedings of ICC '91 - The International Conference on Communication Conference Record*, Denver, USA, 1991.

[LG93] T.D.C. Little and A. Ghafoor. Interval-Based Conceptual Models for Time-Dependent Multimedia Data. *IEEE Transactions on Knowledge and Data Engineering*, 5(4), 1993.

[LL94] A. Laursen and B. Linder. Delivering Real-time Audio/Video with the Oracle Media Server. In *Proceedings of the EOUG Oracle User Forum 94*, Maastricth, Netherlands, April 17-20 1994.

[MD89] W.E. Mackay and G. Davenport. Virtual Video Editing In Interactive Multimedia Applications. *Communications of the ACM*, 32(7), 1989.

[Mer93] P. Merok. Data Models for Digital Film and Video Archives. Master's thesis, Norwegian Institute of Technology, 1993. In Norwegian.

[Mon81] J. Monaco. *How to Read a Film. The Art, Technology, Language, History and Theory of Film and Media.* Oxford University Press, 1981.

[OT93] E. Oomoto and K. Tanaka. OVID: Design and Implementation of a Video-Object Database System. *IEEE Transactions on Knowledge and Data Engineering*, 5(4), 1993.

[RD89] B. Rubin and G. Davenport. Structured Content Modeling for Cinematic Information. *SIGCIHI Bulletin*, 21(2), October 1989.

[Smi92] T.G.A. Smith. If You Could See What I Mean... Descriptions of Video in an Anthropologist's Notebook. Master's thesis, MIT, 1992.

[SP91] T.G.A. Smith and N.C. Pincever. Parsing Movies in Context. In *Proceedings of the 1991 Summer USENIX Conference*, Nashville, USA, 1991.

[Ste91] S.M. Stevens. Next Generation Network and Operating System Requirements for Continuous Time Media. In *Proceedings of the Second International Workshop for Network and Operating System Support for Digital Audio and Video*, Heidelberg, Germany, November 1991.