A MULTIMEDIA MODEL BASED ON STRUCTURED MEDIA AND SUB-ELEMENTS FOR COMPLEX MULTIMEDIA AUTHORING AND PRESENTATION

TIEN TRAN_THUONG[1] and CECILE ROISIN[1, 2]

[1]*Opéra Project, INRIA Rhône-Alpes, Zirst - 655 avenue de l'Europe - Montbonnot – 38334 Saint Ismier Cedex - FRANCE*
[2]*Université Pierre Mendès-France 38000 Grenoble – France*
{tien.tran_thuong@inrialpes.fr, cecile.roisin@inrialpes.fr}

Declarative definition of multimedia presentation such as provided by SMIL standard can be considered as the most significant advance in the multimedia integration domain. However, the requirements of a model that could express richer scenarios for presentations are always a challenge. The work presented here proposes an extended model based on the concept of structured media and sub-elements that allows a finer granularity and a more semantic specification of significant events and locations inside media fragments. The new media fragments can be composed in multimedia scenarios through the specification of temporal, spatial and spatio-temporal relations. Moreover, we propose an abstract animation model that can be combined with the intra-media temporal structuration to specify animation effects in a flexible, no redundant and easy to maintain way. The underlying model is the Madeus model that is a flexible model based on the structural, interval, region and relative constraints. This paper describes the sub-element and the abstract animation models and shows how they are implemented in the Madeus multimedia framework. A complete schema of the video authoring tool based on this model is also specified. Thanks to that, the fine-grained authoring of multimedia scenarios with video media can be done in an easy and effective way.

Key words: multimedia document, multimedia model, media content description, animation, multimedia authoring, multimedia synchronization.

1 Introduction

Multimedia presentation refers to the harmonious integration of presentation of several media (such as text, image, audio, video) into a communication evolving in both space and time axes. Until recently, a majority of multimedia presentations is realized as CD-ROM based multimedia applications. This type of multimedia application has expensive development cost. Moreover the user cannot change neither the scenario nor the content of such multimedia presentation. Today, the emergence of standard multimedia language models, such as HyTime, MHEG, SMIL, MPEG-7, allows authors to declaratively define multimedia documents playable over multimedia browsers and exchangeable between authors. This has made important influences in the development of multimedia applications. In fact, thanks to these multimedia language models, the development of multimedia presentations is much cheaper and easier. Moreover these models allow to define multimedia presentations that can be easily changed to adapt to end-users and end-resources.

With this recent approach - declarative approach, a multimedia presentation is divided into two parts: the multimedia document and the application tool, namely the authoring and/or presentation system. These two parts are related through a multimedia language model. This relation expresses that: the multimedia document is composed by an author through an authoring tool in accordance with the multimedia language model, and then a presentation tool that supports this model can play out the document. By this way, the declarative approach provides more facilities and flexibilities than the code-based approaches. However, the declarative approach currently has yet limitations. One limitation is that current multimedia models do not allow authors to express the fine-grained constraints between media sub-elements, such as, a video character, video shot, video scene, an audio segment, a region on a picture, or a word of a text, etc. Then the author does not have the capability - or has to work hard - to create complex scenarios that require fine-grained synchronizations for multimedia presentations. For instance, to create a Karaoke presentation, the author has to find out about temporal information (begin and end of audio fragments) from the singing to temporally schedule the animating color on each word or part of word of song's text. In fact, the author uses the temporal information extracted from the audio to manually define the schedule of the text animation. As these two media are scheduled independently (there is no explicit constraint among them), there are then no guaranties that the resulting presentation will provide fine-grained

synchronizations between text animation and audio segments: It is worth noting that rendering audio media can introduce delays that can break down the presentation specified in that way. Moreover, the extraction and absolute determination of the temporal information are both very hard to do and due to the absolute time specification, it is difficult to maintain the coherence of the multimedia presentation, for instance when the author modifies the start time of the audio. In other cases, the author cannot define hyperlinks on media's sub-elements like video character, or spatio-temporal synchronizations within it.

The main contribution of this paper is to define a multimedia language model based on the structured-media and sub-elements that gives capacities and facilities for developing complex scenarios of multimedia presentation.

The harmonious combination of media is art creative of author. This creation is unlimited. However, the author is limited at the multimedia language level. He cannot express all his ideas through existing multimedia language model. This work aims at providing an easier and more efficient way to realize complex multimedia presentations. It allows:

- to define more abstract animations and more flexibility in applying them on media;
- to create hyperlinks on media's sub-elements as video character;
- and to make fine-grained synchronizations in all spatial, temporal and spatio-temporal axes.

The animation considered in our work is not autonomous media objects such as an animating image. The animation is modeled as a function changing the presented value of a specific attribute of a media object over time. It gives dynamic effects on media, by example, changing the font size or the color presentation of a text; deforming or moving an image; up or down volume of an audio; or changing the rate display of a video. Therefore, it makes the presentation much more exciting. By this approach, the animation has the same semantic as the SMIL Animation [0]. However, our model proposes a more abstract level of specification. It means that the animation has to be defined just one time, and then it can be applied to animate several media objects. In addition, the author can temporally schedule the application of animation on a media by either absolute or relative way, i.e. it can be synchronized with the presentation of other media objects. For instance, the color animation on a word of song's text will be started when the singing of corresponding word is going to take place.

Media always contain a set of sub-information. This sub-information has to be organized in such way that global information of media is better transmitting to user. This organization is called the *structure* of the media. The multimedia presentation will be much more exciting, and authoring such presentation will be also easier if language models allow author to access into the media structure level. Then, the created multimedia presentation can be sophisticatedly interacted on, by example, pointing to a video character or a video object for asking information about them; or the complex presentation can be easily realized, by example, the *Karaoke* presentation in which the user not only can see the song text fragments colored follow the tune, but also he/she could return to any where in the song by interaction on the text fragments. For that purpose, we propose the *structured media* concept that allows describing internal structure of media beside its data flow. Therefore, media sub-information can be accessible to compose/present with other media object. A more discuss about the relationship between the media coding and indexing, and multimedia composition will be given in section 2.

In our approach, the underlying model for multimedia specification is based on logical, interval and region structures that do not have the capacity to express this internal representation of media. With the *sub-element* concept, this work provides a solution to specify the presentation of media's sub-information in all the multimedia axes through the definition of *subdefActor*, *subInterval* and *subRegion* elements.

The rest of this paper is organized as follows: section 2 discuses about the role of media indexing in multimedia composition. Section 3 presents related work in the domain of multimedia modeling and section 4 presents some examples that illustrate the needs for a new model to cover richer multimedia presentations. Section 5 briefly describes the main features of the structured and interval-based model called Madeus on which we have built our extensions for media structuration. In section 6 we introduce our abstract animation model and the section 7 the structure media model. We focus in this section on the structured video model and we describe a complete schema for the authoring multimedia presentations with these structured media types. Section 8 discuses our concept of *sub-elements* based model that allows harmonious integration the abstract animation and structured media model with the multimedia model and illustrates these extensions through the application of that model in the Madeus multimedia language. Finally, the current achievement of our work and some perspectives are given in the last section.

2 From indexing media to composing multimedia document

Media indexing aims to make easier and effective retrieval of needed media, media fragment or media information from the multimedia database. Although, having such a general sense, the media indexing until now is mainly used for the retrieval domain. In the other side, the multimedia composition needs the retrieval of appropriated media or media fragment and then bases on the media information to compose these media into a multimedia presentation. It only uses the media indexing thus as the retrieval engine for the appropriated media. The retrieved media for composing are then considered as black boxes that are very hard to compose inside the sophisticated presentation, such as, the *Karaoke* presentation. While a further use of the media indexing could help the multimedia composition to avoid the non-trivial problems such as absolute specification and lack of media structure information. Using media indexing could also provide the multimedia composition with the power tools to deeply access into the media structure for fine-grained synchronization composition. This not only supplies facilities for composing, but also allows to create more logical structure presentations (in rapport with absolute composition and traditional media encoding as MPEG-1, MPEG-2).

The main problems preventing the media indexing used in compositing are the lacks of indexing standards dedicating to multimedia composition. A relevant media indexing for multimedia composition has to describe the temporal, spatial and both spatio-temporal structures of media content, instead of only focusing on meta-data or feature description. The segment description scheme tools of MPEG-7 standard (that include all of *StillRegion DS*, *VideoSegment DS*, *AudioSegment DS*, *AuioVisualSegment DS*, *MultimediaSegment DS*, etc.) could respond to this requirement. However the main objective of MPEG-7 is oriented toward archiving and searching, therefore these tools have to be refined for becoming more relevant to multimedia composition. Additionally, one more important need is to have a multimedia composition model and environment to support indexed media.

The work in [0] has tried to address these problems by the integration of all analyzing, indexing and composing in a multimedia authoring environment. However, the systems remain in the specific domain (*Interactive Electronic Technical Manuals*). Moreover, the system is often closed, i.e., it cannot use existing indexed media from external systems. Another work described in [0] provides a sophisticated composition model for video-centered hypermedia document. It provides a video structuration model for composition; unfortunately, it does not describe how this video structuration is performed. More works about this subject are discussed in the next section. We believe that the construction of a fundamental architecture for using indexed media in multimedia composition is an important research issue. This is one of the objectives of the work presented here.

In the other hand, emerging standards such as SVG, SMIL and MPEG-4 define new formats that aim at integrating different media in a presentation. These new formats provide a more explicit definition of the logical structure of the content that is more machine-understandable than the traditional media encoding formats (MPEG-1, MPEG-2, JPEG, etc.). So they naturally better support the index and search engines. Moreover, search engines can exploit the synchronization expressed between media to enhance their results, for instance searching images can be done with the search on the texts synchronized.

3 Related works

In this section, we discuss several works, focusing in three majors modules required to specify a sophisticated multimedia presentation and how our work differs from them. Note that crossing these three modules we can see the works progressed from multimedia indexing to multimedia composing.

3.1 Multimedia content description

Multimedia content description is an approach to index electronic resources. It provides not only a more automatically managing amount of electronic resources that is getting higher and higher but also easiness and flexibility in the use of electronic resources. Nowadays, a number of works have considered the application of DC (Dub-

lin Core) [0], RDF (W3C - Resource Description Framework) and MPEG-7 (Multimedia Content Description Interface) [0] to multimedia content description.

The Dublin Core metadata standard is a simple effective element set for generating metadata for describing a wide range of information resources. The Dublin Core standard currently comprises fifteen elements (Dublin Core Metadata Elements Set), the semantics of which have been established through consensus by an international, cross-disciplinary group. Dublin Core Qualifiers provides a proper way to refine these standard elements for specialized local or domain-specific needs. In [0], a schema for Dublin Core-based video metadata representation has been introduced by J. Hunter & L. Armstrong. The representation of Dublin Core could be encoded in several different syntaxes. The most common and easily understood example is encoding of Dublin Core in HTML format. More complex applications of Dublin Core are harmonization of Dublin Core with other metadata models such as mixed using of Dublin Core with other metadata vocabularies in RDF metadata or harmonization of MPEG-7 with Dublin Core [0][0] for multimedia content description.

RDF is a framework for metadata. Its broad goal is to provide a general mechanism being suitable for describing information about any domain. The foundation of RDF is a model for representing named properties and property values. The RDF model is much like an object-oriented system that comprises a collection of class organized in a hierarchy and offers extensibility through subclass refinement. The representation of RDF in XML format makes XML, a flexible, extensible and application-independent format to represent structured data, possible to specify semantics for data. That will able to give well-defined meaning for Web information being not only *machine-readable* but also *machine-understandable*, which is the main principle of the Web semantic (W3C - Semantic Web Activity). Emphasizing facilities to enable automated processing of Web resources, RDF can be used in a variety of application areas from resource discovery, library catalogs and world-wide directories to syndication and aggregation of news, software, and content to personal collections of music, photos. RDF also provides a simple data model, which can accommodate rich semantic descriptions of multimedia content as J. Saarela, in [0], introduced a video content model based on RDF or J. Hunter & L. Armstrong , in [0], proposed a MPEG-7 description definition language (DDL) based on RDF.

MPEG-7 is a future standard for audiovisual information description to respond growing needs to process further audiovisual resources that will play an increasingly pervasive role in our lives. MPEG-7 is developed by MPEG (Moving Picture Experts Group) working group. It proposes a set of standard descriptors (Ds) and description schemes (DSs) and a definition description language (DDL). The Ds and DSs are the basic tools for describing the audiovisual information and the DDL are the language for defining the new tools in each particular application. Thanks to MPEG-7 work, computational systems can process further audio and visual information. In fact, in [0], L. Rutledge and P. Schmitz proved the need of a media in format MPEG-7 to improve media fragment integration in Web document. Note that the media fragment integration in Web document can be done until now only with textual document as HTML. TV Anytime with their vision of future digital TV that offers the opportunity to provide value-added interactive services has also said that MPEG-7 collection of descriptors and description schemes for multimedia is able to fulfil the metadata requirements for TV Anytime [0]. Many other projects have chosen MPEG-7 to realize systems that allow users to search, browse, and retrieve audiovisual information much more efficiently than they could using today is mainly text-based search engines [0].

As more and more audiovisual information becomes available from many sources around the world and people would like to use it for various purposes, there are then many other standards such as, SMPTE Metadata Dictionary, EBU P/Meta, TV Anytime and so on. Note that, by any way these standards provide only notional basic or basic tools for developing metadata applications. For specific applications they have to refine these standards. Moreover, it is difficult to have a model that can satisfy all requirements of various fields. So hybrid or incorporated approaches are often solutions of sophisticated applications as J. Hunter & L. Armstrong, in [0], have presented a hybrid approach for an MPEG-7 Description Definition Language (DDL). Additionally, because of specific requirements there are works that have to develop their own metadata models. Pushed by XML technology today, these proper models can be easily encoded in the flexible way that can interoperate with the other models. By example, CARNet Media on Demand after analyzed existing standards it has decided to use their proprietary vocabulary for metadata descriptions for building description structures for media and folders [0]; The model of InfoPyramid [0] allows to handle multimedia content description in a multi-abstraction, multi-modal content representation; J. Carrive et al. with the purpose for a format for presenting and storing generic knowledge

about a collection of television program has proposed a terminological constraint network model based on description logics in [0].

The model presented here follows the same approach. Its aim is to bring more sophisticated composition of different information resources to authoring and presentation multimedia documents. We have proposed in [0] a video content model for composing multimedia documents, in which we have used the descriptors from different standards. For instance, we use the Dublin core's element set for bibliographical information (e.g., title, creator, rights) and we have refined the Ds and DSs from MPEG-7 for structural information (e.g., scene, shot, event). We will discuss more about these refinements in section 7.

3.2 Animation

Right now, when it comes to creating animation for the Web, Flash is the king. It enables to create sophisticated Web applications and animation such as e-Learning while delivering low-bandwidth content. However, Flash is based on the binary format (SWF) that brings to end-users limitations such as, expensiveness (users have to purchase an authoring tool), no supporting to textual search, no reusing between authors, requirement of a plug-in program and so on. Recently releasing of SMIL Animation [0] is considered as an accomplishing of lightweight animation effects made by Flash animation technology. In addition, it overcomes the limitations brought by the binary format such as; animation can be authored by any textual editor; supporting textual search; could be played directly by browsers (internet explorer 5.5). Focusing on declarative animation and based on improved path-oriented animation model by putting animation on a time line, SMIL Animation allows to specify animation functionality for XML documents. It is intended for use embedded in other languages such as XHTML, SVG or CSS to provide animation. Finally, SMIL Animation is a set of basic XML animation elements suitable for integration with XML documents.

Our animation work described here is based on this set of basic XML animation elements. However, we have provided an abstract way for the integration of these base XML animation elements with media objects in our multimedia document model. This improvement prevents animation specification in our multimedia documents from becoming complex and redundant as the actual uses of SMIL Animation in the XML-based host languages such as, XHTML + Time or SVG. In addition, the abstraction is a flexible way for using animation elements. In [0] D. Vodislav has shown us the flexibilities of creating an abstract animation on a graphic object. A real animation on the object will be done as the real trajectory start point determined by initial position of the graphic object. We have the same idea as Vodislav with the abstract presentation of animation. However, we go further in creating abstract and independent animation elements that can be applied on different concrete media objects instead of only one media object. A more discussion about this will be found in section 6.

3.3 Media fragment hypermedia and synchronization

Media synchronization and hypermedia in multimedia document have been largely investigated and well described in the research literature [0], [0], [0], [0]. However few of them have supplied with the media fine-grained synchronization, such as SMIL (*W3C - Synchronized Multimedia Integration Language*) a standard to post multimedia on the Web. SMIL provides a way to integrate different media types into a document, but existing tools to compose the multimedia document from these media remain simple (based on two operators *seq* and *par* for temporal synchronization and *layout*/*region* for spatial layout). Therefore the presentations often have to specify media arrangement in both time and space with absolute values. Other works propose more sophisticated composition [0] [0], however they remain for specific applications, such as the work in [0] provides a simple way to fine-grained synchronize fragments of continuous media (audio and video) with the static documents (text, image). However, the level of media structurarion is not deep and rich enough, it only defines *story*, *clip* and *scene* objects and the *scene* is the smallest unit of the media structure. So it can only provide an interaction from the static document, not from the continuous media side as the video object hyperlinks. One more important limitation is the *video-centered* hypermedia model, i.e., the logic of the presentation of the document is limited to the

sequential logical structure of the continuous narrative media (video or audio). So a more general model for fine-grained composition of multimedia presentation is expected.

The *anchor* and *area* technologies are used in the existing standards such as, CMIF, HyTime, SMIL, HTML, etc. to decompose media objects into spatial/temporal fragments for hypermedia and fine-grained synchronization. In [0] Hsu et al. introduced a complete system allowing to generate sophisticated product's electronic documentation from a collection of documents about the product. The system possesses manual and automatic extractors such as, textual extractor from images that allows user to extract specific characteristics from a product's image; the *anchor* technique of HyTime is then applied to encode these extracted fragments for sophisticated hypermedia to other documents describing these specific characteristics of product. Such a mechanism is also founded in other authoring systems such as, GRiNS SMIL editor[1] from Oratrix that allows not only to hypermedia to intra or extra of document but also to synchronize with other media objects or other media fragment objects. Therefore, the current approaches provide some ways for media fragment integration. However, by these ways media fragment integration remains limited by the absolute specification, static region, no structure and lack of signification. Recent releasing of standard SMIL 2.0 has provides the use of the *fragment* attribute of the *area* element that can give a more signification description of media fragments. However, this way is only used for the existing structured media such as, HTML, SVG or SMIL document. In this our work, we introduce our solution for a more semantic media fragment integration for multimedia presentation. It is based on our *structured media* concept that can be found more detail in section 7.

4 Examples of multimedia complex presentation

The examples presented in this section result from our application under development in which we experiment our concept of *sub-element* as the basic structure for our fine-grained synchronization architecture. These examples are grouped into two types of media composition: temporal synchronization and spatio-temporal synchronization. These examples will be used to illustrate our model and its implementation in the other sections of the paper.

4.1 Temporal synchronization

Figure 1 contains two screenshots of a small document presenting the Opera project. The main media is a video clip on the right part of the presentation. On the left side of the video clip, a set of text objects represents the outline of the video clip. During the presentation, each text on the left part is resized and colored in red when the corresponding video segment is played. Such a presentation reminds the user of the video structure during presentation. It is like a slideshow document as performed with simple modeled like [0] or *RealPresenter* of *RealNetworks*, in which titles and their corresponding image slides are dependently played out with segments of training video.

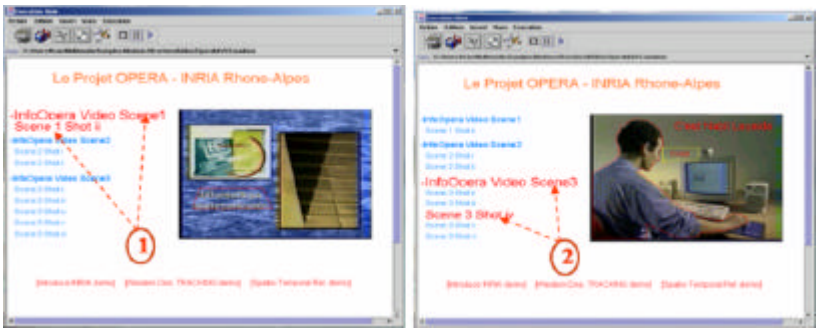

Figure 1. Text animations synchronized with video segments. (1) The second shot of the first scene is playing; (2) The fourth shot of the third scene is playing.

[1] GRiNS for SMIL-2.0 by Oratrix, http://www.oratrix.com/GRiNS/SMIL-2.0/

Even simple, this allows us to show how our model is adapted to media synchronization: the video is logically structured in video segments and represented as a *structured video* media. So the texts outlining the video structure can be easily synchronized with and linked to the video segments corresponding with them. Then not only the texts can be animated synchronously with the video structure, but also the user can interact on these texts to return back or go forward to expected video segment.

Technically, we use the *structured video* for representing video structure indexing, the *sub-actor* for representing a video segment style in use, the *sub-intervals* for temporally representing a video segment in timeline of the presentation. Of course there are references among these elements for the information. Then the synchronization and the links among the video segments and the texts are specified by the *equals* temporal relations between the *sub-intervals* and the *intervals* of text animations and the link targets to beginning of these *sub-intervals*.

4.2 Spatio-Temporal examples

A more developed discussion on the small document above shows us one more interesting application due to the interactions and spatio-temporal synchronization with video objects.

Spatio-Temporal Synchronization: in this example (Figure 2), the author has added a textual speech that is always aligned on the right of character's head during the occurrence of this character. The textual speech is a text media illustrating words pronounced by the character. This effect is created by a spatio-temporal relationship specified between text media and video object.

Video Object Hyperlink: in the same example, we can see a blue contour around the video object presenting a hyperlink on this object. This hyperlink allows to access another document, here the picture on the right. When the mouse enters in a video object region with a hyperlink, a blue contour is displayed around the video object and evolves with the shape of the object. The video hyperlink is similar to a hyperlink on web pages, but it is dynamic, i.e., the video hyperlink on a video object is only activated during the appearance of the object.

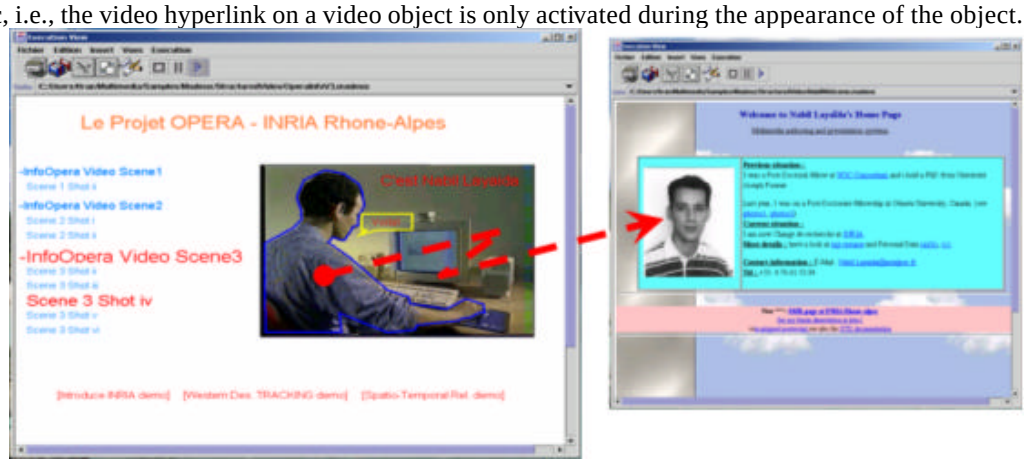

Figure 2. Spatio-temporal synchronization and video object hyperlink.

To allow all of these operations we need to identify more deeply in the logical video structure the video objects, that are described not only in the time but also in the space. Then to spatially locate video objects to compose with other visible media, we use the *sub-region* elements.

The above examples have shown the needs of the sub-elements in the model for editing and presenting a multimedia document with animation and video elements. The rest of this paper will discuss about the integration of these sub-elements in our multimedia model and will describe their implementation in our language. First, we describe in the next section the multimedia document model which is the basis of this work.

5 Logical structure, interval and region based model for multimedia document

A multimedia document model has to realize the integration of a set of media elements through temporal, spatial and hyperlink models. In most existing models, this integration is often partially mixed, for instance, in XHTML+TIME, spacing and timing attributes are directly attached to the media elements. In SMIL, the spatial layout is separated into the Head part of the document. However, the temporal axis of SMIL includes the media element declaration. Such mixed specifications often carry out a complex rendering structure and often generate redundancies when the document grows up. For instance, redundancy occurs when several media elements must be presented at the same time or at the same space. On the other hand, the decomposition of document model in distinct dimensions enables to simplify the authoring and presenting process, for instance it allows to use separate formatters for the spatial and temporal axes. In addition, when the model is decomposed into several axes (object, time, space, hyperlink, etc.), the composition of multimedia presentation is more condensed and more flexible. A media object can be reused for several displaying in presentation.

Following this decomposition approach, our Madeus [0] model can be considered as an extension of SMIL standard for handling the following features: better separate the media, temporal and spatial information; complete the hierarchical temporal operator-based model with relations; provide a more elaborate spatial specification model (with relative placements)More precisely, a Madeus specification has four main parts (see Figure 3).

The *content* part is an abstract level to allow hierarchically describing and organizing data objects that will be used in the presentation. At this level the semantics of specification depends on the document type instead of on its presentation that will be specified later in lower parts. A data object could be any media type such as: *text, image, audio, video, HTML, SVG, Applet, etc*. A data object is associated with its content information through a *FileName* location attribute or a content attribute such as *TextValue*. Metadata can be added to describe information about the media such as, author, title, creator, rights, etc. More importantly, this part allows a multilayered description of structural information of media content, e.g., sequence, scene, shot, event and object of a video media; background and regions of an image media; and so on. This capacity will be developed in section 8. In addition, this part also allows specifying abstract functions as abstract animation elements that will be mentioned in section 6.

The second part is the *Actor* part. If the first part allows to define content data, then this part allows to specify presentation styles and interactions on content data such as *FillColor, LineColor, FontFamily, FontSize*, etc, and *Hyperlink*. Such a separation of content data from style presentation is well known in document system because it allows the reuse of data content, i.e., the same data content can be presented several times through different layout formats. Data content with specific presentation styles is called *DefActor*.

The *Temporal* part is the temporal structure axis of the presentation. It allows conducting the *DefActor* over time. The model used for this level is an interval-based model where placements of *intervals* can be defined by either absolute coordinates or relations among intervals [0]. Each *interval* possesses the timing attributes: *begin*, *duration* and *end* (with the constraint, *end* = *begin* + *duration*). Values of these temporal attributes can be default values inherited from intrinsic values of continuous media, specified by author or automatically calculated by formatting functions through temporal relationships between intervals. A set of intervals can be grouped into a composite interval called *T-Group* and associated to a temporal operator. This model provides therefore a hierarchical temporal structure in which the *interval* is the basic time unit, and one or several *DefActor* can be filled into these basic time units as presentation activities to be performed during these periods. An interesting characteristic that differs from other interval-based models such as SMIL and ZYX [0] is that our model supports constraints that allow flexibility in the resulting presentation, i.e., the author can only specify some key values, the rest will then be automatically calculated.

The *Spatial* structure axis is basically similar to the temporal model but it is applied on two dimensions defining a 2D box hierarchy where the leaves are the *Region* element. The set of spatial relationships available for relative placements among *Region*s are: *left_align, center_align, right_align, top_align, bottom_align, bottom_spacing*, etc. Such a constraint model provides relative layouts that are much more flexible and more comfortable than the absolute spatial layout such as in SMIL model.

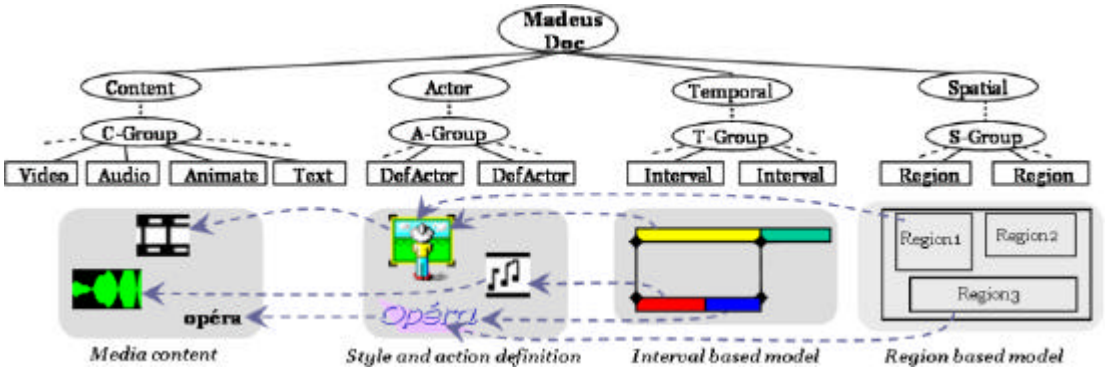

Figure 3. The logical, temporal and spatial structure of the document presenting the Opera team

The interval and region based model is known to be one of the most expressive models among existing models [0]. The limit of this approach is mainly due to the granularity provided by the leaves of the structure. In fact, there are many media objects having rich content information such as image, video or long text for which authors want to set finer-grained synchronizations in order to produce more sophisticated presentation scenarios. Examples of such needs are shown in the section 3. The problem cannot be solved by simply using the existing model and defining deeper hierarchical structures as found as in existing models with the *Anchors* and the *Area* elements. Such a solution is only a temporary limited solution with drawbacks of absolute and non-significant specification. Indeed, media objects do have their own semantics, temporal and spatial organization that the multimedia model must consider when composing media fragments during document composition. This is why we propose the extensions of the next following sections.

6 Abstract animation model

An animation is defined as a time-based function that gives effect to an attribute of a target element during a time interval of the target element duration. The models for animation can belong to one of the following classes: constraints-based, dataflow or path-oriented models [0]. SMIL animation is based on a path-oriented model and provides a set of basic animation elements comprising *Animate*, *AnimateMotion*, *AnimateColor* and *Set* for XML documents. These animation elements carry timing attributes (*Begin*, *End* and *Dur*) for their time layout during presentation of media object. As a refinement from these basic animation elements, we propose an independent definition of animations from the animated media, which provides more flexible and reusable animation components.

Within the multimedia document model, an animation is considered as a dependent continuous element. That means an animation exists in a document only through its relationship with another media object. In existing models, this relationship always is one to one, i.e., an animation is created to only one media and often is declared as a component of a media. But we can frequently observe that several media presentations are affected by the same animation function. Let see a specification of such a presentation in Figure 4. It is an excerpt of the demo animations[2] using HTML+TIME. The specification tries to display five phrases in sequence and applies on each text's display the same animation that changes of display color from white to black. Such a specification is clearly inefficient with five-time the repetition of the same animation, and then could generate tedious work when the author wants to modify that document.

```
<t:seq>
  <p timeContainer="par" timeAction="display"> <t:animateColor attributeName="color" from="white" to="black" dur="3s"
  autoReverse="true"/>In case you were wondering … </p> …
  <p timeContainer="par" timeAction="display"> <t:animateColor attributeName="color" from="white" to="black" dur="3s"
  autoReverse="true"/> Pretty cool, isn't it?</p>
</t:seq>
```

[2] demos of HTML+TIME, http://research.microsoft.com/~pschmitz/demos/H+Tdemos.html

Figure 4. Specification of SMIL animation in a HTML+Time document

SMIL animation elements can be specified outside media elements and then thanks to strength of full XPath to define multi-target media elements such as `<t:animateColor xlink:href="XPath expression" begin = "…" dur="3s" … />`.

It seems to address the redundancy above, but the timing specifications on the animation will break down the time layout of the presentation. The problem could be radically addressed by a cascading animation sheets that provide flexibility, no redundancy, maintainability and remote control. However, the concept is one of the wishes for SVG2.

In our model, the definition of an animation can be separated both from its timing properties and from the animated media elements, following the multi dimensions approach of the Madeus document model discussed in the previous section. Such a separation produces abstract animations that provide high flexibility and unification in using animation (see section 7.1 for a more concrete specification). Programming an abstract animation in our model is performed through two steps: the abstract animation specification and the application specification (see Figure 5):

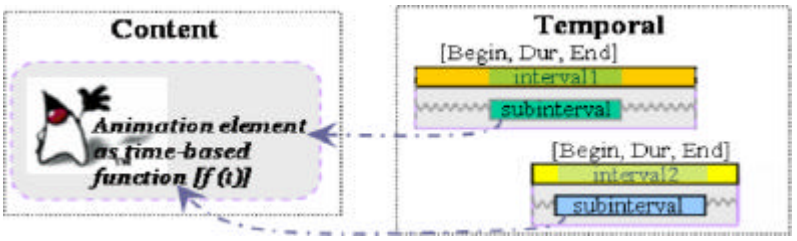

Figure 5. The abstract animation model

- The abstract animation specification defines abstract animations that are presentation-neutral and then will be able to be reused several times for diverse media target elements within the document. Our abstract animation is a refinement of SMIL animation in which timing attributes (*begin, dur* and *end*) and *targetElement* attribute are restricted. That allows animation to be separated from its presentation and its target media. More importantly, we proposed the use of an abstract timing between interval {0, 1} on which abstract time points corresponding to locations on the trajectory, the scaling, the translation, etc. are defined. This abstract interval can be used to map out several real timing layouts for the animation (see Figure 10). Due to the presentation-neutral of the abstract animation, the abstract animation specification is performed in the *Content* part of the Madeus model extended with the animation vocabularies: *Animate*, *AnimateMotion*, *AnimateColor*, etc. A concrete practical about it will be done in section 8.1.
- The application specification defines animation's instances that contain effective timing layouts and one or more target media elements. For the real timing layouts, the abstract animation is filled into temporal objects defined in the *Temporal* part. Then the *targetElement* attribute can be used to identify one or more target media elements explicitly. As known, the animating on a media object must be bounded in media's presentation. Therefore, the animation's temporal object must be the sub element of the target media's temporal object. Such a sub temporal element is called a *sub-interval* of the target object's *interval* (see section 7).

By this way, an animation is not only defined independently from the object on which it is applied but also from a temporal behavior, so it can be reused for several time contexts. In that sense, our approach can be compared to the media construction abstraction (MCA) proposed by [0]. MCA aims at helping users in their design process through MCF diagrams where media composition allows event-based definitions. Therefore, it uses a box and connector paradigm. On the contrary, our animation definition has an interval-based specification where composition results from relations between intervals. Therefore, it allows the specification of synchronizations between animation instances and other objects of the document.

7 Structured media models

Most multimedia models do not semantically handle portions of media element. When they allow access to portions of media, only low level (time instant, space co-ordinates) attributes are available. For instance, in SMIL, the *area* element that can specify a spatial portion of a visual object has been extended for hyperlink spatial and temporal portions of media objects. However, this way for accessing fragments of video is very complex for the author because the *area* elements must be defined by their *coords*, *begin* and/or *end* attributes. The SMIL code fragment below (see Figure 6) shows the selection of hyperlinks from video fragments that are specified with time and space attributes.

```
<video src="http://www.example.org/CoolStuff">
  <area id="joe" begin="0s" end="5s" cords = "0%, 0%, 50%, 50%" href="http://www.example.org/"/>
  <area id="tim" begin="5s" end="10s" cords = "0%, 0%, 50%, 50%" href="http://www.example.org/Tim"/>
</video>
```

Figure 6 Low-level media's portions handling in SMIL 2.0

In this section, we present a more high-level way for specifying complex media. We only focus on video media, because it is a rich media and we have completely developed an experimental system for it that demonstrates our general model.

7.1 Video structure model

Video media has very rich media content, which must be explicit in order to provide semantic access into this media for multimedia composition and presentation. Numerous works have been devoted to the content description of this media [0] [0] [0] [0] and among them, the emergence of the MPEG-7 standard is the most important. The particular context of multimedia document authoring and presentation systems requires specific content description needs of video media. We have been working on that subject for two years and have proposed a video model that is consistent with our multimedia document model [0]. At the top level, the video model comprises different description schemes (DSs) proposed from MPEG-7, it consists of six description schemes following:

- *MetaInfo* and *MediaInfo* descriptions are description schemes to describe the bibliographical information about media such as title, creator, rights, file format, compression format, duration, color system, sound, etc. (from Dublin Core).
- *VideoStructure* description is a low-level description scheme that directly indexes on raw video to extract the structure of the video content. The scheme is based on the hierarchical structure on which the video content is described in terms of sequence, scene, shot, transition and object elements as proposed in several existing approaches for video indexing. We have also identified other elements that can be relevant for multimedia composition, such as: events (a character goes out a car), moving objects and spatio-temporal relationship elements inside a shot level (two cars move in parallel then one passes in front of the other).
- *Summary* description is based on the description scheme from MPEG-7 (Summary DS) that enables browsing video at different levels of granularity.
- The *Semantic* description allows not only to group the elements of the video structure into the semantic elements such as *characters* or *objects* but also to describe the interactions between them, for instance, a *footballer kicks* a *ball*.
- *Thesaurus* description describes semantic terms and expressions to classify elements in the video content description.

In the context of multimedia composition, the most important part in this model is the *VideoStructure* description, because it makes explicit video's internal scenario for composing with other media objects. The *VideoStructure* description model derives from the *VideoSegment* description scheme of the standard MPEG-7, which can describe a recursive or hierarchical structure of a segment of video that can correspond to an arbitrary sequence of frames, or even the full video sequence [0]. The other video description parts provide more comfortable features when composing video inside a document such as, the *Semantic* and *Thesaurus* descriptions provide semantic and multi-target specifications of video elements. For instance, if the author wants to make hyperlinks on

all of occurrences of an actor, he or she has not to specify the link for each occurrence of the actor, the hyperlinks could be made only once by reference to the semantic element of the actor in the *Semantic* description part that has grouped all of these occurrences.

We have briefly described our video model that allows us to deeply access to video content in composing multimedia presentations. The application of the model for video fragment integration into a multimedia presentation requires to harmoniously integrate it into the multimedia document model. According to the decomposition approach of Madeus model, the video fragment integration in Madeus document comprises three steps as all the other media compositions. However, at each step some extensions are needed for adapting to the new feature of fragment composition (see Figure 7):

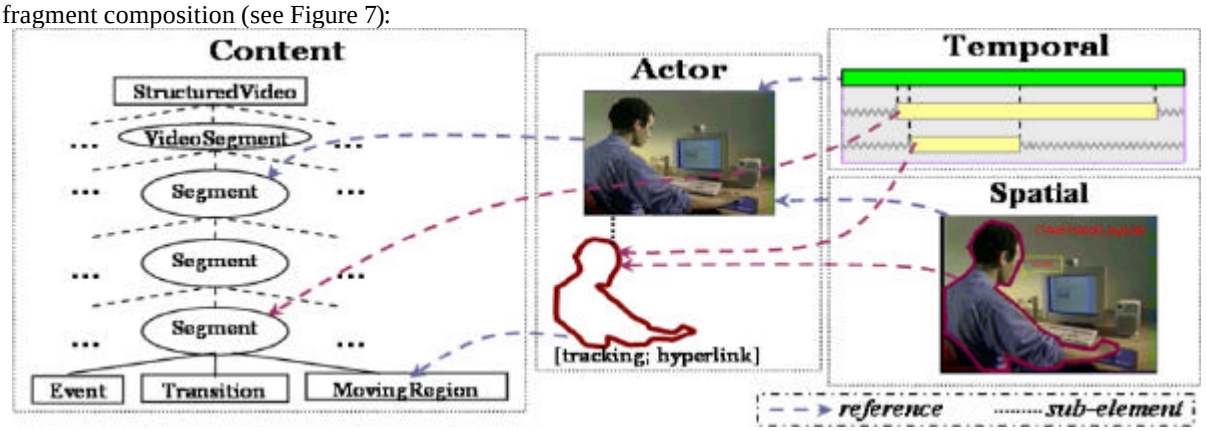

Figure 7. The structured video model in multimedia document (Madeus)

- The *StructuredVideo* specification step generates descriptions about video content in accordance with the video model. The specification could be manual, automatic or semi-automatic. This specification provides a new type of media element in the *Content* part of Madeus document that is differenced from a classic declaration of media used, which only refers to raw data of the media. The *StructuredVideo* element allows not only to refer to the media content but also to include or refer to a description of that media content.
- The usage specification is the definition of the *DefActors* in the *Actor* part that can refer to any video fragment defined in the *StructuredVideo* element in the *Content* part. The reference to the video fragment can be an ID or an XPath expression for multiple selections. The specification provides a more semantic fragment definition than using the *ClipBegin* and *ClipEnd* attributes in existing standards such as HTML or SMIL. In addition, the fragments can be used in a multilayer, e.g., A *DefActor* is defined by referring to a video shot, and if in that video shot there are same sub fragments such as, a moving region, then the author can specify a hyperlink or a tracking action on this moving region.
- The last step in video fragment integration is the specification of the temporal and spatial layout for the video *DefActor* that contains both content and style information available for presentation. As for all time-based media, the temporal object can get its duration information from the intrinsic duration of the corresponding media. The important notice is that if the *DefActor* contents the actions on sub video fragments of the fragment referred by the *DefActor*, then the temporal and spatial objects that play out these actions could be configured automatically through the description information and relatively with the temporal and spatial objects of the *DefActor*. The most interesting is that the author could directly specify every temporal or spatial relation with the sub fragments described within the video fragment preferred by the *DefActor*. Therefore, that model allows system to automatically create these temporal and spatial elements for that fragment. The author can then use this interval/region to synchronize that fragment with other media.

The integration of video content description into Madeus document model allows to express the internal scenario of a video for composition with the other media objects. The integration also requires not only the extension of supporting media type in the *Content* part (*StructuredVideo* element) but also the sub-elements in all *Actor*, *Temporal* and *Spatial* axes. These extensions are discussed in the section 7.

7.2 Other media structure model

The positive result of the first experiment with video fragment integration has encouraged us to continue to structure other media for our multimedia authoring and presentation system. Concretely we have applied a similar approach to audio and text media. We have proposed an audio content description model using description schemes from MPEG-7, such as, the *AudioSegment* description scheme [0] that can describe a temporal audio segment corresponding to a temporal period of an audio sequence. And for the structuration model of the text media we have opted for a classical structure model of textual document that is a hierarchical decomposition of *Chapter*, *Section*, *Paragraph*, *Sentence*, *Phrase*, *Word* and *Character* elements. Such the integration of audio and text fragments allow the Madeus system to compose and present complex multimedia documents as Karaoke, in which video, audio and text can have between them fine-grained presentation constraints.

7.3 Authoring with the structured media

The authoring of multimedia presentation often directly uses media for composition and therefore limits the fragment integration capacity among media objects. We have integrated in the authoring system two new modules (see Figure 8) that allows media fragment composition. Firstly, the analysis/generation module automatically analyses media content and then generates a first version of media content description in accordance with the media content model. Secondly, the authoring media content description module, which takes the generated content description as an input, and allows the author to manually complete or modify it. Finally, a structured media that contains not only the content information for its presentation but also the description of this content for fine-grained synchronization is available for multimedia integration. This authoring schema presents not only a real time generation of media content description but also could reuse existing media content described by standards such as MPEG-7.

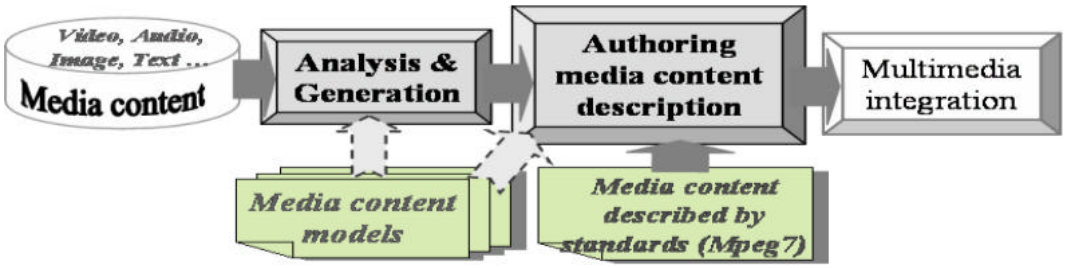

Figure 8. An authoring schema for multimedia document composition with structured media

We have applied on this schema with video media and we have integrated it into the Madeus system: the analysis/generation tool allows generating *StructuredVideo* elements automatically and the *VideoMadeus* editing tool that allows the modification of the content description information of the *StructuredVideo* element and the extraction of any fragment of video for its composition inside a Madeus presentation. That provides facilities for the author not only to specify temporal and spatial layout for the presentation of other media objects according to events in the video sequence, but also to make presentation constraints between the media objects and the video events, such as, a text introducing about an actor displays when the actor appears on screen.

8 Sub-elements specification

As illustrated in the previous section for the integration of the animation and the structured media, it is necessary to extend the components of the Madeus model to adapt the modeling requirements for the internal description of media object: content description, behavior for these sub content, temporal and spatial objects attached to these fragments of media. Thanks to the hierarchical structure-based model of Madeus, these extensions can be easily performed in the Madeus model. We have introduced a new hierarchical structure level on the Madeus document

model that is called *sub-Elements* structure model of Madeus. In the rest of this section we describe these extensions in all the decomposition axes of the Madeus model: *Content, Actor, Temporal* and *Spatial*. For each *sub-Element* type corresponding to each axis, we precisely define the constraints induced by the element in which it is included. Therefore the distinction between *Elements* and *sub-Elements* will be clearly stated. Then a synthesis of the model is given in the last part of this section through the Figure 17 where the global structure of a Madeus document is presented.

The Madeus language syntax is formally described as a XML DTD[3] and therefore it takes full advantage of all XML existing tools. A Madeus XML source document contains the four following main parts:

```
1.  <Madeus Name="DocMadeus" Version="2.0" xmlns='http://www.inrialpes.fr/opera/dtd/madeus'…>
2.    <Content>…</Content> <Actor>…</Actor> <Temporal>…</Temporal> <Spatial>…</Spatial>
3.  </Madeus>
```

Figure 9. General structure of a Madeus document

For each new *sub-Element* introduced in our model, we will give its example specifications according to an excerpt of the example document in the section 4. The extensions brought to the corresponding Madeus part: *Content, Actor, Temporal* or *Spatial*.

8.1 Content specification for animation and structured media

As discussed above the animation and the structured media elements have to be specified in the *content* part, which allows to describe the internal structure of media and to reuse these content elements several times in different parts of the document.

The animations in Madeus documents are specified by reusing a set of the basic animation elements defined by SMIL (see section 6). The *keyTime* attribute (e.g. *KeyTime*="0;0.2;0.8;1") is the same as SMIL ones but it refers to the abstract interval so an animation specification can be used several times by mapping its abstract interval on the concrete intervals as illustrating in Figure 10.

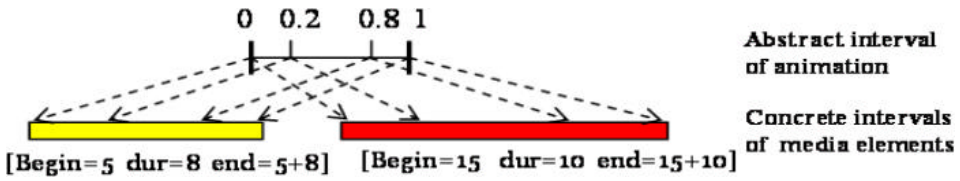

Figure 10. An animation abstract interval maps out two real intervals.

The animation code shown in Figure 11 is the specifications of two animation (*Color* and *Size*) to animate the texts following the structure presentation of video. This example shows the advantage of the abstract and proper specification of animations. The use of these animations for animating various media objects will be described in the section 7.4 that is devoted to *sub-intervals* definitions.

```
4.  <Content> <C-Group> …
5.    <AnimateColor ID="C_ColorAni" attributeName = "LinePaint"  values="rgb(255,0,0);rgb(0,128,255)" keyTimes="0;1"
      calcMode="paced" />
6.    <Animate ID="C_SizeAni" attributeName="FontSize" values = "0;10;10;0" keyTimes="0;0.095;0.97;1" calcMode = "linear"
      additive="sum"/> …
7.  </C-Group>… </Content>
```

Figure 11. Specification of color and size animations in the content part of a Madeus document

The *Content* part of Madeus has also been extended with new media types for structured media comprising *StructuredVideo*, *StructuredAudio*, *StructuredText*. These new types introduce an internal structuration level for the media, which was not available with the previous media types that only represent raw data to play. That provides easiness and significance to integrate the media fragments as explained in section 7. The example below (Figure 12) presents a *StructuredVideo* element that allows not only to locate the video *OperaInfo.mov* but also

[3] L. Villard, Madeus model DTD, http://www.inrialpes.fr/opera/madeusmodel.dtd, 2000.

to describe its content. The easy integration of these video structures with the other media presentations will be illustrated in following sections.

```
8.  <Content> <C-Group> …
9.    <StructuredVideo ID="VideoContentInfoco" FileName="Media/videos/OperaInfo.mov">
10.     <!--MetaInfo,MediaInfo,Thesaurus,Semantics and Summary specification codes here-->
11.     <VideoSegment ID="…" SegmentType="CompleteVideo"…>
12.       <Segment ID="Seq1" SegmentType="Sequence">…
13.         <Segment ID="Scene3" SegmentType="Scene" >…
14.           <Segment ID="Scene3Shotiv" SegmentType="Shot">
15.             <MovingRegion ID="OccNabil2" ListKeyPoint="…" ListKeyTime="…" InterpolationFunction="BSpline" Segment-
                Type="MovingRegion"…/>
16.             </Segment> </Segment> </Segment> <Segment ID="Seq2" …> …</Segment>
17.   </VideoSegment> </StructuredVideo> </C-Group>
18. </Content>
```

Figure 12. Specification of a video content in the content part of a Madeus document

The specifications of the *StructuredAudio* and *StructuredText* elements also look like the *StructuredVideo* specification in which the element types are replaced by the adequate description elements according to their models.

8.2 Actor specification with the subDefActor element

In authoring a multimedia document, the author need the tools to specify actions or styles on **media fragments** such as a *highlight* on a phrase or a word of a text, a *tracking* or *hyperlink* on a moving region of a video segment. A sub-element of the *DefActor* element called *subDefActor* is then provided for these purposes. It uses a *Content* attribute valued with IDs or *XPath* expression to refer to the media segments on which the action or style must be applied. The segments referred must belong to the structured description of the media element. For instance, in the example of the Figure 13, the *subDefActor* element "OccNabil2SubActor" refers to the moving region **OccNabil2** (line 15 of the Figure 12).

The set of *Actions* available for a *subDefActor* can be: *Highlight*, *Tracking*, *Masking*, *Linking* or *toolTip*. In the example of the Figure 13, the *subDefActor* element associates a *tracking* and a *hyperlink* to the **OccNabil2** object, allowing the user to click on the corresponding area on the video to display the target document (see Figure 2). This mechanism can be compared with the *Anchor* and *Area* technique of existing languages. However, it provides a more semantic fragment specification instead of the low-level and absolute content access. In addition, the *defSubActor* element allows to compose fragments with richer styles and actions instead of only the *hyperlink*.

```
19. <Actors> <A-Group> … <DefActor ID="InriaInfocoSeq1" Content="id('Seq1')">…
20.   <subDefActor ID="OccNabil2SubActor" Content = "OccNabil2" Actions="Tracking;HyperLink"
      HRef="../Samples/NabilWelcome.madeus"/>… </ DefActor > … </A-Group> </Actors>
```

Figure 13. Specification of a *SubActor* element in the *Actor* part of a Madeus document

8.3 Temporal specification with the subInterval element

Sub-temporal objects are necessary to carry out: the *animations*, the *subDefActor* objects and the *temporal representation* of the media fragments. A *subInterval* element is defined **inside** an *interval* element for that purpose. The *subInterval* element derives from the *interval* element in our interval-based model. Therefore, as any temporal object, the sub-interval can be involved in any temporal relation of the temporal document specification. The refinement of the *subInterval* through the inheritance is that the *subInterval* element has a **during** temporal constraint with its parent *interval*. There are two main uses of *subInterval* elements: for *scheduling animations* on media elements and for *temporally representing media fragments*. For each case, the *subInterval* element is configured differently.

If the *subInterval* is used for layout of the animations (line 25 in of the Figure 14), the *subInterval* element has to specify the *Animate* attribute to attach one or several animation elements to it. The sub-interval also carries

the *targetElement* attribute that specify the animated media elements. The temporal scheduling for the *subInterval* must be explicitly specified by absolute specification of its timing attributes (*Begin, Duration* and *End*) or by relative specification through the temporal relations (line 32 in of the Figure 14 is the specification of the *Equals* relation to synchronize the *begin* and *duration* of the animation sub-interval with the corresponding video segment).

In the case of media fragment representing, the *subInterval* carries the *subActor* attribute to specify the *subDefActor* elements referring to the media fragments. The media fragments can be the static fragment such as a phrase of a text media, then the time specification for static fragments must be explicitly specified as for the animation case. If the *subDefActor* refers to a continuous fragment belonging to a continuous media such as audio segment or video segment, then the *subInterval* will be automatically scheduled thanks to temporal information of the fragment description (line 30 in of the Figure 14). Such a *subInterval* makes explicit a temporal fragment for further synchronizations with other *interval/subInterval* elements through the temporal relations (line 32 in of the Figure 14). The key point of this model is to maintain the intrinsic time constraints of the *subIntervals* inside their media content *interval*, together with allowing them to be integrated into the timed schedule of the whole document.

The definition of the *subinterval* is not recursive because it only aims at representing and synchronizing subparts of a media object on the timeline.

```
21. <Temporal> <T-Group >
22.   <!—the interval of the texts representing video structure-->
23.   <Interval ID="textInt" Actor ="…;TxtScene1; TxtScene3Shotiv" Duration="pref:43s">…
24.     <!— the animation sub-intervals of the texts interval-->
25.     <subInterval ID="TxtScene3ShotivAni" Animate="C_SizeAni;C_ColorAni" TargetElem="TxtScene3Shotiv"/>…
26.   </Interval>
27.   <!— the structured video interval-->
28.   <Interval ID="StrInriaInfoco" Fill="freeze" Duration="…" Actor="InriaInfocoSeq1" >…
29.     <!— the sub-interval representing video object -->
30.     <subInterval ID="OccNabil2SubInt" subActor="SubActorOccNabil2"/>… </Interval>
31.   <!—the equals synchronizations between video segments with animations on the texts-->
32.   <Equals Interval1="StrInriaInfoco.Scene3Shotiv" Interval2="TxtScene3ShotivAni" />…
33. </T-Group> </Temporal>
```

Figure 14. Specification of *subInterval* elements and *Equals* relations in the *Temporal* part of a Madeus document.

The Figure 15 is a time line representation of the temporal elements and relations defined in the Figure 14. The sub-intervals *StrInriaInfoco.Scene1*, *StrInriaInfoco.Scene1Shotii*, *StrInriaInfoco.Scene3* and *StrInriaInfoco.Scene3Shotiv* of the *StrInriaInfoco* video interval are synchronized with the text fragment sub-intervals by *Equals* relations.

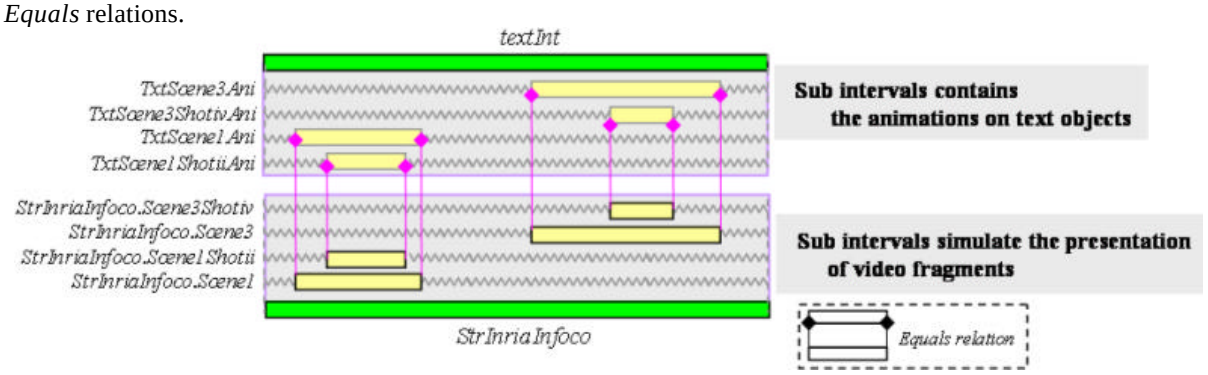

Figure 15. Time line representation of temporal elements and their synchronization

8.4 Spatial specification for subRegion element

In the spatial model, the *subRegion* element plays a similar role as the *subInterval* for representing spatial fragment of visual media object. The examples of the *subRegion* are areas of a person or an object on a picture or a

video or else areas of a word or a phrase of a text. The *subRegion* derives from the *region* element of the region-based model, so it is also a spatial object that is defined by position and dimension properties, and that can be involved in spatial relations of the spatial layout of document. By definition, the *subRegion* element has an **inside** spatial constraint with its parent region. This spatial constraint is automatically specified through the spatial information of the spatial fragment description. The *subRegion* element carries the *subActor* attribute to refer to a *subDefActor* element (line 38 in the Figure 16). Together with its intrinsic position and dimensions, the identification of *subRegion* provides the means to specify more sophisticated spatial relations with other regions. For instance, the spatio-temporal synchronization of that region, e.g., the speak bullet is set on the top of a character's occurrence by the *Top-Align* relation (see Figure 2). If the character's occurrence is a moving region, the *Top-Align* constraint will result in moving the speak bullet following the movement of the occurrence in the video (lines 41 and 42 in the Figure 16). The other applications of the *subRegion* element are interactions on sub areas of visual media objects such as hyperlink, tracking or displaying tip text for the area.

```
34. <Spatial> <S-Group ID="TotoSpa" Width="…" Height="…"> …
35.   <!–the structured video region-->
36.   <Region ID="PosInriaInfoco" Actor="InriaInfocoSeq1" Left="…" Top="…" Width="…" Height="…" …> …
37.     <!–the sub-region representing video object in space-->
38.     <subRegion ID="OccNabil2SubRegion" SubActor="OccNabil2SubActor"/> …
39.   </Region> …
40.   <!–the "Top_align" and the "Left_align" constraints between the video object and a text region, that make the text
      following the movement of the video object region -->
41.   <Top_align Region1="OccNabil2SubRegion" Region2="TextMotionRegion"/>
42.   <Left_align Region1="OccNabil2SubRegion" Region2="TextMotionRegion"/>
43. </S-Group> </Spatial>
```

Figure 16. Specification of the *SubRegion* element in the content part of a Madeus document

8.5 Syntheses

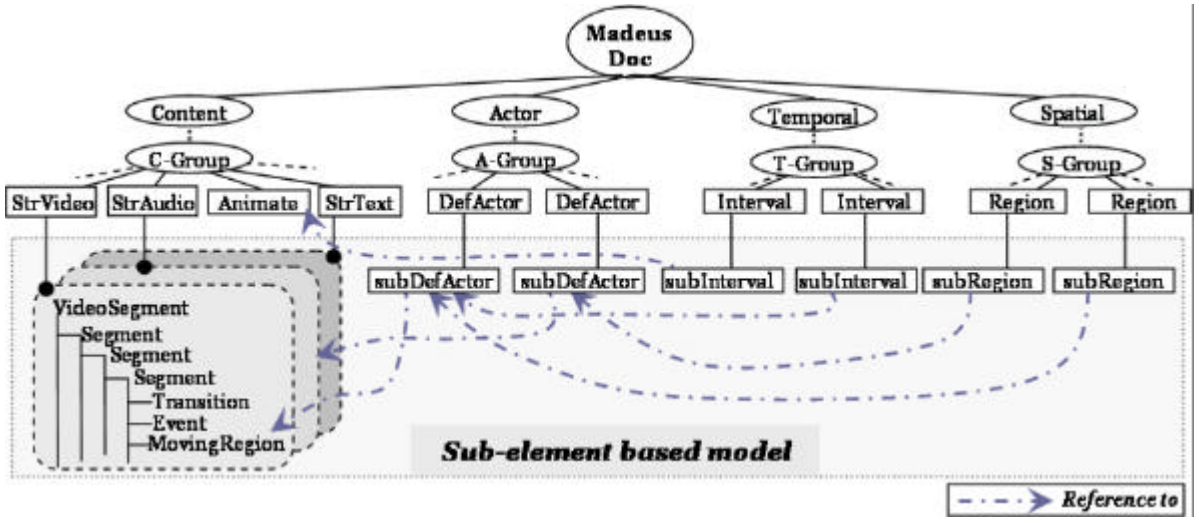

Figure 17. A Madeus document structure with content description, *subDefActor*, *subInterval* and *subRegion* sub-elements

The Figure 17 resumes the definitions of *sub-elements* and their relations described above. As a conclusion, a *sub-element* always belongs to an element (structured audio/video/text in the content, the actor, interval and region) and relates to that element to express its semantic dependency in the corresponding dimension. Notice that except for the content part, the *sub-elements* are not recursive.

9 Conclusion and perspectives

The composition and the presentation of multimedia document is limited if the media objects are considered as atomic elements. Using a declarative approach, we propose a sub-element based model that allows significantly decomposing atomic media in accordance with the adequate content models and fragmentally integrating media objects in all style, temporal and spatial axes of the multimedia presentation. The sub interval and sub region elements in the temporal and spatial axis allow handling sub-parts of media for spatially and/or temporally presenting them and synchronizing them with other media. Moreover, the structured media and sub actor elements allow the semantic definition of these sub parts and provide the means for reusing them.

This model is implemented in the Madeus multimedia document framework that comprises the Madeus language and the prototype of a multimedia authoring and presentation environment also called Madeus. In that context, we provide the author with a convenient interface that allows him to specify/modify sub-elements while keeping the perception of the dependency of these sub-elements to their media from which they are extracted.

Note that the sub-elements are proposed firstly to provide a way to carry out functions such as animations on media objects and secondly to represent the intrinsic information of media. In the former case, the sub-elements can be created and modified as other elements of the document, e.g., a sub-interval that is associated with an animation function can be created and modified for animating a target media object at any moment of the authoring process. In the later case, the sub-elements make up desired internal structures of media. They play the role of a semantic anchor to deeply access to media objects.

A complete schema for generation and using the media content description for multimedia integration is also provided. Media content extractors [0] can be used to analyze and generate the media content description for each media type. The extractors analyze media objects (for instance to identify scenes and shots in a video), then the results are encoded by the descriptor and the description scheme in accordance with the adequate media content model (by example, the *VideoSegment* description scheme for a video sequence). The data row together with this description information allow the multimedia authoring system that then easily and automatically make explicit the internal time and spatial structure presentations of the media object for the semantic multimedia fragment integration. We have developed a video content editing tool called *VideoMadeus* that allows editing and visualization of the video content description. It integrates a cut detecting tool to generate the coarse-grained video content description that is then manually completed and modified through the visual interface of the tool. Such an authoring process for the video media has provided good results and encourages us to continue with other media types as text, image and audio.

Finally the media content description authoring feature that are provided through multi views (timeline, structure, attributes, etc.) could be used for other application than multimedia authoring and presentation. For instance, it could be used for Mpeg-7 multimedia content descriptions with very few adaptations because our model is much closed to Mpeg-7 descriptions and our tool can be easily coupled with analyzer tools.

References

J.F. Allen. “Maintaining Knowledge about Temporal Intervals”. *Comm. ACM*, Nov. 1983.

G. Auffret et al., “Audiovisual-based hypermedia authoring: using structured representations for efficient access to AV documents”. *Proc. ACM Hypertext '99*, pp 169-178.

R. Brunelli et al., “A Survey on the Automatic Indexing of Video Data”, *Journal of Visual and Image Representation*, n°10, pp. 78-112, 1999.

J.Carrive et al., "Using Description Logics for indexing Audiovisual Documents", Int. Workshop on description Logics, pp. 116-120, 1998.

A. Celentano, O. Gaggi, “A Synchronization Model for Hypermedia Documents Navigation”, ACM Symposium on Applied Computing 2000

N. Day, “MPEG-7 Projects and Demos”, AHG on MPEG-7 Applications and Promotions to Industry, Singapore, March 2001.

D. Hillmann, “Using Dublin Core”, DCMI Recommendations, 12 April 2001.

L. H. Hsu et al.,"A Multimedia Authoring-in-the-Large Environment to Support Complex Product Documentation", Multimedia Tools and Application 8, 11-64 (1999).

J. Hunter, "A Proposal for an MPEG-7 Description Definition Language (DDL)", *MPEG-7 AHG Test and Evaluation Meeting*, 15-19 Feb 1999, Lancaster.

J. Hunter, L. Armstrong, "A Comparison of Schemas for Video Metadata Representation", WWW8, Toronto, May 10-14, 1999.

J. Hunter et al., "MPEG-7 Harmonization with Dublin Core: Current Status and Concerns", ISO/IEC JTC1/SC29/WG11 M6160, 53rd MPEG Meeting, Beijing, July 2000

M. Jourdan et al., "A Scalable Toolkit for Designing Multimedia Authoring Environments", *Multimedia Tools and Applications Kluwer Academic Publishers*, vol. 12, num. 2/3, pp. 257-279, Nov. 2000.

C.-S. Li et al., "Multimedia Content Description in the InfoPyramid", *IEEE Inter. Conf. on Acoustics, Speech and Signal Processing (ICASSP-98)*, June, 1998.

J. Nanard et al., "Media Construction Formalism Specifying Abstractions for Multimedia Scenario Design", *NRHM*, 2001.

S.Paek et al, "Self-Describing Schemes for Interoperable MPEG-7 Multimedia Content Descriptions". *IEEE Computer Magazine*, vol. 28, pp. 23 -- 32, September 1995.

C. Roisin et al., "A Proposal for Video Modeling for Composing Multimedia Documents", *Multimedia Modeling (MMM2000)*, Nagano, Japan, 13-15 November 2000.

L. Rutledge, P. Schmitz, "Improving Media Fragment Integration in Emerging Web Formats", the 8th International Conference on Multimedia Modeling (MMM 2001), CWI, Amsterdam, The Netherlands, November 5-7, 2001.

P. Salembier and J. Smith,"MPEG-7 Multimedia Description Schemes", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 11, NO. 6, June 2001.

P. Schmitz et al., "The SMIL 2.0 Animation Modules", http://www.w3.org/TR/smil20/animation.html.

J. Saarela, "Video Content Models based on RDF", W3C workshop on "Television and the Web", Sophia-Antipolis, France, June 1998.

D. Vodislav, "Visual Programming for Animation in User Interfaces", *Proceedings of the 11th IEEE Symposium on Visual Languages (VL'95)*, Darmstadt, Germany, September 1995.

T. Wahl, K. Rothermel, "Representing Time in Multimedia-Systems", *IEEE Int. Conf. on Multimedia Computing and Systems*, May 1994.

Z. Zelenika, "CARNet Media on Demand - Metadata model", web edition 2001-05-21.

S. Boll and W. Klas, "ZYX --- A Semantic Model for Multimedia Documents and Presentations", Kluwer Academic Publishers, Rotorua, New Zealand, 5-8 January 1999.