Integration of structured video in a multimedia authoring system

Cécile Roisin, Tien Tran_Thuong, Lionel Villard

OPERA project, INRIA Rhône-Alpes.
655 avenue de l'Europe, 38330 Montbonnot, France.
{Firstname.Lastname}@inrialpes.fr
http://www.inrialpes.fr/opera

Abstract. In this paper, we integrate some results on automatic structuration of video in a multimedia authoring system. For that purpose, we identify the specific video structuration requirements of this kind of application and we propose an XML syntax for the description of the structure, the temporal and spatial organization of video. Finally we describe a prototype application that takes advantage of this structured video format.

1 Introduction

Multimedia applications have to handle heterogeneous objects such as text, video and picture. Among these applications, we can identify applications for *multimedia documents* when the objective is the organization and the management of collections of different media. In multimedia documents, media objects are organized in both spatial and temporal dimensions according to their intrinsic basic attributes (duration of video, size of picture, etc.) and the *composition rules* given by the author.

Until now, applications consider media objects as atomic objects. But new needs appear, such video annotation, video indexing, or more generally multimedia document indexing with more elaborate synchronization scheme, etc. Media objects are not just atomic, but more complex and structured with additional information than their pure data coding. Emerging problems are therefore how to extract, to represent this information and to use it in multimedia applications.

Numerous works exist in the area of image analysis and in video analysis [18], [12]. As an example, the Movi project at Inria [3] contributes to the description of audiovisual objects, mainly on video structuration. Based on image analysis, several algorithms have been developed in order to group images in shots and to detect objects or events in shots.

There is also an important activity in the domain of specification and development of multimedia applications [4], [16]. In our project we have experimented multimedia composition through *Madeus*, an constraint-based authoring and presentation system [10].

In this paper, we propose to integrate some results on structuration of video (presented in section 4) in a multimedia authoring system (see section 5). We show in section 3 that specific needs of multimedia authoring imply to consider not only the structure of the video but also its temporal and spatial organization.

2 Existing video formats

Digital video information is encoded in different formats in order to meet various application processing needs. Video formats can be classified by their abstraction level:

- At the lower level, formats encode "only" the visual content of video. Formats such as Mpeg2 [14], Quicktime3 and AVI are adapted for storage (thanks to compression features), rendering or videoconference.
- At a medium level, we find formats for basic annotation and video editing. With these formats, tools can manage a video as a set of continuous frames on which annotations can be added. It is even possible to access a subset of frames in order to compose them with other types of media such as text or image [1]. The encoding of additional data is generally stored in a proprietary format. Recent formats, such as Quicktime4 and Mpeg4 [14], provide new facilities for inserting more structured information and for simplifying the edition process.
- At the highest level we find formats which store semantics data of videos (not just frames). This is useful for more sophisticated annotation services, for indexing in query systems and for access-by-content applications. Works on these formats are still in progress (such as the future standard Mpeg7[14]).

All the previous formats are video-centered. On the opposite we have formats which homogeneously integrate different kinds of media. They are mainly used in multimedia systems. Except Smil [17] and Mheg [13] which are standards, these formats are completely dependent to the tool in which they have been specified: Macromedia Director [11], CMIFed [16] or Madeus [10]. Moreover, all of them consider each type of media as an atomic piece used for temporal and/or spatial composition.

Our objective is to extend multimedia formats in order to integrate semantics data of video in a similar way than high level video-centered formats.

3 Requirements for structured video format

Multimedia documents are collections of heterogenous objects (such as video, audio, text, picture) organized in both the spatial and the temporal dimensions. Designing such documents is known to be a complex and error prone task, specially because they have to deal with complex temporal information (tasks synchronizations and objects durations).

The composition of an heterogenous set of media (text, video, sound, etc.) can be processed by placing spatial and temporal relations between objects. For instance, specifying that two objects play in sequence is expressed by a *meets* relation between them [10]. In the same way, for spatial dimension, specifying that two objects are left aligned can be done by placing the *left_align* relation.

Until now, we have considered that video (and audio) media were atomic media. But for many applications it is not enough accurate to manage and synchronize such coarse objects. Consider for example the slide show in Fig. 1: on the right of the screen, there is the video of the show and on the left appears the currently presented slide. It is difficult to synchronize these two objects in such way that during the presentation each slide is synchronized with the corresponding video excerpt. If we decompose and structure the video, we will be able to compose *video elements* such as shots or scenes with other media of the document. In the previous example, the temporal composition will express that each slide must be played at the same time that its corresponding scene in the video. This need is slightly different than video annotation [5] or video editing

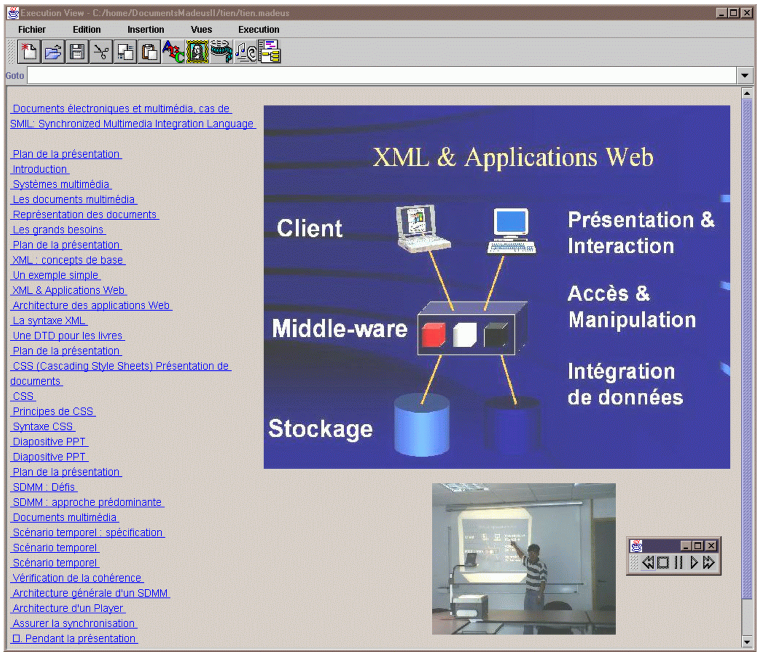

Fig. 1. Slide show multimedia document

such as with the Adobe premiere tool. In fact, a multimedia presentation requires to manage dynamic behaviors due to user interactions and indeterminism due to unstable rendition conditions (load processor and network bandwidth).

Before presenting the format that we have defined for the video, we identify the requirements that such a format must meet in order to allow its use in a multimedia authoring and presentation system. First of all we need to point out and access suitable chunks of video, as provided by its logical organization;

second we want to set temporal relations on these chunks with other media objects of the document, so we need to model temporal and spatial information between video elements in order to ensure global consitensy during composition; finally all these actions must be available through simple and rapid actions.

4 Video structured format

4.1 Basic choices

The choices taken in our video model are mainly motivated by the requirements of the first application we want to realize: we want to compose in space and time fragments of video (the appearance of a character, the beginning of a scene or shot) together with other media objects (text, picture, sound,...) of the multimedia document. Therefore the definition of the video structure must specify the video decomposition in terms of elements together with space and time relations among these video elements.

We choose the markup language XML (eXtensible Markup Language) because it is a language suited to describe structured information and their properties. Moreover, in order to easily integrate video structures in multimedia systems, we choose XML because it is yet used as a description language for multimedia documents such as in Smil [17] and Madeus [10] (on which our prototype has been built, see section 5.2). Even the Expert Group that specifies Mpeg4 envisions the use of XML for Mpeg4 definition [14].

The complete DTD (Document Type Definition) of our video definition can be found in the web site of our project [15].

4.2 High level structuration

Basically, a video is a series of continuous images, which can be organized in a hierarchical structure. Therefore our description format for video will express that organization in terms of the elements that reflect that structure and the relationships among these elements.

Basic entities and relations. In a video structure (Fig. 2), we can identify two categories of elements: temporal elements (Sequence, Scene, Shot, Occurrence, Event,...) and non-temporal elements that identify characters or objects that appear in the video (called Class). The set of time relations used in our video model is a subset of Allen's relations [2]: meets, starts, equal, during, finishes.

As an example, Fig. 4 gives an excerpt of the XML source file of the structuration for the video `Allocine.mpeg`. Notice that the current element is designated by the character "." in the temporal relations. Three types of spatial relations are used in our model: topological relations (Near, At), projective relations (Right, Left, Above, Below, In_front_of, Behind, Beside) and the relation Between. This set could be extended such as in [7].

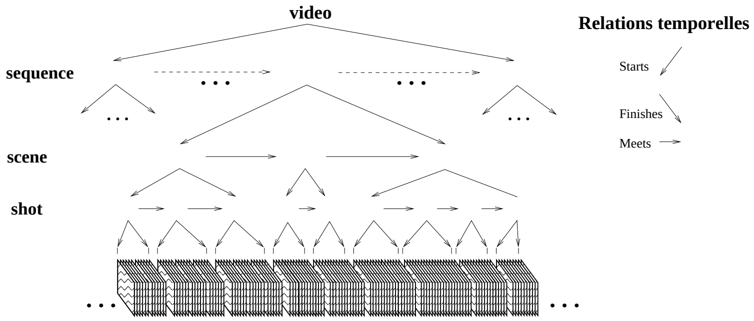

Fig. 2. Basic structure of video

```
<!-Elements of structure video --!>
<!ELEMENT Video (Sequence+,Class*, Rel_Temporal)>
<!ELEMENT Sequence (Scene+,Rel_Temporal)>
<!ELEMENT Class (Instance+)>
<!ELEMENT Scene (Shot+, Rel_Temporal)>
<!-Relations -->
<!ELEMENT Rel_Temporal (Equal | Meets | Finishes | Starts |  During)+>
```

Fig. 3. DTD part of high level video entities

```
<Video Name="allocine’’ FileName="allocine.mpg" NumberImage="902" ...>
   <Sequence  Name="Seq" FirstImage="1" LastImage="902" ...>
     <Scene Name="Scene1"  FirstImage="1"  LastImage="120"  .../>
     <Scene Name="Scene2" FirstImage="121" LastImage="768" ...>
        <Shot Name="Shot4" FirstImage="121" LastImage="173" .../>
        <Shot Name="Shot5" FirstImage="174" LastImage="433" .../>
        <Shot Name="Shot6"  FirstImage="433" LastImage="510"  .../>
        <Rel_Temporal>
          <Starts Interval1 ="."  Interval2 = "Shot4" />
          <Finishes Interval1 ="." Interval2 ="Shot11"  />
          <Meets Intervals = "Shot4 Shot5 Shot6"/>
        </Rel_Temporal>
      </Scene>  ...
   </Sequence>
   <Rel_Temporal> <Equals Interval1="." Interval2="Seq" /> ... </Rel_Temporal>
</Video>
```

Fig. 4. XML structure of the allocine video

Hierarchical structure of video. The basic structure of video consists of three kinds of components, from the smallest to the biggest: shot, scene, and narrative sequence [6]. A *shot* is defined as a series of images, from the moment the camera begins to run to the moment it stops, usually a few seconds. A *scene* is a dramatic unit composed of a series of shots strung together, showing action in one place and at one specific time. A *sequence* is the highest dramatic unit of video, composed of several scenes, all linked together by their emotional and narrative momentum [6].

These elements are organized in the hierarchical structure as follows: a video element contains sequence element; a sequence is composed of scene elements; and a scene includes shot elements. The temporal relations among these elements are defined as follows (see the corresponding XML syntax in Fig. 3):

- sequential relation (*meets*): between two continuous shots of a scene, two continuous scenes of a sequence and two continuous sequences of a video.
- parallel *starts* (resp. *finishes*) relation: between the first (resp. last) shot of a scene and that scene (i.e. its parent in the structure), between the first (resp. last) scene of a sequence and that sequence, between the first (resp. last) sequence and the video.
- parallel *equal* relation: when a shot, scene or sequence is the unique component of a scene, sequence or video.

4.3 Low level structuration: shot description

The description of the former basic components allows to synchronize coarse elements (shots, scenes or sequences) of video in multimedia documents. It can be useful to allow a more fine grained synchronization and for that purpose it is necessary to be able to describe the content of shots such as the occurrence of characters or objects, the transition between shots, etc.

We have defined three types of components in shots: transitions, occurrences of characters or objects and events:

- A *transition* element describes the passage from one shot to the next one. A transition can be: a fade-in, a fade-out, a dissolution or a wipe [6]. As we have chosen to include transitions into shots, we can have meets relations between shots, scenes and sequences (see 4.2).
- An *occurrence* element describes a person or an object that appears in the shot. It refers to the *class* element that corresponds to that person or object. The structure of class elements is defined in section 4.4.
- An *event* element describes a particular situation in the video that is considered as a relevant part such as a motor explosion, a plane taking off, a demonstration, a storm, etc. The event element can also describe a period in a shot in which a spatial property among persons or objects is valid.

These three types of elements are components of shots in the hierarchical structure of the video (see Fig. 6). Events may also belong to a scene when they are spread through several shots. A transition only appears when its shot begins, hence, it is located in time by a relation *starts* with that shot. An occurrence or an event can appear at anytime in the shot in which they are included. Therefore, we use the *during* relation between these elements and that shot. The corresponding part of this description in the DTD is given in Fig. 5.

```
<!ELEMENT Scene (Shot+,Event*,Rel_Temporal?)>
<!ELEMENT Shot (Transition?, Occurrence*, Event*, Rel_Temporal?)>
<!ELEMENT Rel_Temporal (Equals | Meets | Finishes | Starts |  During)+>
```

Fig. 5. DTD part for shot description

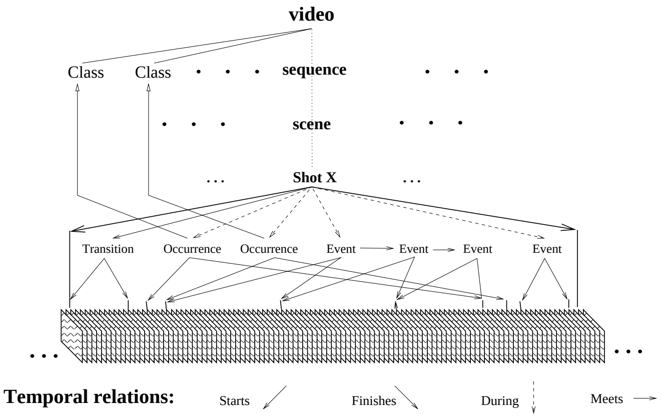

Fig. 6. Structuration of a shot

Spatial relation is a way to define events in shots. It is defined among characters or things that appear at the same time in a shot, such as A stays behind B, A walks on the left side of B, etc. Notice that due to the intrinsic dynamic behavior of the video, spatial relations can change with time. For instance, in the fifth shot of video Allocine.mpeg, there is a Taunus car that follows a Volvo; when the Taunus approaches the Volvo, the Taunus overpasses the Volvo by the left side and then goes ahead the Volvo. To describe these changes, we have to define many periods in the shot corresponding to the different spatial relations among the occurrences. In the former example, we can separate that part of the shot in several periods connected together with a *meets* relation (see Fig. 7).

4.4 Non-temporal elements of video

In order to describe characters or objects in video, we use the *class* element whose content is a list of *instance* elements. The *instances* of a class identify different

```
<Shot Name="Shot5" FirstImage="174" LastImage="433" ...>
  <Transition Name="Trans" Type="Fade-in" FirstImage="174" LastImage="193" ...  />
  <Occurrence Name="Taunus runs" FirstImage="194" LastImage="420" .../>
  <Occurrence Name="Volvo runs" FirstImage="234" LastImage="400" ... />
  <Event Name="Position1" FirstImage="234" LastImage="294" ...>
    <In_front_of-Behind Interval1="Volvo runs" Interval2="Taunus runs"/>  </Event>
  <Event Name="Position2" FirstImage="295" LastImage="332" ...>
    <Near Interval1="Volvo runs" Interval2="Taunus runs"/>    </Event> ...
  <Rel_Temporal>
    <Start Interval1="." Interval2="Trans"/>
    <During Interval1="." Interval2="Position1" Delay="60"/>
    <Meets Intervals="Position1 Position2 ... "/> ...
  </Rel_Temporal>
</Shot>
```

Fig. 7. XML Description of a shot of allocine

occurrences of that class that will appear in shots: each *instance* element points to the corresponding *occurrence* element in the shot. *Class* elements are located at the highest level in the video structure (see Fig. 6). Notice that *class* elements are the only elements in our video structure that are not directly related to time.

5 Implementation

We describe in this section how the video structuration that we have defined in the above section can be used in our multimedia authoring environment Madeus.

5.1 Madeus: a multimedia authoring tool

Madeus is an authoring environment for multimedia documents which meets the following requirements: a high level of expressiveness for both spatial and temporal dimensions; a user-friendly interactive interface; and portability. It differs from other multimedia environments mainly by its constraint-based approach and the tight coupling of presentation services with editing services. Indeed, the authoring services provided in Madeus are based on multiple views [9]: the **presentation view**, the **timeline view** and the **hierarchical view**. The views are strongly linked: any operation performed in a view is passed on the other views in order to ensure data consistency.

5.2 Integration of video structured media

The composition of video entities with other media in a multimedia document is realized in three stages:

– the analysis of the video in order to extract video elements,

- the generation of this resulting information into the XML exchange format as described in section 4,
- and finally the integration of structured video elements in Madeus thanks to editing facilities provided through synchronized views.

The analysis and structuration of the video are processed using a tool named *Videoprep* [3]. This application uses semi-automatic processes to give any video a complete structure. This structuring is obtained in 3 steps: shot cut-out, zone extraction and class definition. For each step, the automatic process can be followed by a checking of the results and by manual editing, using a video player and specific graphic tools. This application uses a proprietary format for storing the video elements that have been detected and structured.

The generation of a XML document is straightforward from the Videoprep format. First, Videoprep data, such as basic components (scenes, shots, ...), numbers of shot frames and spatial positions of occurrences, are directly converted into the XML format. Second, the implicit temporal relations are generated with respect to the rules taken in section 4.

The final step is the composition of video elements with other document elements. This is provided through a new view added in the authoring tool. This view, called the *structured video view*, contains four main parts:

- (1) is the rendering of the selected video element.
- (2) is the description area of the selected video element; for instance, the images that characterize a shot in the selected scene.
- (3) is the visualization of video elements in a hierarchical structure.
- (4) displays quantitative information of the selected video element.

6 Conclusion

In this paper, we have presented a solution for integrating structured video in multimedia authoring tools. We have specified a format that not only describes the logical organization of video elements but also their temporal composition.

The current state of our prototype allows to write a multimedia document that contains video media defined in the format of section 4. It is possible to compose elements of video with other media objects in a time consistent way and to play the resulting document. The structured video view is fully implemented and provides basic functionalities as described in this paper.

Future works evolve towards the modeling of other features of video. One important direction is the spatial organization and the motion of video objects. This work will allow the user to align other media objects (a text for instance) with the position of a video object (for instance a character in a video scene), even if that object moves inside the video. For that purpose, it is necessary to determine spatial relations among video objects thanks to the automatic analysis provided by Videoprep. However the problem is complex because (1) the set of useful spatial relations is user dependent and (2) these relations evolve during time due to moving objects and camera movements. It is clear that manual specification is also required.

Another direction of work is to develop the specification of class elements in the video. In this paper, classes are described as a flat structure (every class element is placed as a child of the root in the XML video structure). However it could be useful to specify a more structured set of classes (with subclasses) that will reflect object classification.

References

1. Adobe,"Premiere",*http://www.adobe.com/products/premiere/*, 1999.
2. J. F. Allen, "Maintaining Knowledge about Temporal intervals", *CACM,* vol. 26, num. 11, pp. 832-843, 1983.
3. Bertolino P., Mohr R., Schmid C., Bouthemy P., Gelgon M., Spindler F., Benayoun S., Bernard H., *Structuring Video Documents for Advanced Interfaces,* INRIA, http://www.inria.fr, to be published, 1999.
4. Buchanam C., Zellweger P.T., "Specifying Temporal Behavior in Hypermedia Documents", *Proc. of the ACM Conf. on Hypertext,* pp. 262-271, December 1992.
5. Carrive J. Pachet F., Ronfard R., "Using Description Logics for indexing Audiovisual Documents", *Int. Workshop on description Logics,* pp. 116-120, 1998.
6. Chandler D., "The 'Grammar' of Television and Film", *UWA,* vol. 1994, http://www.aber.ac.uk/~dgc/gramtv.html.
7. Gapp K.P.,"From Vision to Language: A Cognitive Approach to the Computation of Spatial Relations in 3D Space", *Cognitive Science Program,* vol. Dept. of Computer Science, Universitat des Saarlandes, D-66041 Saarbrucken, Germany.
8. Hammoud R., Chen L., Fontaine D., "An Extensible Spatial-Temporal Model for Semantic Video Segmentation", *First International Forum on Multimedia and Image Processing,* Anchorage, Alaska, May 1998.
9. Jourdan M., Roisin C., Tardif L.,"Multiviews Interfaces for Multimedia Authoring Environments", *Proc. of the 5th Conference on Multimedia Modelling,* Lausanne, 1998.
10. Jourdan M., Layaïda N., Roisin C., Sabry-Ismail L., Tardif L., "Madeus, an Authoring Environment for Interactive Multimedia Documents", *6th ACM Multimedia'98,* Bristol, 12-16 September 1998.
11. Macromedia,"Director", *http://www.macromedia.com/software/director/,* 1999.
12. Meng J.., Chang S.F., "CVEPS - A Compressed Video Editing and Parsing System", *ACM Multimedia'96,* pp. 15-24, ACM Press, Boston, 1996.
13. Meyer-Boudnik T., Effelsberg W.,"MHEG Explained", *IEEE Multimedia Magazine,* vol. 2, num. 1, pp. 26-38, Spring 1995.
14. The MPEG Home Page,*http://drogo.cselt.stet.it/mpeg/*.
15. Opera,*DTD for video,* Inria, http://www.inrialpes.fr/opera/dtdvideo.txt, 1999.
16. Rossum G., Jansen J., Mullender K., Bulterman D., "CMIFed : A presentation Environment for Portable Hypermedia Documents", *Proc. of the ACM Multimedia Conf.,* California, 1993.
17. W3C Recommendation,"Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", *http://www.w3.org/TR/REC-smil,* 15 june 1998.
18. Zhang H. J., Low C. Y., Smoliar S. W., Wu J. H., "Video Parsing, Retrieval and Browsong: An Integrated and Content-Based Solution",*ACM Multimedia'95,* pp. 15-24, ACM Press, San Francisco, 1995.