Technologie-Zentrum Informatik

Technical Report 32

An Environment for Modelling Telecast Structures

L. Blankert, A. Jacobs, A. Miene, Th. Hermes,
G. T. Ioannidis, and O. Herzog

TZI-Bericht Nr. 32
2005

Universität Bremen

TZI-Berichte

Herausgeber:
Technologie-Zentrum Informatik
Universität Bremen
Universitätsallee 21-23
28359 Bremen
Telefon: +49-421-218-7272
Fax: +49-421-218-7820
Email: info@tzi.de
http://www.tzi.de

ISSN 1613-3773

An Environment for Modelling Telecast Structures

L. Bankert, A. Jacobs, A. Miene, Th. Hermes, G.T. Ioannidis, and O. Herzog

Center for Computing Technologies,
University of Bremen,
Universitätsallee 21-23, D-28359 Bremen, Germany,
email:{blars,jarne,andrea,hermes,gtis,herzog}@tzi.de
`http://www.tzi.de/bv`

Abstract

To automatically recognize the temporal structure of a telecast on a semantic level domain knowledge is needed. Up to now, most approaches either directly determine the telecast structure in the same step with the extraction of the features needed for this classification task or use proprietary modelling languages [SSJ93, ZGST94, GSC+95]. The resulting systems are limited to very restricted domains.

In this paper, we introduce a general modelling language that allows to describe the semantic structure of a class or series of telecasts in terms of MPEG-7 descriptors and an environment that supports the creation of structural models. It allows the decomposition into structural elements and the modelling of temporal relations between them and the specification of their visual and audible characteristics. A general modelling language yields the major advantage of separating the modelling task from the analysis and classification task.

As an example we show how the semantic structure of a magazine telecast series is modelled.

1 Introduction and Related Work

Due to the advances in both computational and storage technologies, video as a digital medium has gained popularity over the past years. Creating large digital video libraries has become feasible. However, while storage is not a problem anymore, a successful retrieval from these libraries requires the generation of structural and semantic annotations together with the videos.

While in the past the annotation process was a purely manual one, there has been made quite an effort to support it with semi-automatic or automatic techniques. A lot of work has been done on the temporal structuring of digital videos. Since the temporal structuring on the syntactical level of shot boundaries yields already quite good results (see e.g.[SO03, ABC+03]) the focus is actually switching to approaches on structuring videos on a semantical level. A first step in this direction was the introduction of shot clustering techniques to create higher-level structures based on the similarity of shots [YY96]. This was followed by attempts to create a video table of content to facilitate browsing through videos with hundreds or thousands of shots [RHM99]. A related approach called macro-segmentation was proposed by Aigrain et al. [AJL95].

However, these approaches remain highly syntactical. Although they considerably ease the work of the archivist while annotating a video, they do not provide a clue *what* the video material is about. As pointed out in [AJL95], they lack the "global understanding". While this understanding, i.e. the extraction of the semantics of parts of a video, is still impossible in general, more could be achieved with the use of a priori-knowledge about the video. In the case of temporal structuring this includes knowledge about the visible and audible elements that indicate a change of the scene or content, i.e. a structural model of the telecast.

Swanberg et al. [SSJ93] propose a model-based approach to extract semantic structure from a video, calling it video parsing. They develop a model of a CNN news broadcast. It consists of a set of semantic structural components ("header", "anchorperson", "weather", ...), a finite automaton describing the temporal composition of these components, and a set of "shot models", "frame models" and "region models" allowing to recognise a syntactic element "shot" present in the video as a part of a semantic structural component. The shots are acquired by a shot detection technique similar to those in [MDHH01, BR96].

Zhang et al. [ZGST94] extend this approach with the detection of anchorpersons and apply it to Singapore Broadcasting Corporation news programs. They classify shots into "anchorperson shots", "news shots", "commercial break shots", "weather forecast shots" and starting and ending shots, using frame and region models similar to those in [SSJ93]. These are then matched with an "episode model" describing the temporal structure of the news program. In [GSC$^+$95], the approach is adapted to soccer videos by Gong et al.

Hauptmann and Witbrock [HW98] focus on speech recognition and extraction of closed-captions to segment a news broadcast into commercials and "story"-segments, meaning those parts of the news broadcast with distinct topics. They do not use the term "model", but use an implicit model for their segmentation.

Although these approaches to video parsing show promising results, they are very specific to a certain domain, and their models are relatively fixed. They either directly determine the telecast structure in the same step with the extraction of the features needed for this classification task or use proprietary modelling languages. Up to now there exists no general framework or description language for such models.

In this paper, we introduce a general modelling language that is based on MPEG-7 and allows to decompose a type of telecast into structural elements and to model the temporal relations between these structural elements on the one hand and the visual and audible characteristics of the structural elements on the other hand. We also present an environment that allows for the creation of these structural models. As an example we show how the semantic structure of a magazine telecast can be modelled using that environment.

In contrast to languages like MPEG-7[1] [ISO01a, ISO01b, ISO01c, ISO01d, ISO01e] (see [CSP01] for a brief overview) or SMIL[2], which is also based on MPEG-7 but only allows for the description of a specific telecast instance, the proposed modelling language allows for describing classes of telecasts. This may be e.g. several telecasts of the same series, e.g. magazine or news telecasts.

The resulting models describe the structure of the telecast series in terms of MPEG-7 descriptors. The major advantage is that the modelling task is now separated from the analysis and classification task. The analysis and classification task consists of parsing a given video belonging to the modelled telecast series, extracting the features needed and matching the structure of the video with the corresponding structures defined in the model. To perform this step any analysis tool can be used that extracts the necessary MPEG-7 descriptors used within the model.

This report is organised as follows: In the next section we will identify semantic structure elements occurring in magazine telecasts, based on an example video. We will also investigate which technical structures can be used to recognise these elements, and what kind of relations there are between them. In Sec. 2.4 we will shortly discuss how the identified technical structures can be analysed automatically. A conclusion is given in Sec. 3. The appendix A contains the used XML-Scheme.

2 An Environment for Modelling Telecast Structures

In our approach, we assume that all telecast instances belonging to the same telecast series are set up in line with a given structure that belongs to the series. We also assume that there is some knowledge about the general features that are characteristic for telecasts of one telecast series. We have developed an environment that allows for modelling the structure of arbitrary telecast series. Once the structure of a telecast series is modelled it is possible to analyse arbitrary instances of the telecast series regarding its semantic structure.

We identified three important aspects of the structure of a television magazine telecast. The first is the *semantic structure* of a magazine telecast, that describes, which types of semantic features can occur in a magazine telecast. The second are the *audio-visual features*, which have a close connection to the semantic structure. There are several visual features that support the semantic structure so that it can be easily recognised by the television viewer. The third are the *relationships* among semantic features, among technical features and between semantic and technical features, respectively. This information has to be formally described in a model. Therefore the specification of a description language is needed. After that, the model can be used to analyse arbitrary telecast instances that belong to the modelled class.

[1] Online: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm, last verified on 16/10/04

[2] Online: http://www.w3.org/AudioVideo/, last verified on 16/10/04

This yields the advantage of separating the modelling step from the analysis and classification step. Since the modelling language is based on MPEG-7 descriptors, any analysis tool may be used that supports the extraction of the necessary MPEG-7 descriptors used within the model.

2.1 Representation of Structure Features

We implement our modelling language as an application on the basis of MPEG-7 [ISO01a, ISO01b, ISO01c, ISO01d, ISO01e] and its data definition language (DDL), which is *XML Schema*[3] enriched with some specific extensions. XML Schema offers some advantages to us because it supports object-oriented structures very well. Furthermore different data types can be used and can be defined, which supports the verification and validity of our description and the information contained in it better than, e.g. the document type definition (DTD) of XML.

In addition, we have to add some constructions because of one important difference between typical MPEG-7 applications and our work. MPEG-7 was specified to describe exactly one instance of a video rather than a class of videos. So we had to expand the MPEG-7 descriptors and description schemes for making descriptions possible for classes of television broadcasts.

With these extensions we are able to describe declarations like “a presentation can have a duration time between 30 seconds and two minutes” or “a telecast can contain four or five reports” and so on. The schema itself can be found in the appendix A.

2.1.1 Structure Features.

The semantic structure features can be considered as video sequences. The significant attributes for these video sequences are the type of the semantic structure feature and the coherent time interval information.

The type of the semantic structure feature can be: The whole telecast, telecast blocks (when the telecast is divided into some segments, e.g. because of commercial breaks), commercial blocks (corresponding to commercial breaks), intro, outro, preview block (contains some previews), preview, presentations (e.g. anchorman introducing the theme of the next report), report, short report block (e.g. collection of some short reports or short news), short presentation, short report or other user defined structure features.

In order to represent the time interval information, we set a link from the structure feature that refers to a time interval object. The advantage of this approach is, that we are able to set time relations between these time objects later, which is very important when modelling and representing the relationships between several semantic structure features.

To represent a corresponding time interval is useful for two aspects. On the one hand, the interval information can help to classify different semantic structure features because they may have different typical duration. On the

[3]Online: http://www.w3.org/XML/Schema, last verified on 16/10/04

other hand, the time information can be used to reduce the analysis effort. E.g., when looking for the end of an report beginning from its known starting point, it would be useless to analyse the first minute after the starting point if a report has a specified minimum duration of two minutes.

2.1.2 Audio-visual Features.

For each structure feature we have to model its ties to the audio-visual (technical) features that are characteristic for the structure feature and allow the television viewer to identify the structure. This is important because we do not only want to model the determined semantic structure, but later automatically recognise the structure of arbitrary instances of the modelled telecast series as completely as possible. This can be done by an automatic recognition of the relevant audio-visual features. Then we are able to differentiate for example a report from a presentation or to recognise, when a report begins and when it ends.

MPEG-7 allows for a very precise description of different aspects of visual features.

At first we can distinguish various types of decompositions like video sequences, still images, moving regions and still regions. For all these parts of a video, MPEG-7 allows for the description of analysable visual features in the domains of colour, texture, shape, motion, and localisation (see the visual part of MPEG-7 in [ISO01c]).

There are no universally valid rules, how each visual feature has to be described. It depends on the individual visual feature and the decision about the features to be modelled has to be made when regarding a sample video of the considered class of telecasts.

2.1.3 Relations between Features.

In general we have to distinguish different kinds of relationships:

- Relations between structure features
- Relations between audio-visual features
- Relations between structure features and audio-visual features

For the description of relationships between structure features we use a time model, which is a graph with time intervals as nodes and edges between the nodes describing the temporal relations among them. In this context, we abstract from structure features and handle them as time intervals only.

Allen developed a set of possible relationships between two time intervals [All81, AK83, All90]. These relationships are normative descriptions for temporal relationships in MPEG-7.

Figure 1 shows a sample fragment of a time graph belonging to our discussed sample telecast. MPEG-7 gives normative basics for constructing graphs. With these graph constructs we are able to express complex temporal relationships.

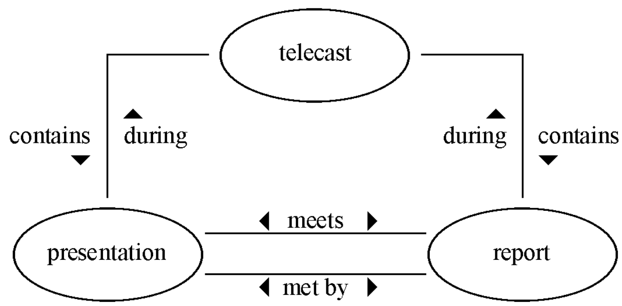

Figure 1: Temporal relationships between structure features, fragment of sample graph

We extend the normative MPEG-7 relationships to enable the declaration of cardinalities for some relationships. By doing this, our model can contain expressions like "A telecast contains at least four and maximally six reports".

Spatial Relationships between visual objects can be modelled in two ways. On one hand, it is possible to transfer the Allen relationships to multi-dimensional spaces. Mukerjee and Joe [MJ90] present an approach which uses the interval-based relationships to describe spatial relationships. While doing this, objects are represented as rectangles (in 2D). Each axis is described individually, so that Allen's relationships can be used.

Another way to describe spatial relationships is to combine topological relationships and direction relationships. Hernández (see [Her94]) proposes a corresponding approach.

We extended the description language so that both approaches can be used when specifying spatial relationships.

To link the information about the semantic structure and the technical characteristics, we describe relationships between structure features and visual features, which can be apprehended as temporal relations, too. Again we can specify a graph to describe the dependencies where we interpret the structure features as well as the visual features as time intervals which we set in relation.

Now, formal descriptions for declarations like "A characteristic frame is shown at the end of a report" are possible. By describing this kind of rules, we could draw implications between automatically analysable audio-visual features and the desired semantic meaning, the structure features.

Figure 2 shows a fragment of a corresponding sample graph.

2.2 Example "Panorama"

In this section we show how to model the semantic structure of a magazine telecast. As an example we chose the magazine telecast "Panorama" from the

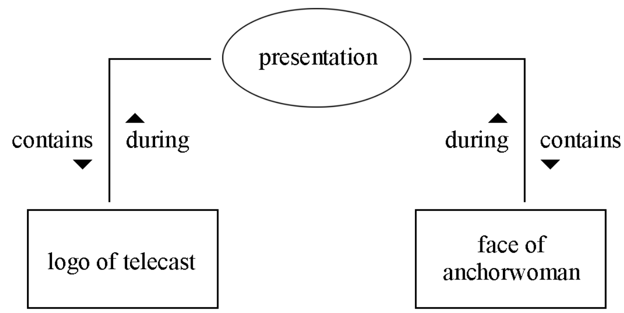

Figure 2: Relationships between structure and visual features

German broadcast station “ARD”.

2.2.1 Identification of Structure Features.

First we consider the semantic structure of the telecast. With the examined telecast “Panorama” we have an example of a television magazine, which follows a relatively simple structural pattern.

We identified the following subset of possible structural components within the concrete example telecast:

- Intro
- Previews on themes, integrated in the intro
- Presentations by an anchorwoman
- Reports
- Outro

This is modelled as follows:

```
<StructureFeatures>
 <StructureFeature id="intro" sfType="INTRO"/>
 <StructureFeature id="panorama" sfType="TELECAST"/>
 <StructureFeature id="presentation" sfType="PRESENTATION"/>
 <StructureFeature id="report" sfType="REPORT"/>
 <StructureFeature id="outro" sfType="OUTRO"/>
</StructureFeatures>
```

We can also model the expected duration of the structures by associating them with time intervals:

```
<TimeObjects>
 <TimeInterval id="panorama_ti" featureRef="panorama"
    graphNodeRef="panorama_node">
  <MaxMediaDuration>P46M</MaxMediaDuration>
  <MinMediaDuration>P44M</MinMediaDuration>
 </TimeInterval>
 <TimeInterval id="intro_ti" featureRef="intro"
    graphNodeRef="intro_node"/>
 <TimeInterval id="presentation_ti" featureRef="presentation1"
    graphNodeRef="presentation_node">
  <MaxMediaDuration>PT1M10S</MaxMediaDuration>
  <MinMediaDuration>PT15S</MinMediaDuration>
 </TimeInterval>
 <TimeInterval id="report_ti" featureRef="report"
    graphNodeRef="report_node">
  <MaxMediaDuration>PT8M56S</MaxMediaDuration>
  <MinMediaDuration>PT6M18S</MinMediaDuration>
 </TimeInterval>
 <TimeInterval id="outro_ti" FeatureRef="outro"
    graphNodeRef="outro_node"/>
</TimeObjects>
```

These time intervals can then be used to model temporal relations between structures.

2.2.2 Relations between Features.

We find out, that the arrangement of structure features follows some rules which do not differ in any instance of the the same television magazine series:

- The telecast begins with an intro, in which the previews on important topics of the telecast are integrated.
- The telecast contains a certain number (here: up to seven) of reports.
- Before and after each report a presentation is shown.
- The telecast ends with an outro.
- There is a presentation before the outro.

Figure 3 shows a structural view on the structure features of the telecast “Panorama”.

As shown before structure features are associated to time intervals. The relationships between them are described using temporal relations [All81]. The aforementioned relations can be modelled as follows:

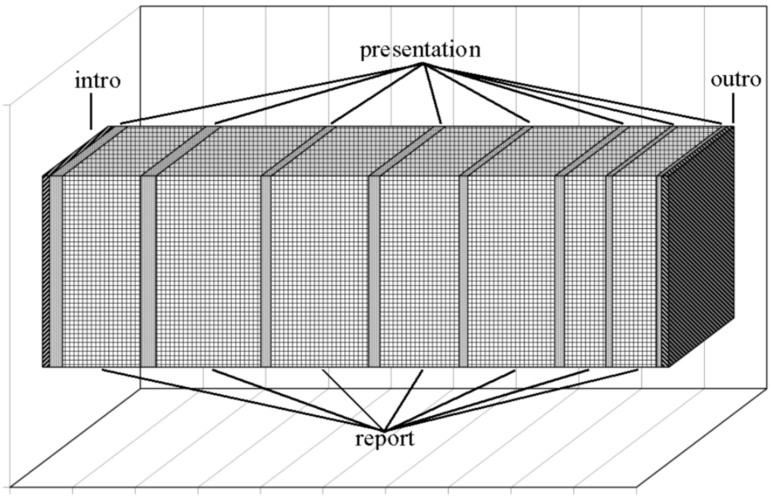

Figure 3: Structure of the telecast “Panorama”

```
<TimeRelationships>
 <TimeNode id="panorama_node" idref="panorama_ti"/>
 <TimeNode id="intro_node" idref="intro_ti"/>
 <TimeNode id="presentation_node" idref="presentation_ti"/>
 <TimeNode id="report_node" idref="report_ti"/>
 <TimeNode id="outro_node" idref="outro_ti"/>
 <TimeRelation source="panorama_node" target="intro_node"
    type="startedby"/>
 <TimeRelation source="panorama_node" target="outro_node"
    type="finishedby"/>
 <TimeRelation source="panorama_node" target="presentation_node"
    type="contains" maxCardinality="8"/>
 <TimeRelation source="panorama_node" target="report_node"
    type="contains" maxCardinality="7"/>
 <TimeRelation source="intro_node" target="presentation_node"
    type="before" directed="true"/>
 <TimeRelation source="outro_node" target="presentation_node"
    type="after" directed="true"/>
 <TimeRelation source="report_node" target="presentation_node"
    type="before" directed="true"/>
 <TimeRelation source="report_node" target="presentation_node"
    type="after" directed="true"/>
</TimeRelationships>
```

When regarding the audio-visual features, spatial relationships have to be considered. In complex visual features, that are aggregations of several visual features, the spatial dependencies have to be considered. Therefore we have to model spatial relations like "The logo is shown in the upper left area of the frame" and "The face of the anchorwoman is on the right side of the frame above a text insertion".

Relationships between structure features and visual features can be apprehended as temporal relations. We already used this nature when we described the relationships between structure features. The duration a visual feature is shown is also represented as a time interval.

Beneath the pure temporal relationships, we can interpret these relationships as logical conclusions. Some interpretations will then sound like this: "If there is a specified characteristic frame, we can assume the end of the intro." Analog procedures are adaptable for the other relationships between semantic and visual features.

By describing these rules, the structure features and their desired semantic meaning are tied to automatically analysable visual features.

2.2.3 Ties between Structure Features and Audio-visual Features.

For the telecast "Panorama" we succeed in finding some selected technical elements for separating and/or classifying the semantic elements. Separating elements allow to recognise the end or beginning of a structural component, while classifying elements provide for recognition of the kind of structural component we currently regard.

Some important visual features appearing in the examined telecast are:

- Insertion of the logo of the telecast and anchorwoman during presentations
- Finalisation of a report with a characteristic frame
- Characteristic frame at the end of the intro
- Characteristic frame at the beginning of the outro

The first visual feature during this examination is the insertion of the logo of the television telecast during the presentations. From the technical point of view, there are two important elements occurring during the presentations. On the one hand there is the logo of the telecast in the upper left area during the presentation (and only during the presentations, except of the outro). On the other hand a presentation is characterised by the fact that the anchorwoman is to be seen in the frames or at least in the majority of the frames (see Fig. 4 at the right).

Other identified visual features are the characteristic frame at the end of the intro (see Fig. 4 at the left) and the characteristic frame at the beginning of the outro (see Fig. 5 at the right). Another example is the anchorwoman scene during presentations shown in Fig. 5 at the left.

Figure 4: Visual features: Last frame of the intro with characteristic colouring of the whole frame and logo of the telecast (*left*) and technical elements of a presentation. The logo of the telecast and the presenter are shown somewhere in the frame (*right*)

Figure 5: Visual features: End of a report (*left*): characteristic background colours, logo of the telecast, text. Begin of the outro (*right*): characteristic colours, logo of the telecast

We describe the visual feature representation of the finalisation of a report with a characteristic frame (see Fig. 5 in the left). We chose three different regions as features for identifying the shown frame. First of all there is a characteristic dark-blue background. Additional visual elements are the logo of the telecast series in the center of the frame and a text insertion below this logo.

In this case, the description of colour aspects is very important. The colour information itself (colour values) is characteristic, which guides us to the description of dominant colours in the frame. Another aspect of colour is, that the white coloured pixels build very compact colour regions. White pixels are not distributed all over the frame but are strongly connected in several small regions. To describe this fact, the dominant colour descriptor has an attribute to describe this spatial connectivity.

Other visual features may be much more complex, but the approach of selecting visual features and describing them is the same. For other visual features it may be more useful to describe the texture or the shape of a characteristic object. At last, the complexity of the description and the use of several features has to depend on each concrete sample. Furthermore, it has to be considered, that the description is powerful enough to distinguish the different characteristic visual features well. Otherwise the results of the analysis and the resulting semantic structure will contain segmentation errors.

The following paragraph shows how the above mentioned visual features can be modelled. The specific values of the color and shape descriptors are omitted.

```
<AudioVisualFeatures>
 <StillRegion id="darkbackground">
  <VisualDescriptor xsi:type="mpeg7:DominantColorType">
   ...
  </VisualDescriptor>
 <StillRegion id="biglogo">
  <VisualDescriptor xsi:type="mpeg7:ContourShapeType">
   ...
  </VisualDescriptor>
 </StillRegion>
 <StillRegion id="smalllogo">
  <VisualDescriptor xsi:type="mpeg7:ContourShapeType">
   ...
  </VisualDescriptor>
 </StillRegion>
 <StillRegion id="face">
  <VisualDescriptor xsi:type="SimpleFaceRecognitionType">
   <HasFace>true</HasFace>
  </VisualDescriptor>
 </StillRegion>
 <StillRegion id="outrobackground">
  <VisualDescriptor xsi:type="mpeg7:DominantColorType">
   ...
```

```
  </VisualDescriptor>
 </StillRegion>
</AudioVisualFeatures>
<TimeModel>
 <TimeObject>
  <TimeInterval id="biglogo_ti" featureRef="biglogo"
    graphNodeRef="biglogo_node"/>
  <TimeInterval id="darkbackground_ti" featureRef="darkbackground"
    graphNodeRef="darkbackground_node"/>
  <TimeInterval id="smalllogo_ti" featureRef="smalllogo"
    graphNodeRef="smalllogo_node"/>
  <TimeInterval id="face_ti" featureRef="face"
    graphNodeRef="face_node"/>
  <TimeInterval id="outrobackground_ti" featureRef="outrobackground"
    graphNodeRef="outrobackground_node"/>
 </TimeObjects>
 <TimeRelationships>
  <TimeNode id="biglogo_node" idref="biglogo_ti"/>
  <TimeNode id="darkbackground_node" idref="darkbackground_ti"/>
  <TimeNode id="smalllogo_node" idref="lofo2_ti"/>
  <TimeNode id="face_node" idref="face_ti"/>
  <TimeNode id="outrobackground_node idref="outrobackground_ti"/>
  <TimeRelation source="intro_node" target="biglogo_node"
    type="finishedby"/>
  <TimeRelation source="intro_node" target="darkbackground_node"
    type="finishedby"/>
  <TimeRelation source="presentation_node" target="smalllogo_node"
    type="contains"/>
  <TimeRelation source="presentation_node" target="face_node"
    type="contains"/>
  <TimeRelation source="report_node" target="biglogo_node"
    type="finishedby"/>
  <TimeRelation source="report_node" target="darkbackground_node"
    type="finishedby"/>
  <TimeRelation source="outro_node" target="smalllogo_node"
    type="startedby"/>
  <TimeRelation source="outro_node" target="outrobackground_node"
    type="startedby"/>
 </TimeRelationships>
</TimeModel>
```

2.3 Example “Report Mainz”

To further show the applicability of our model, we looked at another example, the telecast “Report Mainz” from the German broadcast station “SWR”. It follows a very similar structure: It begins with an intro, which ends with a

Figure 6: Visual features in “Report Mainz”: Last frame of the intro with characteristic colouring of the whole frame and logo of the telecast “Report Mainz” (*left*) and a presentation with an anchorman in the frame (*right*)

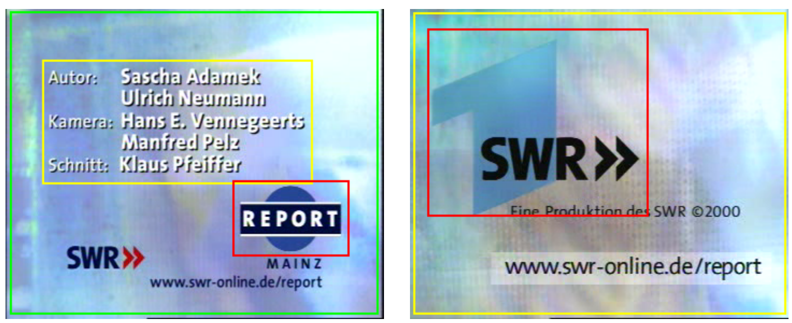

Figure 7: Visual features in “Report Mainz”: End of a report (*left*): characteristic background colours, logo of the telecast, text. End of the outro (*right*): characteristic colours, logo of the telecast

characteristic frame. After the introduction, a number of reports follow, each of which is preceded and followed by presentations. Each report ends with a characteristic frame. The outro ends with a characteristic frame, too.

Figure 6 shows visual features characteristic for the introduction and a presentation, respectively. Figure 7 shows corresponding visual features for the end of a report and the end of the outro.

The model for the telecast “Panorama” can easily be adopted to suit the telecast “Report Mainz”. Only one temporal relation has to be changed (there is a characteristic frame at the end of the outro, instead of at the beginning). The color and shape descriptions for the visual features have to be changed appropriately.

2.4 Analysis on MPEG-7 Descriptors

Since the modelling step is now seperated from the analysis and classification step any approach on extracting the descriptor values specified in the model may be used.

For lots of visual feature representations there are some notes for extracting the descriptor values in the visual part of MPEG-7 (also see [ISO01c]).

Furthermore a lot of work has been done concerning the extraction of MPEG-7 visual descriptors. Ohm et al. [OCK+02] give a good introduction in colour descriptors, covering the extraction methods for colour features. The texture descriptors and suitable methods to extract texture features for MPEG-7 descriptions are discussed by Choi et al. [CWRM02]. Giving hints in extracting values for shape descriptors is the goal of Bober et al. [BPK02] related to the MPEG-7 visual part. Jeannin et al. [JDM02] consider the MPEG-7 motion description and related motion analysis.

3 Conclusion

In this work we presented our approach on modelling semantic video structures. We introduced our modelling language which allows the creation of models for any class or series of telecasts with a significant and distinct semantic structure. Our modelling language is based on the MPEG-7 standard and we explained our extensions to this standard in order to create a model of a telecast series or class instead of a single instance of a telecast.

We also gave an example on how to create a model for the magazine telecast series "Panorama" using our environment.

Our future work will concentrate on the automatic matching between a model and corresponding videos of a certain telecast series or class. Therefore future work will also be done concerning the development and integration of tools for an automatic extraction of MPEG-7 descriptors.

Acknowledgement

This work was partly supported by the research project AVAnTA [MH00] which is funded by the German Research Foundation (DFG) within the strategic research initiative No. 1041 "V3D2 - Distributed Processing and Delivery of Digital Documents".

References

[ABC+03] Arnon Amir, Marco Berg, Shih-Fu Chang, Winston Hsu, Giridharan Iyengar, Ching-Yung Lin, Milind Naphade, Apostol (Paul) Natsev, Chalapathy Neti, Harriet Nock, John R. Smith, Belle Tseng, Yi Wu, and Donqing Zhang. Ibm research trec-2003 video retrieval system. In *Proceedings of TREC Video Retrieval Evaluation (TRECVID 2003). Gaithersburg, MD, NIST.*, November 2003.

[AJL95] P. Aigrain, P. Joly, and V. Longueville. Mediumknowledge-based macro-segmentation of video into sequences, 1995.

[AK83] J. F. Allen and J. A. Koomen. Planning using a temporal world model. In *Proc. of the 8th IJCAI*, pages 741–747, Karlsruhe, Germany, 1983.

[All81] J. F. Allen. An interval-based representation of temporal knowledge. In *Proc. of the 7th IJCAI*, pages 221–226, Vancouver, Canada, 1981.

[All90] J. F. Allen. Towards a general theory of action and time. In J. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 464–479. Kaufmann, San Mateo, CA, 1990.

[BPK02] Miroslaw Bober, F. Preteux, and Whoi-Yul Kim. Shape descriptors. In Thomas Sikora B. S. Manjunath, Philippe Salembier, editor, *Introduction to MPEG-7*, chapter 15. John Wiley and Sons, Ltd., 2002.

[BR96] John S. Boresczkky and Lawrence A. Rowe. A comparison of video shot boundary detection techniques. In *Proceedings of the IS-T/SPIE Conference on Storage and Retrieval for Image and Video Databases IV*, volume 2664, pages 170–179, San Diego/La Jolla, CA, USA, feb 1996.

[CSP01] Shih-Fu Chang, Thomas Sikora, and Atul Puri. Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[CWRM02] Yanglim Choi, Chee Sun Won, Yong Man Ro, and B. S. Manjunath. Texture descriptors. In Thomas Sikora B. S. Manjunath, Philippe Salembier, editor, *Introduction to MPEG-7*, chapter 14. John Wiley and Sons, Ltd., 2002.

[GSC+95] Yihong Gong, Lim Teck Sin, Chua Hock Chuan, HongJiang Zhang, and Masao Sakauchi. Automatic parsing of tv soccer programs. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 167–174, may 1995.

[Her94] D. Hernández. *Qualitative Representation of Spatial Knowledge.* Springer, Berlin, Heidelberg, 1994.

[HW98] Alexander G. Hauptmann and Michael J. Witbrock. Story segmentation and detection of commercials in broadcast news video. In *Advances in Digital Libraries*, pages 168–179, 1998.

[ISO01a] ISO/IEC. 15938-1:2001: Multimedia content description interface - part 1 systems, 2001. Version 1.

[ISO01b] ISO/IEC. 15938-2:2001: Multimedia content description interface - part 2 description definition language, 2001. Version 1.

[ISO01c] ISO/IEC. 15938-3:2001: Multimedia content description interface - part 3 visual, 2001. Version 1.

[ISO01d] ISO/IEC. 15938-4:2001: Multimedia content description interface - part 4 audio, 2001. Version 1.

[ISO01e] ISO/IEC. 15938-5:2001: Multimedia content description interface - part 5 multimedia description schemes, 2001. Version 1.

[JDM02] Sylvie Jeannin, Ajay Divakaran, and Benoit Mory. Motion descriptors. In Thomas Sikora B. S. Manjunath, Philippe Salembier, editor, *Introduction to MPEG-7*, chapter 16. John Wiley and Sons, Ltd., 2002.

[MDHH01] A. Miene, A. Dammeyer, Th. Hermes, and O. Herzog. Advanced and adapted shot boundary detection. In D. W. Fellner, N. Fuhr, and I. Witten, editors, *Proc. of ECDL WS Generalized Documents*, pages 39–43, 2001.

[MH00] Andrea Miene and Otthein Herzog. AVAnTA - Automatische Video Analyse und textuelle Annotation. *it + ti Informationstechnik und Technische Informatik*, 42(6):24 – 27, 2000.

[MJ90] A. Mukerjee and G. Joe. A qualitative model for space. In *Proc. of AAAI-90*, pages 721–727, Boston, MA, 1990.

[OCK+02] Jens-Rainer Ohm, Leszek Cieplinski, Heon J. Kim, Santhana Krishnamachari, B. S. Manjunath, Dean S. Messing, and Akio Yamada. Color descriptors. In Thomas Sikora B. S. Manjunath, Philippe Salembier, editor, *Introduction to MPEG-7*, chapter 13. John Wiley and Sons, Ltd., 2002.

[RHM99] Yong Rui, Thomas S. Huang, and Sharad Mehrotra. Constructing table-of-content for videos. *Multimedia Systems*, 7(5):359–368, 1999.

[SO03] A. Smeaton and P. Over. Trec 2003 video retrieval evaluation overview. presentation slides, November, 17 2003.

[SSJ93] Deborah Swanberg, Chiao-Fe Shu, and Ramesh Jain. Knowledge guided parsing in video databases. In *Proceedings of the IS-T/SPIE Conference on Storage and Retrieval for Image and Video Databases*, volume 1908, pages 13–24, San Jose, CA, USA, 1993.

[YY96] M.M. Yeung and B.L. Yeo. Time-constrained Clustering for Segmentation of Video into Story Units. In *International Conference on Pattern Recognition*, August 1996.

[ZGST94] HongJiang Zhang, Yihong Gong, Stephen W. Smoliar, and Shuang Yeo Tan. Automatic parsing of news video. In *Proceedings of the International Conference on Multimedia Computing and Systems*, pages 45–54, may 1994.

A Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    targetNamespace="urn:tzi:xbml:schema:2003"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:mpeg7="urn:mpeg:mpeg7:schema:2001"
    xmlns="urn:tzi:xbml:schema:2003"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

 <xs:element name="Xbml">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="Header" type="HeaderType"/>
    <xs:element name="Structure" minOccurs="0"
        type="StructureType"/>
    <xs:element name="AudioVisual" minOccurs="0"
        type="AudioVisualType"/>
    <xs:element name="TextInsertions" minOccurs="0"
        type="TextInsertionsType"/>
    <xs:element name="TimeModel" minOccurs="0"
        type="TimeModelType"/>
   </xs:sequence>
   <xs:attribute name="id" type="xs:ID" use="optional"/>
  </xs:complexType>
 </xs:element>

 <xs:complexType name="HeaderType">
  <xs:sequence>
   <xs:element name="Comment" minOccurs="0"
        maxOccurs="unbounded" type="xs:string"/>
   <xs:element name="PeriodBegin" minOccurs="0"
        type="mpeg7:timePointType"/>
   <xs:element name="PeriodEnd" minOccurs="0"
        type="mpeg7:timePointType"/>
   <xs:element name="Telestation" minOccurs="0"
        type="xs:string"/>
   <xs:element name="CreationInformation" minOccurs="0"
        type="mpeg7:CreationInformationType"/>
  </xs:sequence>
```

```
 <xs:attribute name="id" type="xs:ID" use="required"/>
 <xs:attribute name="name" type="xs:string" use="optional"/>
</xs:complexType>

<xs:complexType name="StructureType">
 <xs:sequence>
  <xs:element name="StructureFeatures" minOccurs="0"
       type="StructureFeaturesType"/>
  <xs:element name="TimeModel" minOccurs="0"
       type="TimeModelType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="StructureFeaturesType">
 <xs:sequence>
  <xs:element name="StructureFeature" minOccurs="0"
       maxOccurs="unbounded" type="StructureFeatureType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="StructureFeatureType">
 <xs:sequence>
  <xs:element name="Description" minOccurs="0"
       type="xs:string"/>
  <xs:choice>
   <xs:sequence>
    <xs:element name="TimeObjectStructureRef"
       minOccurs="0" type="xs:IDREF"/>
    <xs:element name="TimeObjectAudioVisualStructureRef"
       minOccurs="0" type="xs:IDREF"/>
   </xs:sequence>
  </xs:choice>
 </xs:sequence>
 <xs:attribute name="name" use="optional"
       type="xs:string"/>
 <xs:attribute name="sfType" use="required"
       type="StructureFeatureTypeType"/>
 <xs:attribute name="id" use="required"
       type="xs:ID"/>
</xs:complexType>

<xs:simpleType name="StructureFeatureTypeType">
 <xs:restriction base="xs:string">
  <xs:enumeration value="TELECAST"/>
  <xs:enumeration value="TELECASTBLOCK"/>
```

```
  <xs:enumeration value="COMMERCIALBLOCK"/>
  <xs:enumeration value="INTRO"/>
  <xs:enumeration value="OUTRO"/>
  <xs:enumeration value="PREVIEWBLOCK"/>
  <xs:enumeration value="PREVIEW"/>
  <xs:enumeration value="THEMEBLOCK"/>
  <xs:enumeration value="PRESENTATION"/>
  <xs:enumeration value="REPORT"/>
  <xs:enumeration value="SHORTREPORTBLOCK"/>
  <xs:enumeration value="SHORTPRESENTATION"/>
  <xs:enumeration value="SHORTREPORT"/>
  <xs:enumeration value="USERDEFINEDBLOCK"/>
  <xs:enumeration value="USERDEFINEDSEQUENCE"/>
 </xs:restriction>
</xs:simpleType>

<xs:complexType name="AudioVisualType">
 <xs:sequence>
  <xs:element name="AudioVisualFeatures" minOccurs="0"
       type="AudioVisualFeaturesType"/>
  <xs:element name="TimeModel" minOccurs="0"
       type="TimeModelType"/>
  <xs:element name="SpatialModel" minOccurs="0"
       type="SpatialModelType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="AudioVisualFeaturesType">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="StillRegion" minOccurs="0"
       maxOccurs="unbounded" type="StillRegionType"/>
  <xs:element name="MovingRegion" minOccurs="0"
       maxOccurs="unbounded" type="MovingRegionType"/>
  <xs:element name="VideoSegment" minOccurs="0"
       maxOccurs="unbounded" type="VideoSegmentType"/>
  <xs:element name="AudioSegment" minOccurs="0"
       maxOccurs="unbounded" type="AudioSegmentType"/>
 </xs:choice>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="StillRegionType">
 <xs:complexContent>
  <xs:restriction base="mpeg7:StillRegionType">
   <xs:sequence>
    <xs:element name="Name" minOccurs="0"
```

```
           type="xs:string"/>
    <xs:sequence minOccurs="0">
     <xs:element name="TimeObjectAudioVisualRef"
           type="xs:IDREF" minOccurs="0"/>
     <xs:element name="TimeObjectAudioViusalStructureRef"
           type="xs:IDREF" minOccurs="0"/>
     <xs:element name="SpatialObjectAudioVisualRef"
           type="xs:IDREF" minOccurs="0"/>
    </xs:sequence>
    <xs:choice minOccurs="0">
     <xs:element name="SpatialLocator"
           type="mpeg7:RegionLocatorType"/>
     <xs:element name="SpatialMask"
           type="mpeg7:SpatialMaskType"/>
    </xs:choice>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
     <xs:element name="VisualDescriptor"
           type="mpeg7:VisualDType"/>
     <xs:element name="VisualDescriptionScheme"
           type="mpeg7:VisualDSType"/>
     <xs:element name="GridLayoutDescriptors"
           type="mpeg7:GridLayoutType"/>
    </xs:choice>
   </xs:sequence>
  </xs:restriction>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="MovingRegionType">
 <xs:complexContent>
  <xs:restriction base="mpeg7:MovingRegionType">
   <xs:sequence>
    <xs:element name="Name" minOccurs="0"
       type="xs:string"/>
    <xs:sequence minOccurs="0">
     <xs:element name="TimeObjectAudioVisualRef"
           type="xs:IDREF" minOccurs="0"/>
     <xs:element name="TimeObjectAudioVisualStructureRef"
           type="xs:IDREF" minOccurs="0"/>
     <xs:element name="SpatialObjectAudioVisualRef"
           type="xs:IDREF" minOccurs="0"/>
    </xs:sequence>
    <xs:choice minOccurs="0">
     <xs:element name="SpatioTemporalLocator"
           type="mpeg7:SpatioTemporalLocatorType"/>
     <xs:element name="SpatioTemporalMask"
```

```
          type="mpeg7:SpatioTemporalMaskType"/>
    </xs:choice>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
     <xs:element name="VisualDescriptor"
          type="mpeg7:VisualDType"/>
     <xs:element name="VisualDescriptionScheme"
          type="mpeg7:VisualDSType"/>
     <xs:element name="VisualTimeSeriesDescriptor"
          type="mpeg7:VisualTimeSeriesType"/>
    </xs:choice>
   </xs:sequence>
  </xs:restriction>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="VideoSegmentType">
 <xs:complexContent>
  <xs:restriction base="mpeg7:VideoSegmentType">
   <xs:sequence>
    <xs:element name="Name" type="xs:string" minOccurs="0"/>
    <xs:sequence minOccurs="0">
     <xs:element name="TimeObjectAudioVisualRef"
          type="xs:IDREF" minOccurs="0"/>
     <xs:element name="TimeObjectAudioVisualStructureRef"
          type="xs:IDREF" minOccurs="0"/>
     <xs:element name="SpatialObjectAudioVisualRef"
          type="xs:IDREF" minOccurs="0"/>
    </xs:sequence>
    <xs:choice minOccurs="0">
     <xs:element name="MediaTime"
          type="mpeg7:MediaTimeType"/>
     <xs:element name="TemporalMask"
          type="mpeg7:TemporalMaskType"/>
    </xs:choice>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
     <xs:element name="VisualDescriptor"
          type="mpeg7:VisualDType"/>
     <xs:element name="VisualDescriptionScheme"
          type="mpeg7:VisualDSType"/>
     <xs:element name="VisualTimeSeriesDescriptor"
          type="mpeg7:VisualTimeSeriesType"/>
    </xs:choice>
    <xs:element name="Mosaic" minOccurs="0"
       maxOccurs="unbounded" type="mpeg7:MosaicType"/>
   </xs:sequence>
  </xs:restriction>
```

```
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="AudioSegmentType">
 <xs:complexContent>
  <xs:restriction base="mpeg7:AudioSegmentType">
   <xs:sequence>
    <xs:element name="Name" minOccurs="0"
       type="xs:string"/>
    <xs:sequence minOccurs="0">
     <xs:element name="TimeObjectAudioVisualRef"
           minOccurs="0" type="xs:IDREF"/>
     <xs:element name="TimeObjectAudioVisualStructureRef"
           minOccurs="0" type="xs:IDREF"/>
    </xs:sequence>
    <xs:choice minOccurs="0">
     <xs:element name="MediaTime"
           type="mpeg7:MediaTimeType"/>
     <xs:element name="TemporalMask"
           type="mpeg7:TemporalMaskType"/>
    </xs:choice>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
     <xs:element name="AudioDescriptor"
           type="mpeg7:AudioDType"/>
     <xs:element name="AudioDescriptionScheme"
           type="mpeg7:AudioDSType"/>
    </xs:choice>
    <xs:element name="Name" minOccurs="0"
       type="xs:string"/>
   </xs:sequence>
  </xs:restriction>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="SimpleFaceDetectionType">
 <xs:complexContent>
  <xs:extension base="mpeg7:VisualDType">
   <xs:sequence>
    <xs:element name="hasFace" type="xs:boolean"/>
    <xs:element name="minQuantity" minOccurs="0"
       type="xs:nonNegativeInteger"/>
    <xs:element name="maxQuantity" minOccurs="0"
       type="xs:nonNegativeInteger"/>
    <xs:element name="minSize" minOccurs="0"
       type="mpeg7:zeroToOneType"/>
    <xs:element name="maxSize" minOccurs="0"
```

```
       type="mpeg7:zeroToOneType"/>
   </xs:sequence>
   <xs:attribute name="id" use="optional"
       type="xs:ID"/>
   <xs:attribute name="name" use="optional"
       type="xs:string"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="TextInsertionsType">
 <xs:sequence>
  <xs:element name="TextInsertion" minOccurs="0"
       maxOccurs="unbounded" type="TextInsertionType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="TextInsertionType">
 <xs:sequence>
  <xs:element name="Duration" minOccurs="0"
       type="TimeIntervalType"/>
  <xs:element name="Visual" minOccurs="0">
   <xs:complexType>
    <xs:complexContent>
     <xs:restriction base="AudioVisualType">
      <xs:sequence>
       <xs:element name="VisualFeatures"
           type="VisualFeaturesType"/>
       <xs:element name="TimeModel" minOccurs="0"
           type="TimeModelType"/>
       <xs:element name="SpatialModel" minOccurs="0"
           type="SpatialModelType"/>
      </xs:sequence>
     </xs:restriction>
    </xs:complexContent>
   </xs:complexType>
  </xs:element>
  <xs:element name="Texts"
       type="TextsType"/>
  <xs:element name="Context" minOccurs="0"
       type="ContextType"/>
  <xs:element name="TimeObjectAudioVisualStructureRef"
       minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="id" use="required"
```

```
       type="xs:ID"/>
 <xs:attribute name="name" use="optional"
       type="xs:string"/>
</xs:complexType>

<xs:complexType name="VisualFeaturesType">
 <xs:complexContent>
  <xs:restriction base="AudioVisualFeaturesType">
   <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="StillRegion" minOccurs="0"
       maxOccurs="unbounded" type="StillRegionType"/>
    <xs:element name="MovingRegion" minOccurs="0"
       maxOccurs="unbounded" type="MovingRegionType"/>
    <xs:element name="VideoSegment" minOccurs="0"
       maxOccurs="unbounded" type="VideoSegmentType"/>
   </xs:choice>
  </xs:restriction>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="TextsType">
 <xs:sequence minOccurs="0">
  <xs:element name="Text" minOccurs="0"
       maxOccurs="unbounded" type="TextType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="ContextType">
 <xs:sequence>
  <xs:element name="StructureFeatureRef" minOccurs="0"
       maxOccurs="unbounded" type="StructureFeatureRefType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="TextType">
 <xs:sequence>
  <xs:element name="TextRegion"
       type="mpeg7:RegionLocatorType"/>
  <xs:element name="TextSemantic" minOccurs="0"
       maxOccurs="unbounded" type="TextSemanticType"/>
  <xs:element name="TextColor" minOccurs="0"
       type="TextColorType"/>
 </xs:sequence>
 <xs:attribute name="id" use="required"
       type="xs:ID"/>
```

```
 <xs:attribute name="name" use="optional"
       type="xs:string"/>
 <xs:attribute name="minRows" use="optional"
       type="xs:positiveInteger"/>
 <xs:attribute name="maxRows" use="optional"
       type="xs:positiveInteger"/>
 <xs:attribute name="optional" use="optional"
       type="xs:boolean"/>
 <xs:attribute name="scrollingText" use="optional"
       type="xs:boolean"/>
 <xs:attribute name="scrollingTextDir" use="optional"
       type="xs:positiveInteger"/>
</xs:complexType>

<xs:complexType name="DurationType">
 <xs:sequence>
  <xs:element name="MinMediaDuration" minOccurs="0"
       type="mpeg7:mediaDurationType"/>
  <xs:element name="MaxMediaDuration" minOccurs="0"
       type="mpeg7:mediaDurationType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="StructureFeatureRefType">
 <xs:attribute name="id" type="xs:ID" use="optional"/>
 <xs:attribute name="name" type="xs:string" use="optional"/>
 <xs:attribute name="idref" type="xs:IDREF" use="required"/>
</xs:complexType>

<xs:complexType name="TextSemanticType">
 <xs:sequence>
  <xs:element name="Context" minOccurs="0"
       type="ContextType"/>
  <xs:element name="Meaning" maxOccurs="unbounded"
       type="xs:string"/>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="TextColorType">
 <xs:complexContent>
  <xs:extension base="mpeg7:VisualDType">
   <xs:sequence>
    <xs:element name="ColorSpace" minOccurs="0"
       type="mpeg7:ColorSpaceType"/>
    <xs:element name="ColorQuantization" minOccurs="0"
       type="mpeg7:ColorQuantizationType"/>
```

```
    <xs:element name="Value" maxOccurs="8">
     <xs:complexType>
      <xs:sequence>
       <xs:element name="Index">
        <xs:simpleType>
         <xs:restriction>
          <xs:simpleType>
           <xs:list itemType="mpeg7:unsigned12"/>
          </xs:simpleType>
          <xs:length value="3"/>
         </xs:restriction>
        </xs:simpleType>
       </xs:element>
       <xs:element name="ColorVariance" minOccurs="0">
        <xs:simpleType>
         <xs:restriction>
          <xs:simpleType>
           <xs:list itemType="mpeg7:unsigned1"/>
          </xs:simpleType>
          <xs:length value="3"/>
         </xs:restriction>
        </xs:simpleType>
       </xs:element>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="TimeModelType">
 <xs:sequence>
  <xs:element name="TimeObjects" minOccurs="0"
       type="TimeObjectsType"/>
  <xs:element name="TimeRelationships" minOccurs="0"
       type="TimeGraphType"/>
 </xs:sequence>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="TimeObjectsType">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:element name="TimeInterval" minOccurs="0"
       maxOccurs="unbounded" type="TimeIntervalType"/>
  <xs:element name="TimeMoment" minOccurs="0"
```

```
      maxOccurs="unbounded" type="TimeMomentType"/>
 </xs:choice>
 <xs:attribute name="id" type="xs:ID" use="optional"/>
</xs:complexType>

<xs:complexType name="TimeIntervalType">
 <xs:sequence>
  <xs:element name="StartPoint" minOccurs="0"
      type="mpeg7:mediaTimePointType"/>
  <xs:element name="EndPoint" minOccurs="0"
      type="mpeg7:mediaTimePointType"/>
  <xs:element name="MaxMediaDuration" minOccurs="0"
      type="mpeg7:mediaDurationType"/>
  <xs:element name="MinMediaDuration" minOccurs="0"
      type="mpeg7:mediaDurationType"/>
 </xs:sequence>
 <xs:attribute name="id" use="required"
      type="xs:ID"/>
 <xs:attribute name="featureRef" use="optional"
      type="xs:IDREF"/>
 <xs:attribute name="graphNodeRef" use="optional"
      type="xs:IDREF"/>
 <xs:attribute name="name" use="optional"
      type="xs:string" />
</xs:complexType>

<xs:complexType name="TimeMomentType">
 <xs:sequence>
  <xs:element name="StartPoint" minOccurs="0"
      type="mpeg7:mediaTimePointType"/>
  <xs:element name="EndPoint" minOccurs="0"
      type="mpeg7:mediaTimePointType"/>
  <xs:element name="MaxMediaDuration" minOccurs="0"
      type="mpeg7:mediaDurationType"/>
  <xs:element name="MinMediaDuration" minOccurs="0"
      type="mpeg7:mediaDurationType"/>
 </xs:sequence>
 <xs:attribute name="id" use="required"
      type="xs:ID"/>
 <xs:attribute name="featureRef" use="optional"
      type="xs:IDREF"/>
 <xs:attribute name="graphNodeRef" use="optional"
      type="xs:IDREF"/>
 <xs:attribute name="name" use="optional"
      type="xs:string"/>
</xs:complexType>
```

```
<xs:complexType name="TimeGraphType">
 <xs:complexContent>
  <xs:extension base="mpeg7:DSType">
   <xs:sequence>
    <xs:element name="TimeNode" minOccurs="0"
       maxOccurs="unbounded" type="NodeType"/>
    <xs:element name="TimeRelation" minOccurs="0"
       maxOccurs="unbounded" type="TimeRelationType"/>
   </xs:sequence>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="TimeRelationType">
 <xs:complexContent>
  <xs:extension base="mpeg7:RelationType">
   <xs:attribute name="minCardinality" use="optional"
       type="xs:integer"/>
   <xs:attribute name="maxCardinality" use="optional"
       type="xs:integer"/>
  </xs:extension>
 </xs:complexContent>
</xs:complexType>

<xs:complexType name="SpatialModelType">
 <xs:sequence>
  <xs:element name="SpatialObjects">
   <xs:complexType>
    <xs:sequence>
     <xs:element name="SpatialObject" minOccurs="0"
       maxOccurs="unbounded"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>
  <xs:element name="Relationships"
       type="SpatialRelationshipsType"/>
 </xs:sequence>
 <xs:attribute name="id" use="optional"
       type="xs:ID"/>
</xs:complexType>

<xs:complexType name="SpatialRelationshipsType">
 <xs:complexContent>
  <xs:extension base="mpeg7:DSType">
   <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Node" type="NodeType"/>
    <xs:element name="Relation" minOccurs="0"
```

```
        maxOccurs="unbounded"
        type="SpatialRelationGuesgenType"/>
    </xs:choice>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>

 <xs:complexType name="SpatialRelationGuesgenType">
  <xs:sequence>
   <xs:element name="XRelation" minOccurs="0"
        type="mpeg7:RelationType"/>
   <xs:element name="YRelation" minOccurs="0"
        type="mpeg7:RelationType"/>
  </xs:sequence>
  <xs:attribute name="id" use="optional"
        type="xs:ID"/>
 </xs:complexType>

 <xs:complexType name="NodeType">
  <xs:complexContent>
   <xs:extension base="mpeg7:ReferenceType">
    <xs:attribute name="id" use="required"
        type="xs:ID"/>
    <xs:attribute name="name" use="optional"
        type="xs:string"/>
   </xs:extension>
  </xs:complexContent>
 </xs:complexType>
</xs:schema>
```