



8

Description of a Single Multimedia Document

Ana B. Benitez¹, José M. Martinez², Hawley Rising³ and Philippe Salembier⁴

¹Columbia University, New York, USA, ²Universidad Politécnica de Madrid, Madrid, Spain, ³Sony Network and Software Solutions Center of America, San Jose, California, USA, ⁴Universitat Politècnica de Catalunya, Barcelona, Spain

8.1 INTRODUCTION

MPEG-7 provides description tools for describing one multimedia document. Multimedia documents supported by MPEG-7 are not only typical audiovisual documents such as image, videos, audio sequences and audiovisual sequences but also special types of multimedia documents such as ink content, MPEG-4 presentations and Web pages with different media (e.g. images and text). These tools are applicable to multimedia content of any format including digital and analog multimedia content.

In this chapter, we present the MPEG-7 tools that describe the management, structure and semantics of a single multimedia document. The management tools describe the creation, media and usage of the content. The structural tools describe spatial and/or temporal segments of multimedia content, their attributes and the relationships among them. Examples of segments are still regions in images and a set of frames in videos. The semantic tools describe semantic entities in narrative worlds depicted by multimedia content together with their attributes and the relationships among them. Examples of semantic entities are objects, events and concepts. The tools described in this chapter use tools that are described in other chapters of the book such as audio description tools, video description tools, basic description tools [8] and collection description tools [2]. Some of these tools can also describe certain aspects of multimedia collections such as the creation date, the media format and rights information, among others [2].

An example of a content management, structure and semantic description of an image is illustrated by Figure 8.1. In this example, the creation information tools specify the photographer who took the picture as well as the place and time where and when

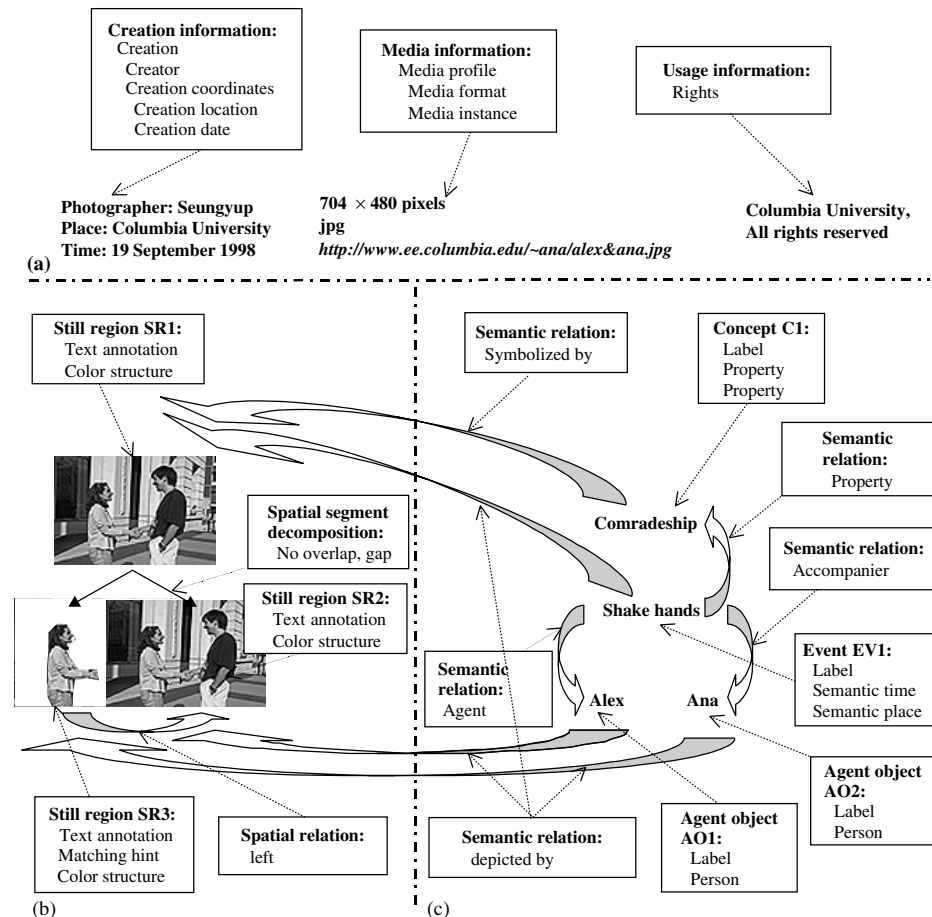


Figure 8.1 Example of a content management: (a) structure; (b) semantic; and (c) description of an image

the photograph was taken, respectively. The size, coding format and locator of the image is described using the media information tools. Rights information is represented by the usage information tools. The image and two regions of the image are described using still region tools. The decomposition of the image into the two regions and a spatial relationship between these two regions are captured using segment decomposition tools and structural relation tools, respectively. In terms of semantics, two persons, an event and a concept in the image are described using semantic entity tools. Several relationships among these semantic entities are characterized using semantic relation tools. The description in Extensible Markup Language (XML) format for this and the other examples in this chapter is included in the corresponding appendix stored on the DVD accompanying this book.

8.2 CONTENT MANAGEMENT

The content management tools allow the description of the life cycle of multimedia content from its creation to its usage through encoding. They are organized in three

main areas dealing with the description of the **media**, of the **creation** process and of the **usage**. The media information describes storage format, coding parameters, media quality, media location, and so on. The creation information gathers author-generated information about the generation or production process of the content. This information is related to the content but it is generally not explicitly depicted by the content itself. It describes features such as the title, the author of the script, the director, the character, the target audience, the genre, reviews, rating and so forth. The usage information describes rights, financial aspects and availability of the content.

In the simplest case, the content will be created or recorded only once. In this situation, the content can be represented as a unique media instance with its associated format, coding scheme, creation information and usage information. A typical example of this simplest case is a picture from a consumer digital photograph album (see Figure 8.1). Nevertheless, more complex scenarios have to be considered, for example, where a single event, called a *reality* (e.g. a sports event), is recorded in different modalities or with different encoding formats. In this case, different content entities will be produced. The media identification tools describe each content entity by providing a unique identifier for the content entity and the audiovisual domain information. For example, one may record the reality of Ana and Alex shaking hands with a digital photo camera and a digital video camera (see Figure 8.2, where these two content entities are illustrated). In this case, the two content entities correspond to the same reality. The reality captured by one or more content entities can then be described as one or more narrative worlds, as defined in Section 8.4, using the semantic tools presented in that section.

Each content entity can also be encoded with various coding schemes or parameters and stored in various formats. The combination of a coding scheme with its associated parameters and a format corresponds to a **media profile**. The concept of a media profile refers to different variations that can be produced from a master media instance by selecting various coding schemes, parameters or storage formats. Within the same media profile, several copies or instances of the same content can be created. The original media profile is called the **master** profile. If the master is digital, any copy of this instance is indistinguishable from the original and is considered as an instance of the master media profile. In the case of an analog master, copies can be considered instances of the same master media profile if the replication process causes an insignificant loss of quality. Otherwise, each copy generates a new media profile. Finally, the process of generating new instances from existing media instances by changing media formats, encoding schemes and/or parameters creates new media profiles.

The content management description shown in Figure 8.2 illustrates an image and a video content entities. The image content entity has two profiles; the master profile is of high quality and has only one instance. For applications where this quality is not necessary (e.g. for publication on a Web page), the second profile has a reduced spatial resolution of the image and a coarser quantization. This example includes two instances of the second profile. The video content entity in Figure 8.2 has only one profile with one instance but captures the same reality as the image content entity, that is, 'Ana and Alex are shaking hands in front of Columbia University, on September 19th, 1998'. These two examples also illustrate the relationships between the media, creation and usage descriptions.

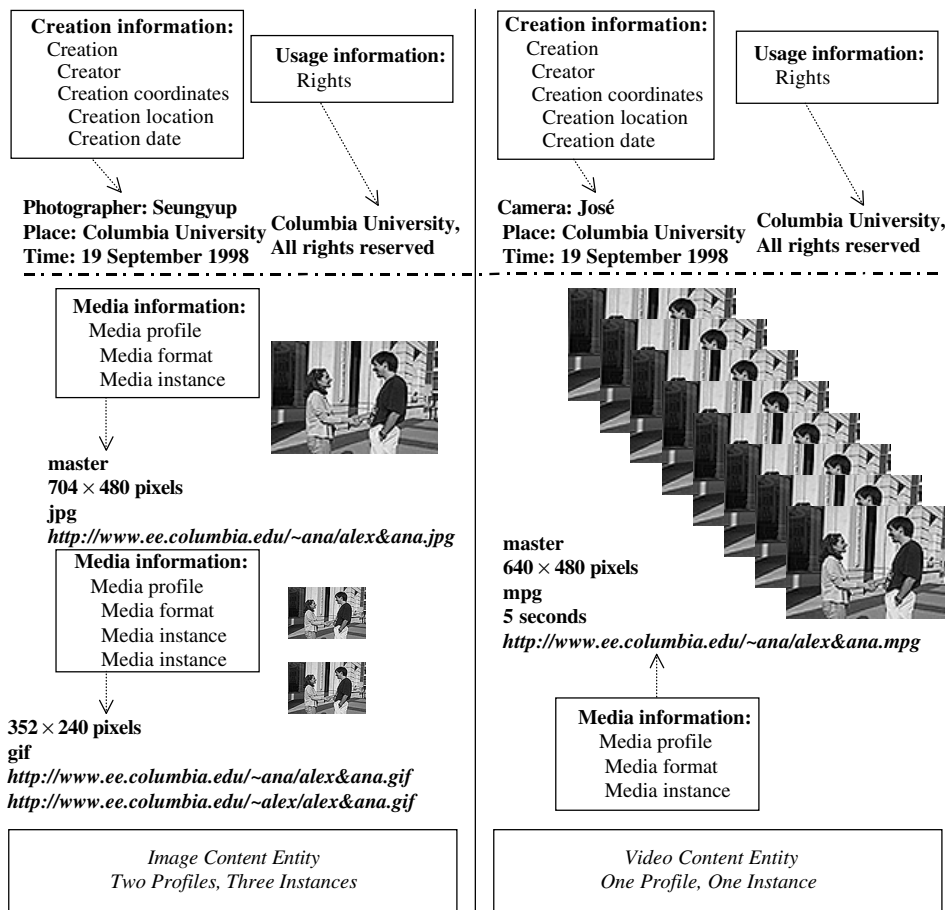


Figure 8.2 Examples of media, creation and usage descriptions of two different content entities capturing the same the reality – ‘Ana and Alex are shaking hands in front of Columbia University, on September 19th, 1998’

Beside the description of media profiles and instances, content management tools can also provide transcoding hints that allow the creation of additional media variations for applications that need to adapt the content for transmission or delivery under specific network conditions or to terminals with specific characteristics. Note that the transcoding hints provide information on how to generate new content from the current piece of content. The variation tools [3] have a different purpose as they allow the description of relationships between preexisting variations of the content.

8.2.1 Media Information

The description of the media features of a content entity is wrapped in the MediaInformation Description Scheme. The MediaInformation description scheme is composed

of an optional **MediaIdentification** Descriptor and one or more **MediaProfile** description schemes.

The **MediaIdentification** Descriptor identifies the content entity independently of the available media profiles and associated media instances. It includes a unique identifier and optional information about the AV domain (i.e. the type of the content in terms of generation such as synthetic or natural creation, in terms of application).

The **MediaProfile** description scheme allows the description of one media profile of the content entity. The **MediaProfile** description scheme is composed of a **MediaFormat** Descriptor, a **MediaInstance** description scheme, a **MediaTranscodingHints** Descriptor and a **MediaQuality** Descriptor.

- The **MediaFormat** Descriptor describes the coding parameters and the format of a media profile. For example, content type (image, video, audio etc.), **file format**, file size, coding scheme, parameters describing frames, pixels, or audio samples are defined, among others.
- The **MediaInstance** description scheme identifies and locates the different instances available in the same media profile. It contains a unique instance identifier and the location of the instance. It relies on a **MediaLocator** or a **LocationDescription** element.
- The **MediaTranscodingHints** Descriptor specifies transcoding hints of the media profile being described. The purpose of this description tool is to improve the quality and reduce the complexity for **transcoding** applications.
- The **MediaQuality** Descriptor contains rating information about the signal quality of a media profile. The quality of audio or visual content may decrease when the signal goes through compression, transmission or signal conversion. It can describe both subjective and objective quality ratings.

Note that the use of these tools is often optional. In addition, the **MediaProfile** description scheme is recursive, in other words, it can contain other **MediaProfile** description schemes for describing the multiplexing of elementary multimedia streams to form composed content entities. Examples include MPEG-4 streams that include multiple media objects, and DVDs where multiple audio streams are available for different languages.

8.2.2 **Content Creation**

The creation information tools describe author-generated information about the generation or production process of a content entity. This information cannot usually be extracted from the content entity itself. That is, the information is related to, but not explicitly depicted by, the actual content entity.

The description of the creation information is wrapped in the **CreationInformation** description scheme, which is composed of one **Creation** description scheme, an optional **Classification** description scheme and zero or more **RelatedMaterial** description schemes.

- The **Creation description scheme** describes the creation of the content entity, including places, dates, actions, materials, staff (technical and artistic, for example, directors and

actors) and organizations involved. Titles and abstracts can also be described using the Creation description scheme.

- The *Classification description scheme* classifies the content for searching and filtering. It may be used in conjunction with the user-preference tools [4]. It includes user-oriented classifications such as language, subject, genre, production type (e.g. documentary and news), as well as service-oriented classifications such as purpose, parental guidance, market segment and media review.
- The *RelatedMaterial description scheme* specifies additional information about the content entity available in other materials.

8.2.3 Content Usage

The description of the usage information is wrapped in the UsageInformation description scheme, which contains an optional Rights data type, an optional Financial data type and zero or more Availability and UsageRecord description schemes.

- The *Rights data type* gives access to information about the right holders and the access rights. No rights information is explicitly described by MPEG-7. The MPEG-7 standard simply provides links to the right holders and other information related to rights management and protection. The Rights data type provides these references in the form of unique identifiers that are under management by external authorities. The underlying strategy is to enable MPEG-7 descriptions to provide access to current rights owner information without dealing with the information and negotiation directly.
- The *Financial data type* contains information related to the costs generated and incomes produced by the multimedia content. The notions of partial costs and incomes allow the classification of different costs and incomes as a function of their type. Total and subtotal costs and incomes can be calculated by the application from these partial values.
- The *Availability description scheme* describes the access availability of the content entity's media instances. It contains tools for referencing the associated media instance and describing, among others, the type of publication medium, the disseminator, additional financial information such as publication costs, price of use, and the availability period and type (e.g. live, repeat, first showing, conditional access, pay per view).
- The *UsageRecord description scheme* describes the past use of the content. It contains a reference to the associated instances of the Availability description scheme.

Note that the UsageInformation description scheme may be updated or extended each time the content is used (e.g. instances of the UsageRecord description scheme and the Income element in the Financial data type) or when there are new ways to access to the content (e.g. instances of the Availability description scheme).

8.3 CONTENT STRUCTURE

The content structure tools describe the spatial, temporal and media source structure of multimedia content, as a set of interrelated segments. They describe the segments, their attributes and the hierarchical decompositions and structural relationships among

segments. In this section, we describe in more detail the MPEG-7 segment entity, segment attribute, segment decomposition and structural relation tools.

8.3.1 Segment Entities

The MPEG-7 segment entity tools describe segments of multimedia content in space, time and/or media source. The segment entity tools are derived from the Segment description scheme, which is an abstract type that represents an arbitrary fragment or section of multimedia content. The description schemes derived from the Segment description scheme describe generic or specific types of segments.

The generic segment entity tools describe spatiotemporal segments of generic types of multimedia content such as images, videos, audio sequences and audiovisual sequences. The most basic visual segment entity tool is the `StillRegion` description scheme, which describes a spatial region of a 2-D image or a video frame (a still region is a group of pixels in the digital case). Figure 8.1 shows three examples of still regions in an image corresponding to the full image (whose identity is 'SR1' in Figure 8.1) and the two people depicted by the image ('SR2' and 'SR3'). Other purely visual segment entity tools are the `VideoSegment` description scheme and the `MovingRegion` description scheme, which describe, respectively, a temporal interval and a spatiotemporal region of a video sequence (a group of frames in a video and a group of pixels in a group of video frames, respectively, in the digital case). The `AudioSegment` description scheme represents a temporal interval of an audio sequence (a group of samples in the digital case). Both the `AudioVisualSegment` description scheme and the `AudioVisualRegion` description scheme describe audio and visual data in an audiovisual sequence. In particular, the `AudioVisualSegment` description scheme describes a temporal interval of audiovisual data, which corresponds to both the audio and video in the same temporal interval, whereas the `AudioVisualRegion` description scheme describes an arbitrary spatiotemporal segment of audiovisual data, which corresponds to the audio in an arbitrary temporal interval and the video in an arbitrary spatiotemporal region. Figure 8.3 shows examples of a video segment, a moving region, an audiovisual segment and several audiovisual regions.

There is also a set of MPEG-7-specific segment entity tools that describe segments of specific multimedia content types such as `mosaic`, 3-D, image- or `videotext`, ink content, multimedia content and video editing. For example, the `Mosaic` description scheme extends from the `StillRegion` description scheme and describes a mosaic or panoramic view of a video segment. A mosaic is usually constructed by aligning and blending together the frames of a video segment upon each other using a common spatial reference system. The `StillRegion3D` description scheme represents a 3-D spatial region of a 3-D image. The `ImageText` description scheme and the `VideoText` description scheme extend, respectively, from the `StillRegion` description scheme and the `MovingRegion` description scheme. They describe a still region and a moving region that correspond to the text, respectively. The `InkSegment` description scheme represents a spatiotemporal segment of ink content created by a pen-based system or an electronic white board. The `MultimediaSegment` description scheme represents a composite of segments forming a multimedia presentation such as an `MPEG-4` presentation or a Web page. Finally, the analytic edited video segment description schemes such as the `AnalyticClip` description scheme and the

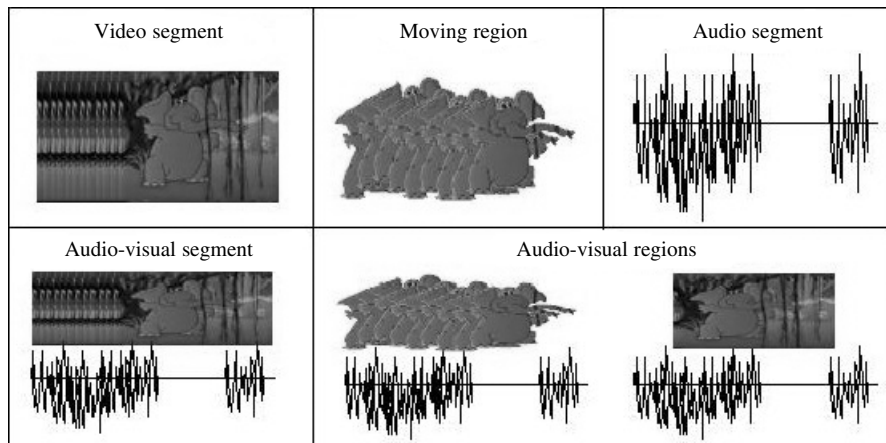


Figure 8.3 Examples of a video segment, a moving region, an audio segment, an audiovisual segment, an audiovisual region composed of a moving region and an audio segment and an audiovisual region composed of a still region

AnalyticTransition description scheme extend originally from the VideoSegment description scheme and describe, respectively, different types of shots and transitions between the shots resulting from video **editing work**. The video editing description is ‘analytic’ in the sense that it is made a posteriori on the final edited video content. Figure 8.4 shows examples of a mosaic, a 3-D still region, an image text, an ink segment, a multimedia segment, two analytic clips and an analytic transition.

The segment entity **tools** can describe segments that are not connected but composed of several separated **connected components**. **Connectivity** refers here to both spatial and temporal dimensions. A temporal segment (instance of the VideoSegment, AudioSegment, AudioVisualSegment, InkSegment description schemes and the audio part of the

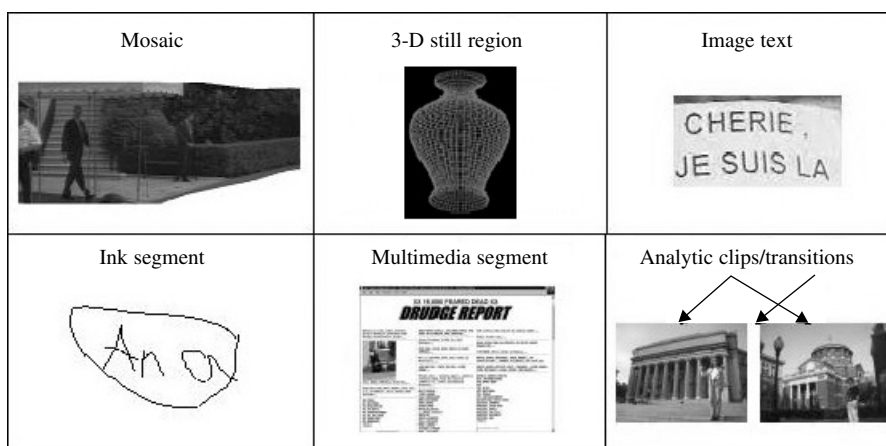


Figure 8.4 Examples of a mosaic, a 3-D still region, an image text, an ink segment, a multimedia segment, two analytic clips and an analytic transition

AudioVisualRegion description scheme) is said to be temporally connected if it is continuous over a temporal interval (e.g. a sequence of continuous video frames and/or audio samples). A spatial segment (instance of the StillRegion description scheme) is said to be spatially connected if it is a continuous spatial region (e.g. a group of connected pixels). A spatiotemporal segment (instance of the MovingRegion description scheme and the video part of the AudioVisualRegion description scheme) is said to be spatially and temporally connected if it appears in a temporally connected segment and if it is formed of a spatially connected segment at each time instant. Note that this is not the classical definition of connectivity in a 3-D space.

Figure 8.5 illustrates several examples of temporal and spatial segments that are connected or composed of several connected components. Figure 8.6 shows examples of a connected and a nonconnected moving region. In Figure 8.6(b), the moving region is not connected because it is not instantiated in all the frames and, furthermore, it involves several spatial connected components in some of the frames.

The Segment description scheme is abstract and therefore cannot be instantiated on its own. However, the Segment description scheme contains elements and attributes that are common to the different segment types. Among the common properties of segments, there is information related to the creation, usage, media, text annotation and importance for matching and point of view. Derived segment entity tools can describe other visual, audio and spatiotemporal properties of segments. Note that in all cases, the description tools attached to the segment entity tools are global to the union of the connected components composing the segment being described. At this level, it is not possible to describe the individual connected components of the segment. If connected components have to be described individually then the segment has to be decomposed into various **subsegments**

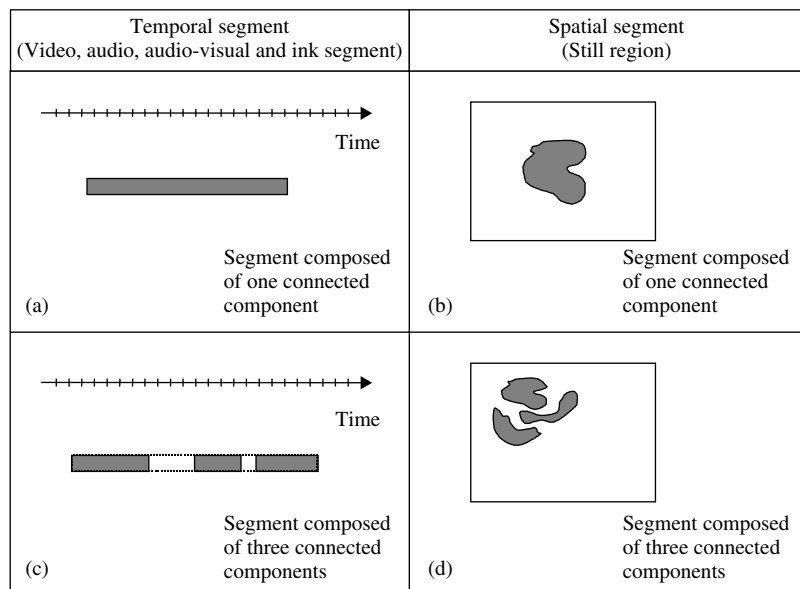


Figure 8.5 Examples of segments: (a) and (b) segments composed of one single connected component; (c) and (d) segments composed of three separated connected components

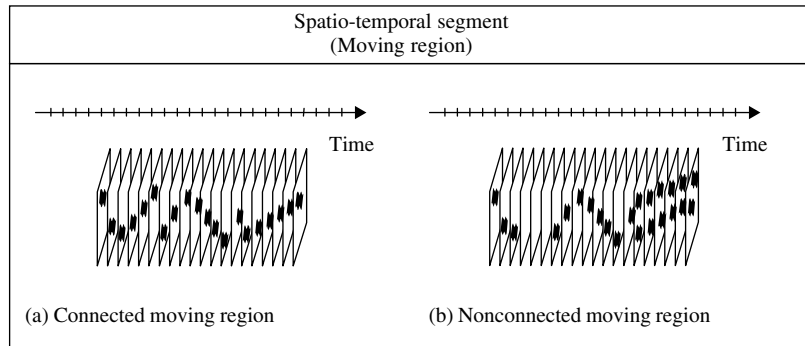


Figure 8.6 Examples of connected: (a) nonconnected and (b) moving region

corresponding to its individual connected components using the [segment decomposition](#) tools (see Section 8.3.3).

8.3.2 Segment Attributes

As mentioned before, any kind of segment can be described in terms of its media information, creation information, usage information, text annotations, visual features, audio features and other segment attributes. Specific features such as the characterization of the connected components of the segment, the segment importance, the importance of some of its descriptors can also be described. As an example, the properties of the still regions in the image description in Figure 8.1 are described using text annotation, color structure and matching hint tools.

The Mask Descriptor is an abstract type that represents the connected components of a segment in time, space and media, among others. The [SpatialMask](#), [TemporalMask](#) and [SpatioTemporalMask](#) Descriptors extend the Mask Descriptor to describe the localization of separated connected components (subregions or subintervals) in space and/or in time. They are used to describe the connectivity of nonconnected segments (see Section 8.3.1). For example, the [SpatialMask](#) Descriptor describes the localization of the spatial connected components of a still region. Similarly, the [TemporalMask](#) Descriptor describes the localization of the temporal connected components of a video segment, an ink segment, an audio segment, an audiovisual segment, or the audio part of an audiovisual region. Finally, the [SpatioTemporalMask](#) Descriptor describes the localization of the spatiotemporal components of a moving region or the visual part of an audiovisual region. In the InkSegment description scheme, the [SceneGraph](#), [MediaSpaceMask](#) and [OrderedGroupDataSetMask](#) Descriptors, which also extend the Mask Descriptor, describe the localization of separated subgraphs in a scene graph, data subintervals in a file and subsets in an ordered group data set for ink segments.

The importance of segments and segment descriptors is described by the [MatchingHint](#) and [PointOfView](#) Descriptors, respectively. The [MatchingHint](#) Descriptor describes the relative importance of instances of audio or visual descriptors or description schemes (e.g. the color structure of a still region) or parts of audio or visual descriptors or description schemes (e.g. the fourth Value element of a color structure description of a still

region) in instances of segment entity tools. For specific applications, the `MatchingHint` Descriptor can improve the retrieval performance because the most relevant descriptors for matching may depend on the application but also may vary from segment to segment. On the other hand, the `PointOfView` Descriptor describes the relative importance of segments given a specific viewpoint. The `PointOfView` Descriptor assigns values between zero and one to segments on the basis of a viewpoint specified by a string (e.g. 'Home team' for a soccer game).

Other segment attribute tools describe specific media, creation and handwriting recognition information related to ink segments. These tools are the `InkMediaInfo`, `HandWritingRecogInformation` and `HandWritingRecogResult` description schemes. The `InkMediaInformation` description scheme describes parameters of the input device of the ink content (e.g. width, height and lines of the writing field's bounding box), writer's handedness (e.g. 'right' or 'left') and ink data style (e.g. 'cursive' or 'drawing'). The `HandWritingRecogInformation` description scheme describes the handwriting recognizer and the ink lexicon used by the recognizer. The `HandWritingRecogResult` description scheme describes handwriting recognition results, in particular, the overall quality and some results of the recognition process with corresponding accuracy scores.

Finally, depending on the nature of the segment, visual and audio descriptors or description schemes can be used to describe specific features related to the segment. Examples of visual features are color, shape, texture and motion, whereas examples of audio features involve audio spectrum, audio power, fundamental frequency, harmonicity, timbre, melody and spoken content, among others.

8.3.3 Segment Decompositions

The segment decomposition tools describe the decomposition or subdivision of segments into subsegments. They support the creation of segment hierarchies to generate, for example, `tables of contents`. The decomposition is described by the `SegmentDecomposition` description scheme, which is an abstract type that represents an arbitrary decomposition of a segment. Segment decomposition tools derived from the `SegmentDecomposition` description scheme describe basic types of segment decompositions and the decompositions of specific types of segments. It is important to note that MPEG-7 does not specify how a segment should be divided into subsegments or describe the segmentation process; however, it can describe the criteria used during the segmentation and the dimensions affected by the segmentation: space, time and/or media source.

The basic segment decomposition tools are abstract types that describe spatial, temporal, spatiotemporal and media source decompositions of segments. These tools are the `SpatialSegmentDecomposition`, `TemporalSegmentDecomposition`, `SpatioTemporalSegmentDecomposition` and `MediaSourceDecomposition` description schemes, accordingly. For example, an image can be decomposed spatially into a set of still regions corresponding to the objects in the image, which, at the same time, can be decomposed into other still regions. Figure 8.1 exemplifies the spatial decomposition of a still region into two still regions. Similar decompositions can be generated in time and/or space for video and other multimedia content. Media source decompositions divide segments into its media constituents such as audio tracks or viewpoints from several cameras.

The subsegments resulting from decomposition may overlap in time, space and/or media source; furthermore, their union may not cover the full time, space and media extents of the parent segment, thus leaving gaps. Two attributes in the SegmentDecomposition description scheme indicate whether a decomposition leaves gaps or overlaps. Note that, in any case, the segment decomposition implies that the union of the spatiotemporal and media spaces defined by the child segments is included in the spatiotemporal and media spaces defined by their parent segment (i.e. children are contained in their parents).

Several examples of decompositions for temporal segments are included in Figure 8.7. Figure 8.7(a) and (b) show two examples of segment decompositions with neither gaps nor overlaps (i.e. a partition in the mathematical sense). In both cases, the union of the temporal extents of the children corresponds exactly to the temporal extent of the parent, even if the parent is itself nonconnected (see Figure 8.7b). Figure 8.7(c) shows an example of decomposition with gaps but no overlaps. Finally, Figure 8.7(d) illustrates a more complex case in which the parent is composed of two connected components and its decomposition generates three children with gaps and overlaps: the first child is itself nonconnected and composed of two connected components, the two remaining children are composed of a single connected component.

The decomposition of a segment may result in segments of different nature. For example, a video segment may be decomposed into other video segments as well as into still or moving regions. However, not all the combinations of parent segment type, segment decomposition type and child segment type are valid. For example, an audio segment can only be decomposed in time or media into other audio segments; the spatial decomposition of an audio segment into still regions is not valid. The valid decompositions

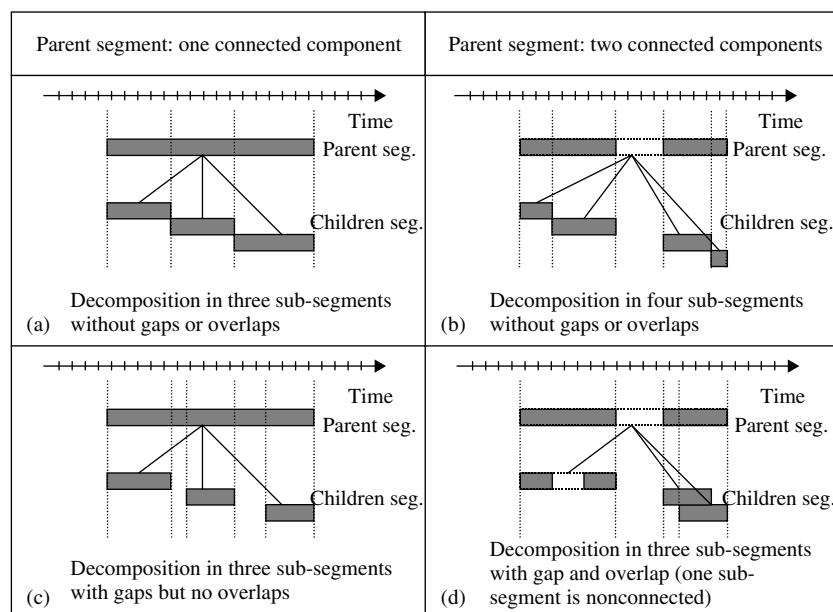


Figure 8.7 Examples of segment decompositions: (a) and (b) decompositions with neither gaps nor overlaps; (c) and (d) decompositions with gaps and/or overlaps

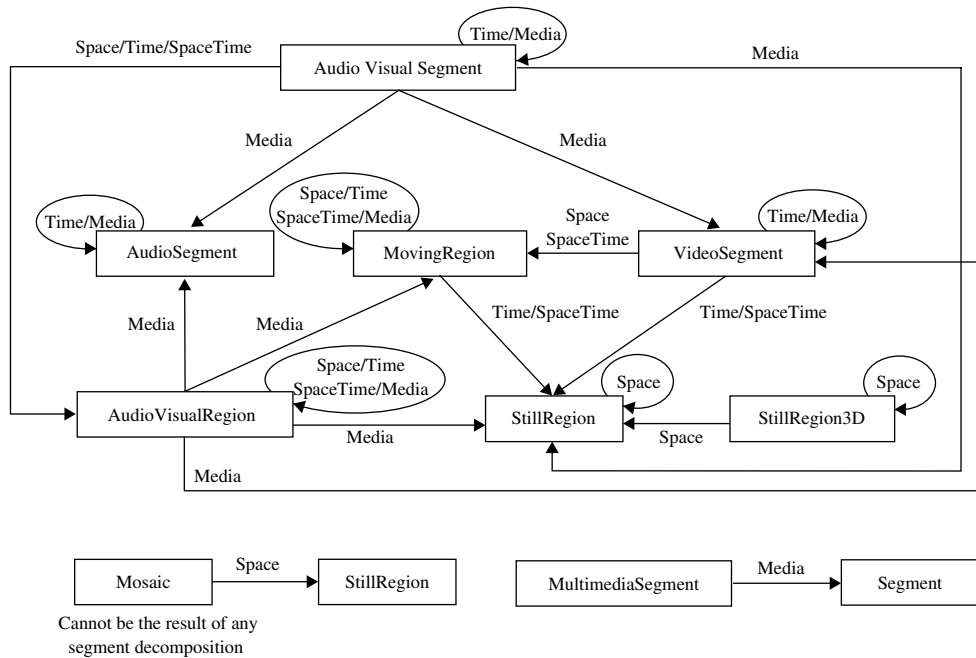


Figure 8.8 Valid segment decompositions among some specific types of segments. ‘time’, ‘space’, ‘spaceTime’ and ‘media’ stand for temporal, spatial, spatiotemporal and media source segment decomposition, respectively. Each description scheme represents itself and its derived description schemes

among some specific types of segments are shown in Figure 8.8. Each description scheme in the figure represents itself as well as its derived description schemes. For example, as shown in the figure, a still region can only be decomposed in space into other still regions including image texts. The Mosaic description scheme is a special case of the StillRegion description scheme, which cannot be the result of any segment decomposition.

An example of image description with several spatial decompositions is illustrated in Figure 8.9. The full image (whose identity is ‘SR1’) is described using the StillRegion description scheme whose creation (title, creator), usage (copyright), media (file format), text annotation (textual description) and color properties are described using the Creation-Information, UsageInformation, MediaInformation, TextAnnotation and ColorStructure description tools. This first still region is decomposed into two still regions, which are further decomposed into other still regions. For each decomposition step, Figure 8.9 indicates whether gaps and overlaps are generated. The complete segment hierarchy is composed of nine still regions (note that ‘SR9’ and ‘SR7’ are single segments composed of two separated connected components). For each region, Figure 8.9 shows the type of feature that is described.

Describing the hierarchical decomposition of segments is useful to design efficient search strategies (global search to local search). It also allows the description to be scalable: a segment may be described by its direct set of description tools, but it may also be described by the union of the description tools that are related to its subsegments.

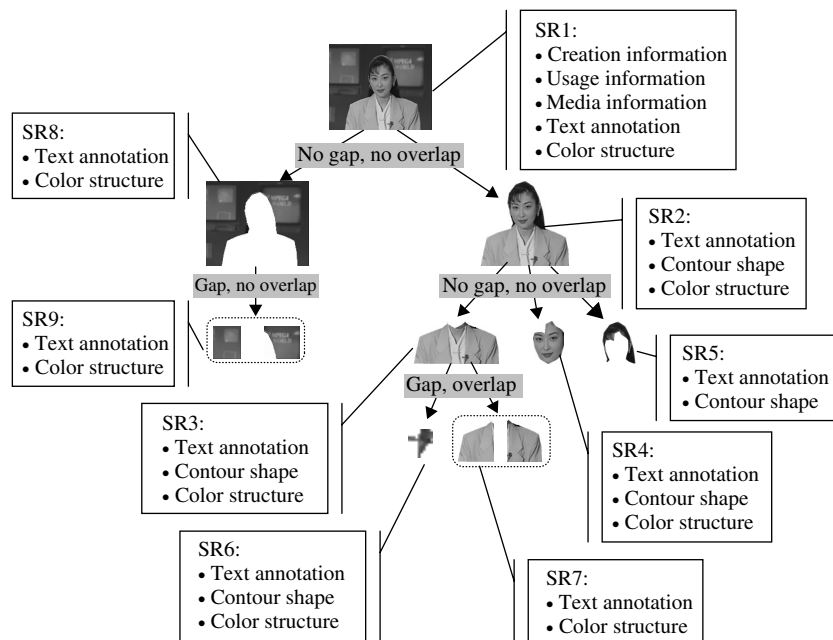


Figure 8.9 Example of an image description as a hierarchy of still regions and associated features

8.3.4 Structural Relations

The description of the content structure in MPEG-7 is not constrained to rely on hierarchies. Although, hierarchical structures provided by the segment decomposition tools are adequate for efficient access, retrieval and scalable description, they imply constraints that may make them inappropriate for certain applications. In such cases, the MPEG-7 graph, base and structural relation tools can be used to describe more general segment structures. MPEG-7 has standardized common structural relations such as *left* and *precedes* but it allows the description of non-normative relations (see soccer example below). Figure 8.1 shows an example of the spatial relationship *left* between the still regions ‘SR2’ and ‘SR3’.

The structural relation tools include the **SpatialRelation** Classification Scheme (CS) and the TemporalRelation CS, which specify spatial and temporal relations, respectively. The normative structural relations in MPEG-7 are listed in Table 8.1. For each normative structural relation, MPEG-7 has also standardized the inverse relation. Note that the

Table 8.1 Normative structural relations in MPEG-7 listed by type

Type	Normative relations
Spatial	South, north, west, east, northwest, northeast, southwest, southeast, left, right, below, above, over, under
Temporal	Precedes, follows, meets, metBy, overlaps, overlappedBy, contains, during, strictContains, strictDuring, starts, startedBy, finishes, finishedBy, coOccurs, contiguous, sequential, coBegin, coEnd, parallel, overlapping

inverse relation is defined as follows: ‘A InverseRelation B’ \Leftrightarrow ‘B Relation A’, where A and B are the source and the target of *InverseRelation*, respectively.

The *SpatialRelation* CS specifies spatial relations that apply to entities that have 2D spatial attributes, which may be implicitly or explicitly stated. Normative spatial relations describe how entities are placed and relate to each other in 2-D space (e.g. *left* and *above*). The normative spatial relations may apply to entities such as still regions (see Section 8.3.1) and objects (see Section 3.2), among others.

In a similar way, the *TemporalRelation* CS specifies temporal relations that apply to entities that have implicitly or explicitly stated temporal attributes. Normative temporal relations include binary relations between two entities corresponding to Allen’s temporal interval relations [5] (e.g. *precedes* and *meets*) and n-ary relations among two or more entities such as *sequential* and *parallel*, among others. The normative temporal relations may apply entities such as video segments (see Section 8.3.1), moving regions (see Section 8.3.1) and events (see Section 8.4.2), among others.

Finally, the generic relations specified by the *GraphRelation* CS (e.g. *identity* [1]) and the *BaseRelation* CS (e.g. *union* and *disjoint* [1]) can also be used to describe relations among segments.

To illustrate the use of the segment entity and structural relation tools, consider the example in Figure 8.10 and Figure 8.11. This example shows an excerpt from a soccer match and its description. Two video segments and four moving regions are described. A graph of structural relationships describing the structure of the content is shown in Figure 8.11. The video segment ‘Dribble&Kick’ involves the moving regions ‘Ball’, ‘Goalkeeper’ and ‘Player’. The moving region ‘Ball’ remains *close to* the moving

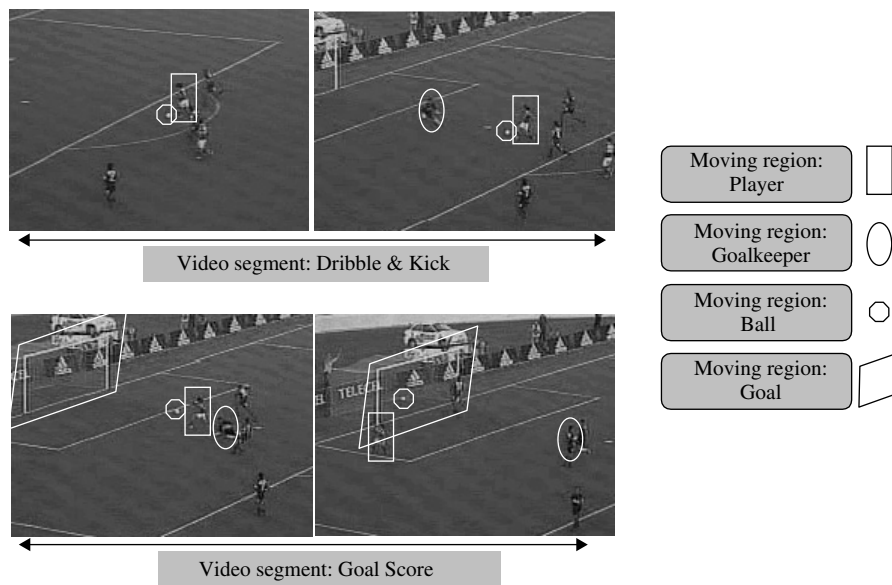


Figure 8.10 Excerpt from a soccer match with the corresponding video segments and moving regions that participate in the graph in Figure 8.11

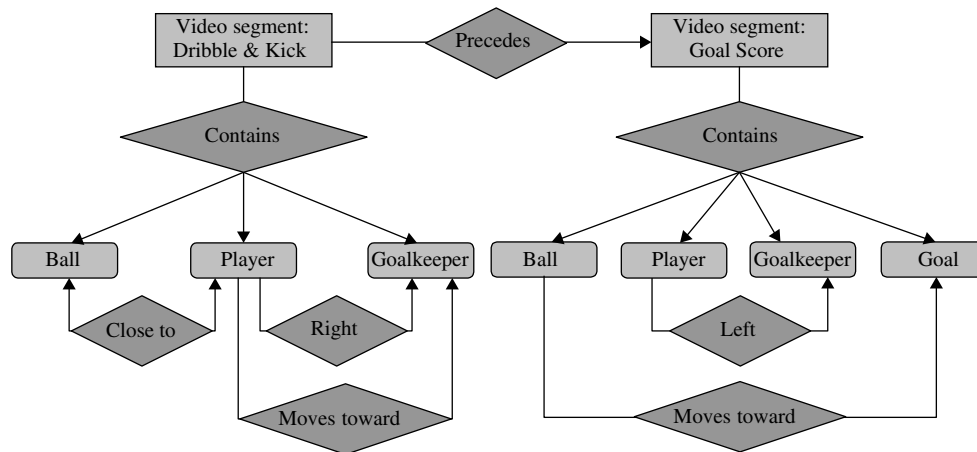


Figure 8.11 Example of a graph of structural relationship for the video segments and moving regions in Figure 8.10

region ‘Player’, which is *moving toward* the moving region ‘Goalkeeper’. The moving region ‘Player’ is on the *right* of the moving region ‘Goalkeeper’. The video segment ‘GoalScore’ involves three moving regions corresponding to the same three objects as the three previous moving regions including the moving region ‘Goal’. In this video segment, the moving region ‘Player’ is on the *left* of the moving region ‘Goalkeeper’ and the moving region ‘Ball’ *moves toward* the moving region ‘Goal’. This example illustrates the flexibility of this kind of representation. Note that this description is mainly structural because the relationships specified in the graph edges are purely spatiotemporal and visual and the nodes represent segments. The only explicit semantic information in the description is available from the text annotation (where key words such as ‘Ball’, ‘Player’, or ‘Goalkeeper’ are used). All the relations in this example are normative except for *close to* and *moves toward*.

8.4 CONTENT SEMANTICS

The MPEG-7 tools for describing the semantics of multimedia content can be best understood by analyzing how semantic descriptions of anything are constructed in general. One way to describe semantics is to start with **events**, understood as occasions when something happens. **Objects**, **people** and **places** can populate such occasions and the times at which they occur. Some descriptions may concentrate on the latter aspects, with or without events, and some may even begin with a place, or an object, and describe numerous related events. Furthermore, these entities can have properties and states through which they pass as what is being described transpires. There are the interrelations among these entities. Finally, there is the world in which all of this is going on, the background, the other events and other entities, which provide context for the description.

If a description is being given between people, much of the information exchange is done with reference to common experiences, past events and analogies to other situations. A semantic description may be more or less complete, and may rely on a little or a lot

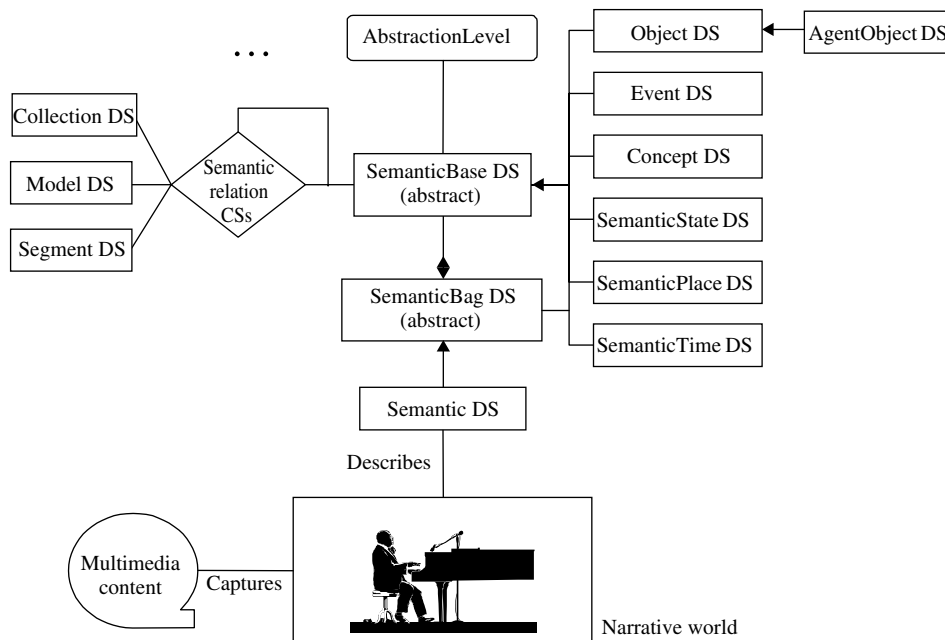


Figure 8.12 MPEG-7 tools that describe the semantics of multimedia content. The figure makes use of Unified Modeling Language (UML) and Extended Entity Relation (EER) graphical formalisms. Rectangles, rounded-edge rectangles and diamonds represent entities, attributes and relations, respectively; the edges that start with shaded triangles and shaded diamonds represent inheritance and composition, respectively; lines with names represent associations

of generic or contextual information and on other descriptions. Since the ultimate goal is to describe the semantics of multimedia content and since much of these semantics are narrative in quality (as in a movie that tells a story or an Expressionist symphony), MPEG-7 refers to the participants, background, context and all the other information that makes up a single narrative as a **narrative world**. One piece of content may have multiple narrative worlds or vice versa, but the narrative worlds exist as separate semantic descriptions.

The components of the above semantic descriptions roughly fall into entities that populate narrative worlds, their attributes and the relationships among them. In this section, we describe in more detail the tools provided by MPEG-7 to describe such semantic entities, attributes and relationships, which are shown in Figure 8.12. We start by introducing the abstraction model of MPEG-7 semantic descriptions and finish by discussing application issues in using the MPEG-7 content semantic tools for implicitly or explicitly describing the semantics of multimedia content.

8.4.1 **Abstraction** Model

Besides the concrete semantic description of specific instances of multimedia content, MPEG-7 content semantic tools also allow the description of abstractions and abstract quantities.

Abstraction refers to taking a semantic description of a specific instance of multimedia content (e.g. ‘Alex is shaking hands with Ana’ in the image shown in Figure 8.1) and generalizing it either to multiple instances of multimedia content (media abstraction, e.g. ‘Alex is shaking hands with Ana’ for any picture or video including the images and videos shown in Figure 8.2) or to a set of concrete semantic descriptions (formal abstraction, e.g. ‘Alex is shaking hands with any woman’ or ‘Any man is shaking hands with any woman’). When dealing with the semantics of multimedia content, it becomes necessary to draw upon descriptive techniques such as abstraction because these techniques are part of the way the world is understood and described by humans. MPEG-7 considers two types of abstraction, which are called *media abstraction* and *formal abstraction*. Moreover, MPEG-7 also provides ways of representing abstract properties and concepts that do not arise by a media or a formal abstraction. As examples, Figure 8.1 shows a concrete description of an image that contains an abstract property or concept ‘comradeship’ and Figure 8.13 illustrates a media abstraction and a formal abstraction of the description in Figure 8.1.

An abstraction, or more precisely, a *lambda abstraction*, in logic, creates a function from a statement [6]. This process is effected by replacing one or more of the constant expressions in the statement by a variable. There are two types of models for such abstraction, typed and untyped. The more useful method is typed abstraction. For this abstraction, the variable is given a type, such as a data type, and the constants that may be substituted for it must be of this type. In this process, the statement is changed into

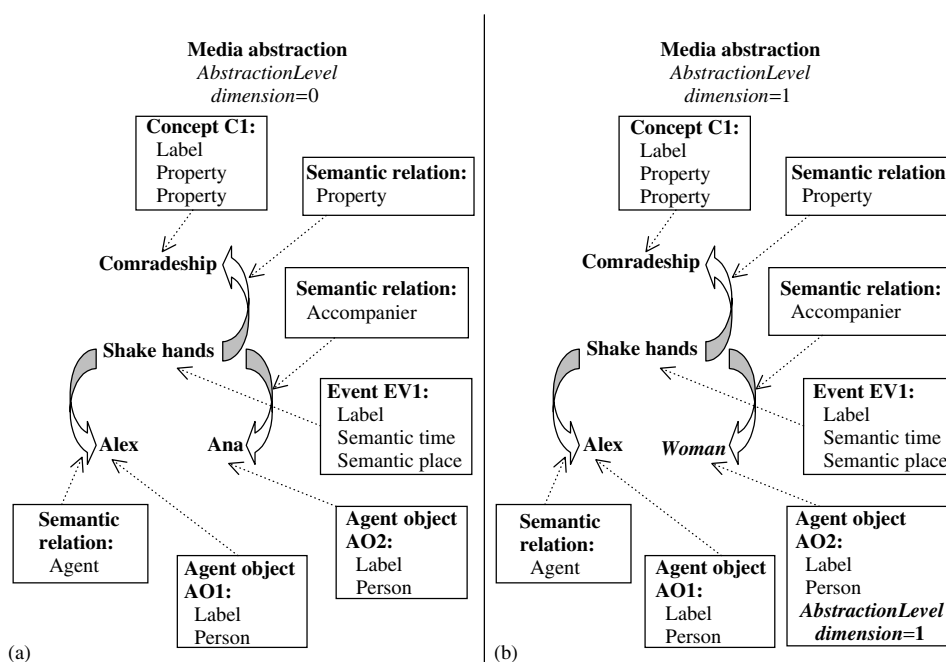


Figure 8.13 Examples of a media abstraction: (a) formal abstraction and (b) for the semantic description in Figure 8.1. Section 8.4.3 explains how to use the AbstractionLevel data type to indicate media and formal abstractions

a set of statements, one for each choice of constants. Abstraction thus replaces a single instance with a more general class. In describing media, two types of abstractions can occur: media abstractions and formal abstractions. The method for indicating these two types of abstractions is to use the dimension attribute in the `AbstractionLevel` data type (see Section 8.4.3).

A media abstraction is a description that has been separated from a specific instance of multimedia content and can describe all instances of multimedia content that are sufficiently similar (similarity depends on the application and on the detail of the description). If we consider the description ‘Alex is shaking hands with Ana’ of the image shown in Figure 8.1, a media abstraction is the description ‘Alex is shaking hands with Ana’ with no links to the image and that is, therefore, applicable to any image or video depicting that event (see Figure 8.13a). Another example is a news event broadcast on different channels to which the same semantic description applies. The variable in a media abstraction is the media itself.

A formal abstraction describes a pattern that is common to a set of examples. The common pattern contains placeholders or variables to be filled in that are common to the set. A formal abstraction may be formed by gathering a set of examples, determining the necessary placeholders from these examples and thus deriving the pattern, or by first creating the common pattern, and then creating the specific examples from the placeholder. The description is a formal abstraction as long it contains variables, which, when filled in, would create specific examples or instances, either media abstractions or concrete descriptions. Formal abstractions of the semantic description in Figure 8.1 could be ‘Alex is shaking hands with any woman’ and ‘Any man is shaking hands with any woman’, where ‘any woman’ and ‘any man’ are the variables of the formal abstractions (see Figure 8.13b). The variables in a formal abstraction can be replaced or filled in with concrete instances using the semantic relation *exemplifies* (see Section 8.4.4).

MPEG-7 also provides methods of describing abstract quantities such as properties (see the `Property` element in Section 8.4.3) and concepts (see the `Concept` description scheme Section 8.4.2). For instance, the property ‘ripeness’ can be an attribute of the object ‘banana’. Concepts are collections of properties that define a category of entities but cannot fully characterize it, that is, abstract quantities that are not the result of an abstraction ‘comradeship’ (see semantic description in Figure 8.1). An abstract quantity may be describable only through its properties, and therefore is not abstract because it comes from the replacement of elements of a description by generalizations or variables. Note that the same abstract quantity could appear as a property or as a concept in a description. In order to describe relationships of one or more properties, or allow multiple semantic entities to be related to one or more properties, or specify the strength of one or more properties, the property or collection of properties must be described as a concept using the `Concept` description scheme.

8.4.2 Semantic Entities

The MPEG-7 semantic entity tools describe semantic entities such as narrative worlds, objects, events, concepts, states, places and times. The semantic entity tools are derived from the `SemanticBase` description scheme, which is an abstract type that represents

any semantic entity. The description schemes derived from the SemanticBase description scheme describe specific types of semantic entities (see Figure 8.12) such as narrative worlds, objects and events.

A narrative world in MPEG-7 is represented using the Semantic description scheme, which is described by a number of semantic entities and the graphs of their relationships. The Semantic description scheme is derived from the SemanticBag description scheme, which is an abstract type representing any kind of collection of semantic entities and their relationships. The Semantic description scheme represents one type of those collections, especially describing a single narrative world (e.g. world 'Earth' in a world history documentary or world 'Roman Mythology' in the engravings on a Roman vessel).

Specific types of semantic entities are described by description schemes derived from the SemanticBase description scheme. This abstract type holds common functionalities shared in describing any semantic entity: labels used for categorical searches, a textual description, properties, links to and descriptions of the media and relationships to other entities. In order to facilitate the use of a description of one narrative world as context for another or to allow a narrative world to be embedded in another, the Semantic description scheme (via the SemanticBag description scheme) is also derived from the SemanticBase description scheme. Other description schemes derived from the SemanticBase description scheme are the Object, AgentObject, Event, SemanticPlace, SemanticTime, SemanticState and Concept description schemes. These represent entities that populate the narrative world such as an object, agent and event; the where and when of things; a parametric entity; and an abstract collection of properties, respectively.

The Object and Event description schemes describe perceivable semantic entities, objects and events that can exist or take place in time and space in narrative worlds. The Object description scheme and Event description schemes are recursive to describe the subdivision of objects and events into other objects and events without identifying the type of subdivision. The AgentObject description scheme extends from the Object description scheme to describe an object that acts as a person, a group of persons, or an organization in a narrative world. For example, the semantic description in Figure 8.1 shows one event and two agent objects corresponding, respectively, to 'Shake hands' (whose identity is 'EV1' in Figure 8.1) and to the two people 'Alex' and 'Ana' ('AO1' and 'AO2', respectively) depicted by the image.

The **SemanticPlace** and **SemanticTime** description schemes describe a location and time in a narrative world, respectively. The event 'Shake hands' in Figure 8.1 has associated a location and a time. The SemanticPlace and SemanticTime description schemes can also describe lengths and durations. Notice that these are semantic descriptions of place or time, not necessarily numerical in any sense. They serve to give extra semantic information beyond pinpointing a location or instant. The SemanticPlace and SemanticTime description schemes can describe locations and times that are fictional or that do not correspond to the locations and times where the multimedia content was recorded, which are described using the CreationInformation description scheme. For example, a movie could have been recorded in Vancouver during winter (CreationInformation description scheme) but the action is supposed to take place in New York during fall (Semantic description scheme).

The **SemanticState** description scheme describes the state or parametric attributes of a semantic entity at a given place and/or time in the narrative world. The SemanticState description scheme can describe the changing of an entity's attributes' values in space and time. An example of the state of an object is the piano's weight; an example of the state of a time is the cloudiness of a day, which varies with time during the day. In a description that is being updated (e.g. a running description of a news event), the SemanticState description scheme can be used to give the current state of a semantic entity. It can also be used to parametrize the strength of a relationship between entities (e.g. the ripeness of a banana).

Finally, the **Concept** description scheme forms one part of the semantic description abstraction model. Using the Concept description scheme, concepts are described as collections of one or more properties and given the status of a semantic entity in their own right. As described in Section 8.4.1, the model allows abstraction of the media and abstraction of semantic entities. These are formal abstractions in the sense that they take specific elements of a description and replace them with variables or placeholders. Two other elements of abstraction involve properties of semantic entities. Properties can be modifiers to existing elements of the description. But one can also create collections of properties. In some cases, this leads to an abstraction, that is, there is a category of entities for which this collection is a definition. It is also possible that such a collection of properties does not define a category, but rather is distributed across many categories, not confined to any one. This collection is then an abstract entity that did not occur as the result of some abstraction. We call such an abstraction a concept. An example of a concept could be any affective property of the media, for instance, 'suspense', or a quantity not directly portrayed by the media, such as 'ripeness'. A concept can also represent the embodiment of an adjectival quality. Examples of such use are 'comradeship', 'happiness' and 'freedom'. Concept 'comradeship' (C1) participates in the semantic description shown in Figure 8.1.

Figure 8.14 illustrates most types of semantic entity tools in the description of a piano recital video. The textual description of the example in Figure 8.14 could be as follows: 'Tom Daniels, who is a musician, plays the piano harmoniously at Carnegie Hall from 7 to 8 P.M. on October 14, 1998, in memory of his tutor. The piano weights 100 Kg.'. The Object description scheme is used to describe the object 'piano'. 'Tom Daniels', 'Tom's tutor' and 'musician' are described using the AgentObject description scheme. The Event and Concept description schemes describe the event 'Play' and the concept 'harmony', respectively. Finally, 'Carnegie Hall' and '7 to 8 P.M., October 14, 1998' are described using the SemanticTime and SemanticPlace description schemes, respectively. Figure 8.14 also demonstrates some normative semantic relations that are presented in Section 8.4.4.

Semantic entities involved in descriptions of narrative worlds can change status or functionality, and therefore be described by different semantic entity tools. For instance, a picture can be an object (when it hangs in a museum), or a location ('In the picture'), or the narrative world itself (when describing what picture depicts 'In the picture Caesar is talking to Brutus'). If the same entity is described by multiple semantic entity tools, these descriptions can be related with the semantic relation *identifier* (see Section 8.4.4).

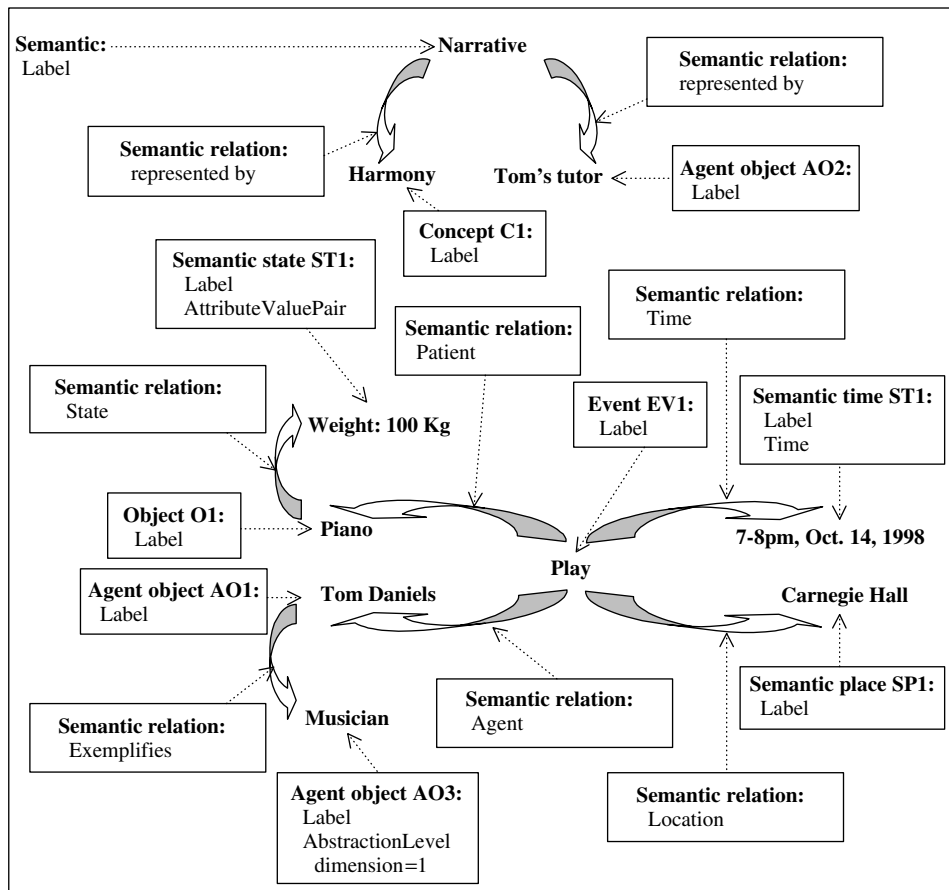


Figure 8.14 Example of a semantic description of a piano recital video

8.4.3 Semantic Attributes

In MPEG-7, semantics entities can be described by labels, by a textual definition, or in terms of properties or of features of the media or segments in which they occur. Other semantic attribute tools describe abstraction levels and semantic measurements in time and space. In addition, the Agent description scheme, the Place description scheme and the Time data type presented in [8] are used to characterize agent objects, semantic places and semantic times, respectively.

The **SemanticBase** description scheme contains one or more Label elements, an optional Definition element and zero or more Property elements, among others. The Label element assigns a label to a semantic entity. A label is similar to a 'descriptor' or 'index term' in Library and Information Science. It is a type used for classifying and retrieving the descriptions of semantic entities (e.g. 'man' for the agent object 'Alex' in Figure 8.1). A semantic entity can have multiple labels, one for each 'index term'. The labels can be used to retrieve all the semantic entities sharing the

same label(s). The Definition element provides a textual definition of a semantic entity. The textual definition of a semantic entity can be done with either free text or structured text (e.g. the textual description of the agent object ‘Alex’ in Figure 8.1 could be ‘bipedal primate mammal (Homo sapiens) that is anatomically related to the great apes but distinguished especially by notable development of the brain’). The Property element describes a quality or adjectival property associated with a semantic entity (e.g. ‘tall’ and ‘slim’ for the agent object ‘Alex’ in Figure 8.1). This is one way of describing abstract quantities in MPEG-7, the other being the Concept description scheme, as described in Section 8.4.2. The MediaOccurrence element in the SemanticBase description scheme describes the appearance or occurrence of a semantic entity in one location in the multimedia content with a media locator, locators in space and/or time, audio descriptors or description schemes and visual descriptors or description schemes. The purpose of the MediaOccurrence element is to provide access to the same information as the Segment description scheme, but without the hierarchy and without extra temporal and spatial information. There are some applications for which this information – location of the media, the temporal and/or spatial localization in the media and the audio and visual description tools at that location – is sufficient. If the description requires more information or access to the media, it should use the semantic relations such as *depicts*, *symbolizes*, or *references* instead of the **MediaOccurrence** element to complement the semantic description. Three types of semantic entity occurrences are considered: a semantic entity can be perceived, symbolized, or referenced (the subject of) in the media.

The **AbstractionLevel** data type in the SemanticBase description scheme describes the kind of abstraction that has been performed in the description of a semantic entity. When it is not present in the description of a semantic entity, the description is concrete – it describes the world depicted by a specific instance of multimedia content and references the multimedia content (through the MediaOccurrence element or the segment entity tools). If the AbstractionLevel element is present in the description, a media abstraction or a formal abstraction is in place. If the description contains a media abstraction, then the dimension attribute of the AbstractionLevel element is set to zero. If a formal abstraction is present, the dimension attribute is set to one or higher. A semantic description that abstracts specific semantic entities gets an AbstractionLevel element of dimension attribute one. Higher values of the dimension attribute of the AbstractionLevel element are used for abstractions of abstractions, such as structural abstractions in which the elements of a graph are not variables but classes of variables. In this case, the graph relations are the only relevant parts of the description. Examples of concrete descriptions, media abstractions and formal abstractions are discussed in Section 8.4.1.

Finally, the Extent and Position data types are used in the SemanticPlace and SemanticTime description schemes to describe semantic measurements in space or time, respectively. The Extent data type describes the size or extent of an entity with respect to a measurement type (e.g. ‘4 miles’ and ‘3 weeks’). The Position data type describes the position in space or time of an entity in semantic terms by indicating the distance and direction or angle from an origin (e.g. ‘4 miles from New York City’ and ‘The third day of April’).

8.4.4 Semantic Relations

The semantic relation tools describe semantic relations. MPEG-7 has standardized common semantic relations such as *agent* and *time* but it allows the description of non-normative relations. Figure 8.1 shows examples of several semantic relationships between the objects, the event and the concept in the description. The normative semantic relations in MPEG-7 are listed in Table 8.2. As for structural relations, MPEG-7 has also standardized the inverse of each semantic relation. Figures 8.1, 8.14 and 8.15 illustrate examples of some of these relations.

The semantic relation tools include the SemanticRelation CS, which specifies semantic relations that apply to entities that have semantic information, which may be implicitly or explicitly stated. Normative semantic relations describe, among others, how

Table 8.2 Normative semantic relations in MPEG-7 listed by type

Type	Normative relations
Semantic	agent, agentOf, patient, patientOf, experiencer, experiencerOf, stimulus, stimulusOf, causer, causerOf, goal, goalOf, beneficiary, beneficiaryOf, them, themOf, result, resultOf, instrument, instrumentOf, accompanier, accompanierOf, summarizes, summarizedBy, state, stateOf combination, specializes, generalizes, similar, opposite, exemplifies, exemplifiedBy, interchangeable, identifier, part, partOf, contrasts, property, propertyOf, user, userOf, component, componentOf, substance, substanceOf, entailment, entailmentOf, manner, mannerOf, influences, dependsOn, membershipFunction key, keyFor, annotates, annotatedBy, shows, appearsIn, reference, referenceOf, quality, qualityOf, symbolizes, symbolizedBy, location, locationOf, source, sourceOf, destination, destinationOf, path, pathOf, time, timeOf, depicts, depictedBy, represents, representedBy, context, contextFor, interprets, interpretedBy



Figure 8.15 Picture of a Maya vessel whose semantic description is shown in Figure 8.16 [7]

semantic entities relate in a narrative, how the definitions of semantic entities relate and the localization of semantic entities in space, time and media. As shown in Figure 8.12, the normative semantic relations may apply to entities such as semantic entities (see Section 8.4.2), collections (see [2]), models (see [2]) and segments (see Section 8.3.1), among others.

Normative semantic relations may describe how several semantic entities relate in a narrative or story (e.g. *agent*, *patient* and *accompanier*). For example, semantic relations may describe relations between objects and/or events that draw upon typical linguistic relations such as an object being the agent, patient, experiencer, accompanier, stimulus, cause, goal, or instrument of an event, or an event being the result or a summary of another event, among others.

Other normative semantic relations may describe how the definitions of several semantic entities relate to each other (e.g. *combination*, *specializes* and *exemplifies*). As examples, semantic relations may describe a semantic entity being the combination of the meanings of two or more semantic entities or a semantic entity being a specialization, similar in meaning to, opposite in meaning to, or an example of another semantic entity, among others.

There are semantic entities that may also describe the spatial, temporal, or media localization of semantic entities in segments, models, collections, or other semantic entities (e.g. *depicts*, *symbolizes* and *context*). For example, semantic relations may describe an event taking place at a specific place and time; an object being perceived, referenced, or symbolized in a segment or collection; or a narrative world being the context or interpretation of another narrative world, among others.

Finally, the relations specified by the GraphRelation CS (e.g. *equivalent* [1]), the BaseRelation description scheme (e.g. *member* and *refines* [1]), the SpatialRelation CS (e.g. *above* and *left*, see Section 8.3.4) and the TemporalRelation CS (e.g. *precedes* and *during*, see Section 8.3.4) can also be used to describe relations among semantic entities.

To illustrate the use of the semantic entity and relation tools, consider the example in Figure 8.15 and Figure 8.16. This example shows the picture of a Maya vessel and its semantic description [7]. The textual description of the example in Figure 8.16 could be as follows: ‘Vessel created in Guatemala in the 8th Century that is an example of Maya art. The vessel is 14 cm tall and it has several paintings. The paintings show the realm of the lords of death with a death figure that dances and another figure that holds an axe and a hand stone. The paintings represent sacrifice.’ The semantic description comprises several semantic entities described using description schemes derived from the SemanticBase description scheme as indicated within parenthesis in Figure 8.16. Several relationships among the semantic entities are also described and represented as named arrows in Figure 8.16. The object ‘Vessel’ is the *patient of* the event ‘Create’, whose *location* and *time* are ‘Guatemala’ and ‘8th Century’, respectively. The object ‘Vessel’ is 14 cm tall, which is described using the SemanticState description scheme. The object ‘Vessel’ *exemplifies* the object ‘Maya art’. One of the *components of* the object ‘Vessel’ is the object ‘Paintings’, which *depicts* the narrative world ‘Paintings world’. The *location of* this narrative world is the place ‘realm’, which is also the *location of* the object ‘Lords of Death’. In the narrative world ‘Paintings world’, agent object ‘Death figure’ is the *agent of* the event ‘Dance’ and the event ‘Hold’ has the agent object ‘Figure’ as an *agent* and

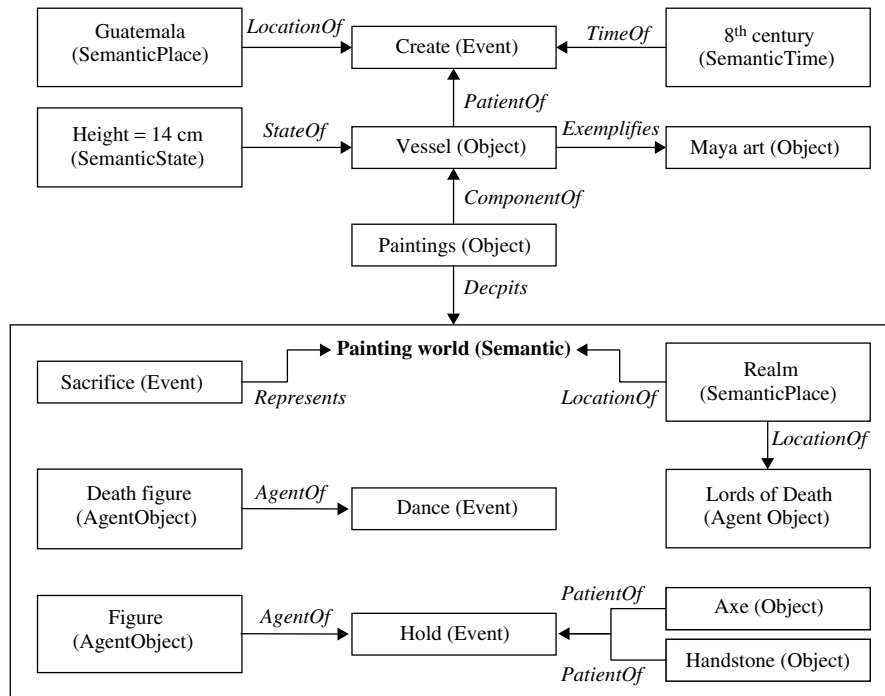


Figure 8.16 Semantic description for the picture shown in Figure 8.15 including several descriptions of semantic entities and relationships among them

the objects 'Axe' and 'Handstone' as *patients*. The narrative world 'Paintings world' is *represented* by the event 'Sacrifice'. All the relationships between semantic entities are normative.

8.4.5 Implicit and Explicit Semantic Description

The content semantic tools provide methods to create very brief or quite extensive semantic descriptions. Because certain relationships are considered more common than others, brevity in semantic description can be achieved by using implicit methods provided by these tools. By contrast, when it is desired to fully describe the interrelation of semantic entities occurring in the media, explicit methods may be used. There are three choices between implicit and explicit methods in creating MPEG-7 semantic descriptions. The first choice is the use of embedded relation tools in the SemanticBase description scheme versus graph tools in the Semantic description scheme. The second choice is the use of the Property element versus the Concept description scheme. The third choice is the use of recursion versus relation tools. These three choices shall be described in this section.

Relation tools in MPEG-7 are quite flexible. From within a particular semantic entity description, say an instance of the Object description scheme, any number of relations may be specified by giving the names of these relations and their targets. In this case, the source of the relation is assumed to be the object being described. One can also

describe a collection of the relations among the semantic entities in a narrative world using the Graph description scheme in the Semantic description scheme. Graph descriptions consist of lists of relation descriptions or of node and relation descriptions. The most explicit way of describing relations is to create an instance of the Graph description scheme with Node elements that reference the entities in the narrative world and the relations among them.

Properties may be described explicitly or implicitly. Implicit properties are described by using the Property element in the SemanticBase description scheme. The relation between such properties and the entity being described is automatically *propertyOf*. In order to indicate a more specific relationship, or allow multiple entities to be related to a single property, or specify the strength of a property, the property or group of properties must be described explicitly. This is done by using the Concept description scheme. Note that this also allows the property to be described as a semantic entity in its own right, with labels, a definition and relationships to other semantic entities.

Recursion allows another method for implicit relationships. Instances of both the Object and Event description schemes may contain other instances of the Objects and Event description schemes, respectively. The relation here is one of composition. It does not specify how the objects and events are composed or the relation between siblings in the decomposition. To do this, relations need to be made explicit, using the Relation or Graph description schemes. But for restricted domains, where these relations are well known, this recursion offers a shorthand for a semantic tree, completely complementary to the trees created by the segment entity tools.

Another implicit relation, the recursion of the Semantic description scheme and, for that matter, for any description scheme derived from the SemanticBag description scheme, deserves special mention. Although this may be represented without making the parent or sibling relations explicit, the meaning cannot be presumed, except by external semantics provided in the application domain. For this reason, this method of recursion usually will require explicit relations. An instance of the Semantic description scheme, for instance, is a description of a single narrative world. If it contains other instances of the Semantic description scheme, these must be either narrative worlds contained in the parent, contextual information, or information used in anaphoric (metaphor, simile, or analogy) description. Where there is an ambiguity that is not clarified by the current semantics, the role of the embedded instance of the Semantic description scheme must be made clear. An example of new semantics would be to derive 'Thesaurus' from the SemanticBag description scheme and specify that the instances of the Semantic description scheme contained in the Thesaurus are examples of complex media terms.

We illustrate the concepts by showing four ways to describe the relationships within the narrative world 'Painting world' in Figure 8.16 (the XML is in the chapter appendix included in the DVD accompanying this book). The first is to explicitly collect together the description of the relationships between the semantic entities, shown in Figure 8.16, into a graph description. In the second, we explicitly list the node descriptions of the graph; such an addition might be used if the graph description were to be used often with graph matching software. In the third, we do a full-blown, fully explicit description of the graph. The nodes are matched to the semantic entities of the narrative world 'Painting world', the relations are between the nodes in the graph, the relationships have associated

identifiers so that the graph may be referenced. The fourth example shows how such a graph can be abstracted. In this case, the graph is not particularly general (unless all Maya paintings depict the same scene, which might be the case for a ritual object), but its elements may then be filled in or matched to any descriptions of the objects, events and people populating the scene.

8.5 CONCLUSION

MPEG-7 provides a comprehensive suite of tools for describing individual multimedia documents that has the potential to revolutionize current multimedia applications. In particular, these tools can describe the content management, structure and semantics of multimedia content allowing efficient searching, filtering and browsing of multimedia content.

REFERENCES

- [1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Englewood Cliffs, N.J.
- [2] J. Smith and A. B. Benitez, Description of Collection of Audio-Visual content and Models, Chapter 10.
- [3] P. van Beek and J. Smith, Navigation and Access, Chapter 9.
- [4] P. van Beek, K. Yoon and A. M. Ferman, User Interaction, Chapter 11.
- [5] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of ACM*, **26**(11), 832–843 (1983).
- [6] http://www.getty.edu/research/institutestandards/cdwa/3.cataloging_examples/index.html, Accessed on 6/1/2001.
- [7] Getty Research Institute, Getty Standards Program, Categories for the Description of Works of Art. J. Paul Getty Trust and the College Art Association (2001). Cataloging Examples: Maya Vessel.
- [8] T. Walker, J. Heuer and J. M. Martinez, MPEG-7 Basic MDS elements, Chapter 7.