

**IST-2000-28304**



**DELIVERABLE # 6**

**Description Schemes for Smart  
Accessing of AV Content**



**The research was conducted within the SPATION project,  
supported by the E.U. grant no IST-2000-28304.**

**Version: 1.0  
Date: 28 November 2002  
Security level : Public**

## Table of content

<a href="#">1</a>	<a href="#">Table of content</a>	2
<a href="#">2</a>	<a href="#">Introduction</a>	4
<a href="#">2.1</a>	<a href="#">Purpose of the document</a>	4
<a href="#">2.2</a>	<a href="#">Structure of the document</a>	4
<a href="#">3</a>	<a href="#">Smart Multimedia Access</a>	5
<a href="#">4</a>	<a href="#">MPEG-7 description schemes for smart accessing of multimedia content</a>	6
<a href="#">4.1</a>	<a href="#">User Interaction</a>	8
<a href="#">4.1.1</a>	<a href="#">User Preferences</a>	9
<a href="#">4.1.2</a>	<a href="#">Usage History</a>	15
<a href="#">4.2</a>	<a href="#">Content Management</a>	18
<a href="#">4.2.1</a>	<a href="#">Media description tools</a>	18
<a href="#">4.2.2</a>	<a href="#">Creation &amp; production description tools</a>	21
<a href="#">4.2.3</a>	<a href="#">Usage description tools</a>	21
<a href="#">4.3</a>	<a href="#">Content Organization</a>	23
<a href="#">4.3.1</a>	<a href="#">Collections</a>	23
<a href="#">4.3.2</a>	<a href="#">Models</a>	27
<a href="#">4.3.3</a>	<a href="#">Probability models</a>	28
<a href="#">4.3.4</a>	<a href="#">Analytic models</a>	28
<a href="#">4.3.5</a>	<a href="#">CollectionModel DS</a>	30
<a href="#">4.3.6</a>	<a href="#">Cluster Models</a>	31
<a href="#">4.3.7</a>	<a href="#">Classification Models</a>	32
<a href="#">5</a>	<a href="#">MPEG-21 tools</a>	34
<a href="#">5.1</a>	<a href="#">User model</a>	34
<a href="#">5.2</a>	<a href="#">MPEG-21 Parts</a>	35
<a href="#">5.2.1</a>	<a href="#">Part 1: Vision, Technology and Strategy</a>	35
<a href="#">5.2.2</a>	<a href="#">Part 2: Digital Item Declaration</a>	35
<a href="#">5.2.3</a>	<a href="#">Part 3: Digital Item Identification</a>	36
<a href="#">5.2.4</a>	<a href="#">Part 4: Intellectual Property Management Tool Representation and Communication System</a>	36
<a href="#">5.2.5</a>	<a href="#">Part 5: Right Expression Language</a>	36
<a href="#">5.2.6</a>	<a href="#">Part 6: Right Data Dictionary</a>	36
<a href="#">5.2.7</a>	<a href="#">Part 7: Digital Item Adaptation</a>	36
<a href="#">5.3</a>	<a href="#">Digital Item Adaptation (Part 7)</a>	38
<a href="#">5.3.1</a>	<a href="#">User characteristics</a>	38
<a href="#">5.3.2</a>	<a href="#">Content preferences</a>	38
<a href="#">5.3.3</a>	<a href="#">PresentationPreferences</a>	39
<a href="#">5.3.4</a>	<a href="#">AccessibilityCharacteristics</a>	39
<a href="#">5.3.5</a>	<a href="#">MobilityCharacteristics</a>	40
<a href="#">6</a>	<a href="#">SPATION's user Interface and MPEG-7/21 tools</a>	40
<a href="#">6.1</a>	<a href="#">User Interface functionalities support</a>	40
<a href="#">6.1.1</a>	<a href="#">Device browser</a>	40
<a href="#">6.1.2</a>	<a href="#">Video, Music, and Photo browser</a>	41
<a href="#">6.2</a>	<a href="#">The UMA project</a>	43
<a href="#">6.2.1</a>	<a href="#">UMA demos</a>	45
<a href="#">7</a>	<a href="#">Content and Metadata Transcoding</a>	45
<a href="#">7.1</a>	<a href="#">Content Transcoding</a>	45

---

<u>7.2</u>	<u>Metadata Adaptation</u> .....	45
<u>7.2.1</u>	<u>A use case</u> .....	46
<u>7.2.2</u>	<u>Experiment description</u> .....	46
<u>7.2.3</u>	<u>Extension of MPEG-7/21 Tools</u> .....	51
<u>7.2.4</u>	<u>Metadata Adaptation Engine Specification</u> .....	53
<u>8</u>	<u>Extraction Methods</u> .....	53
<u>8.1</u>	<u>Detection of salient events in soccer games</u> .....	54
	<u>The Low-level Descriptors</u> .....	54
	<u>The proposed algorithm based on Controlled Markov Chain Model</u> .....	55
	<u>Experimental Results</u> .....	58
<u>8.2</u>	<u>Metadata Integration</u> .....	58
	<u>Integration of different descriptions</u> .....	59
	<u>Performance Evaluation</u> .....	66
<u>9</u>	<u>Conclusions</u> .....	68
<u>10</u>	<u>Bibliography</u> .....	69

# 1 Introduction

## 1.1 Purpose of the document

The purpose of this document is to present the available tools that can be useful to easily access and manage multimedia documents, such as audio-visual sequences or content descriptions, in a multimedia home network.

While several partial solutions to this problem have been proposed, only two recently introduced standards aim to provide a more general and complete set of tools: MPEG-7 and MPEG-21. The objective of MPEG-7 is to provide a set of descriptors and descriptions schemes that allow a standard description of a certain multimedia content. The ongoing MPEG-21 standard, also called Multimedia framework, is aiming to provide solutions for the transparent delivery and consumption of any type of digitally stored information.. This includes both content and content descriptions.

Instead of finding ad-hoc solutions for the specifically addressed SPATION problem, it has been decided to adopt the aforementioned standards even if they are not completely defined (MPEG-21) and not yet widely supported by marketed multimedia products (MPEG-7).

This means that SPATION will serve as test bed for these two standards and, at the same time, will serve as an example of how those can be used to develop consumer and professional applications. Starting from the requirements set by the spation scenarios a detailed analysis of the solutions offered by MPEG7 and MPEG21 is provided.

## 1.2 Structure of the document

In Section 2 of this report an introduction to the problem of digital document access in a multimedia network is presented and discussed.

Section 3 and 4 are devoted to an overview of the current MPEG-7 and MPEG-21 proposals, focusing on the proposed standard tools that could be used to facilitate the access and management of multimedia content and meta-data.

In Section 5 the MPEG-7 tools introduced in the previous section are reconsidered with respect to the user's interface functionalities defined in the SPATION Deliverable 5.

In Section 5.2 a related project, named Universal Multimedia Access (UMA), is described, and the analogies with SPATION are discussed. In particular we present the adopted architectural solution, some open issues and some demos

A further abstraction level is then described in Section 6, where the MPEG-21 multimedia framework is adopted to implement a transparent delivery and consumption of multimedia content and descriptions.

Finally in section 7 extration of events in a football match is presented as a case study.

## 2 Smart Multimedia Access

One of the key aspects of the SPATION project is the transparent access and use of multimedia resources and services in a heterogeneous home network. This topic can be decomposed into two main problems, one related to the network technology and the other related to the possible solutions that can guarantee application interoperability.

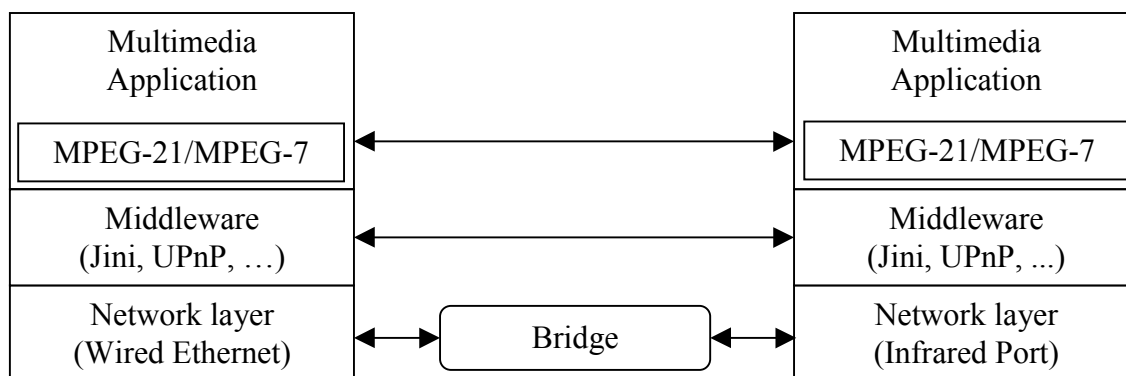
The first topic has been described in Deliverable 2 [1] of the project, where an overview of the current state of the art on multimedia network technology solutions is presented. In the above mentioned document, it has been pointed out how available solutions allow the interconnection and communication between different digital devices such as PC, Set-top Box and Handheld devices.

Hence, in this document, it will be assumed that a transparent network, in the sense that any application can access resources and services distributed through the network, is available. We will focus on how interoperability issues can be solved at the application level of the network stack, considering the distribution and fruition of multimedia content.

More in detail, this document aims to identify the technologies that can be useful to support the management of the overall multimedia resources across the home-network, taking into account the SPATION scenarios [2].

Figure 1 describes the basic network architecture that will be adopted in the development of the project. With respect to the network architecture described in Deliverable 2, one more sub-layer has been introduced between the Multimedia Application and the Middleware layer.

This sub-layer will allow applications to use and manage resources distributed over the network.

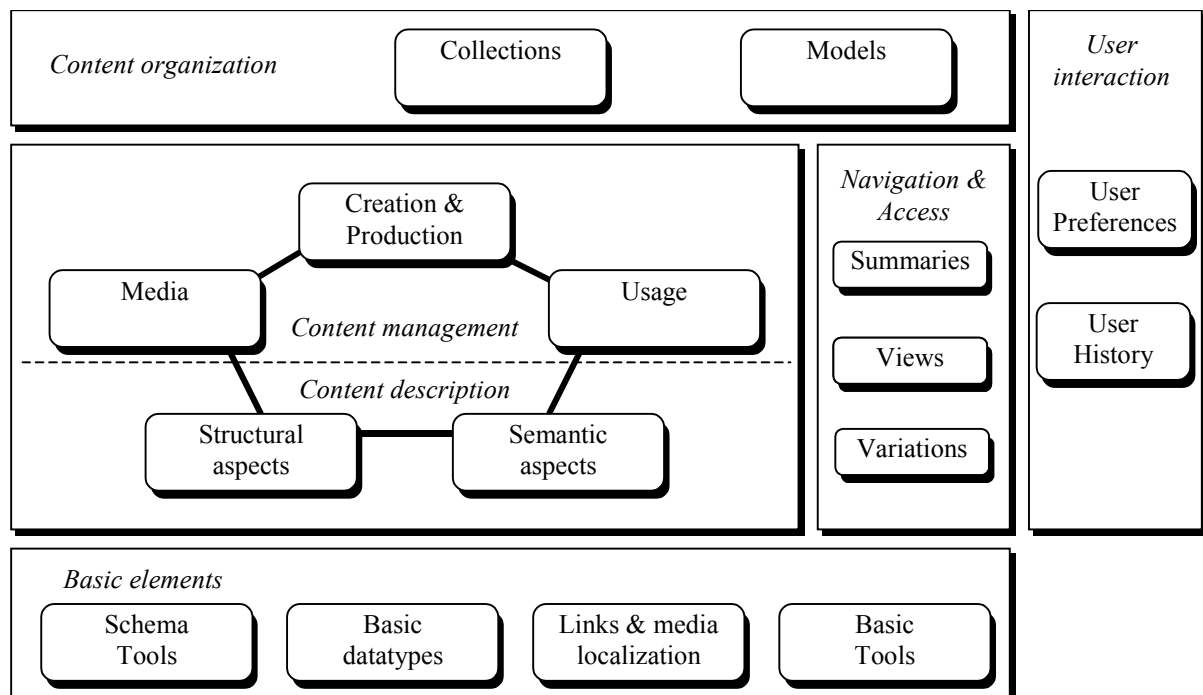


**Figure 1:** Interoperability solutions, full middleware support.

### **3 MPEG-7 description schemes for smart accessing of multimedia content**

Considering what type of descriptors would be needed for the SPATION scenarios. Mainly three types of information can be identified. The first is related to the description of the network and terminal capabilities, the second is related to the users preferences, and the last one is about multimedia content description, collection, summaries, etc. In the following sections we will investigate to which MPEG7 descriptors this information can be mapped. In chapter 4 we will described where MPEG21 can be applied

The MPEG-7 standard contains a part (Part 5) called MULTIMEDIA DESCRIPTION SCHEMES (MDS). The MDS is structured as shown in Figure 2.



**Figure 2.** Overview of the MDS.

The *User Interaction*, shown on the top left in figure 3, describes preferences of users pertaining to the consumption of the AV content, as well as usage history. In fact, the MPEG-7 AV content descriptions can be matched to the preference description in order to select and personalize AV content for more efficient and effective access, presentation and consumption.

Also, MPEG-7 provides DSs for AV content management (see *Content Management* in Figure 2); these tools describe the information about creation and production, media coding and storage, file formats, content usage.

Finally, another set of tools, defined in *Content Organization*, addresses the organization of the content by classification, by the definition of collections of multimedia documents and by modeling.

In the following sections *User Interaction* (section 3.1), *Content Management* (section 3.2) , and the *Content Organization* (section 3.3) will be analyzed .

### 3.1 User Interaction

This clause defines the tools (Table 1) related to user interaction with multimedia content.

<i>Tool</i>	<i>Functionality</i>
<b>User Preferences</b>	Tools for describing user preferences pertaining to multimedia content, enabling effective user interaction and personalization of content access and consumption.
<b>Usage History</b>	Tools for describing usage history of users of multimedia content, enabling effective user interaction and personalization of content access and consumption.

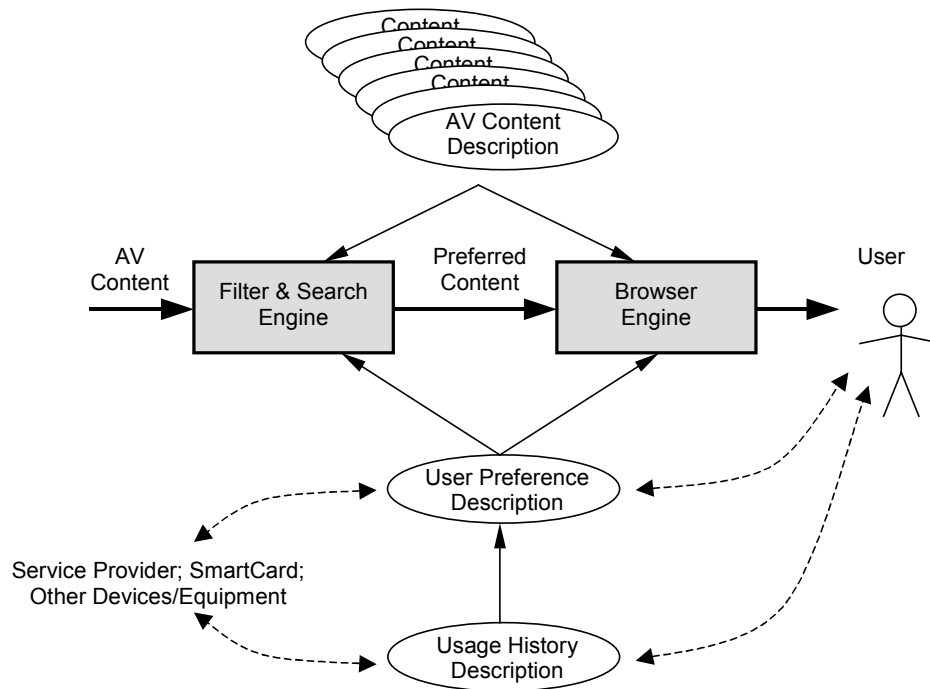
**Table 1.** User interaction tools.

The key concepts used in this clause are illustrated in Figure 3.

A user interacts with multimedia content by using a multimedia system. The multimedia system is used to find multimedia content, e.g. by searching or filtering, and to consume multimedia content, e.g., by viewing or listening. Descriptions of the multimedia content are provided to the system to enable efficient searching, filtering and browsing.

Descriptions of the user's preferences are also provided to the system to enable personalized searching, filtering and browsing of multimedia content. The descriptions of the user's preferences are used to find preferred multimedia content and to present preferred views of the content automatically.

The multimedia system may also generate a usage history description based on a history of the user's interactions with the multimedia content. The usage history descriptions may be used directly for personalized searching, filtering and browsing, or may be mapped to a description of the user's preferences. Both user preferences descriptions and usage history descriptions may be exchanged with third parties (e.g., service providers) or with other devices.



**Figure 3.** Personalized filtering, search and browsing of multimedia content.

### 3.1.1 User Preferences

The *User Preference* DS describes preferences for different types of content and modes of browsing, including context dependency in terms of time and place. Additionally, the *User Preference* DS describes the weighting of the relative importance of different preferences, the privacy characteristics of the preferences and whether preferences are subject to update, such as by an agent that automatically learns through interaction with the user.

#### Basic user preferences tools

- PreferenceConditionType → Describes a combination of time and/or place associated with a set of user preferences. The combination of time and/or place forms the condition under which the particular set of preferences applies.
  - Place → Place associated with user preference.
  - Time → Time associated with user preferences.
- UserChoiceType → Used to indicate the value of a condition set by a user, with respect to actions taken by a processor of descriptions. The values allowed are defined as follows.
  - *True*: indicates that the condition is true.
  - *False*: indicates that the condition is false.
  - *User*: indicates that the user must be asked whether the condition is true or false.
- PreferenceValueType → Used to indicate the value, priority or weight assigned to a particular preference element, relative to sibling preference elements of the same type in a description. The range of the preference values is from -100 to 100. Positive integer values indicate preference. Negative values indicate non-preference. The zero value

indicates that the user is neutral in terms of preference versus non-preference. A default, positive, value of 10 corresponds to a nominal preference.

### **UserPreferences DS**

- UserPreferencesType → Specifies preferences pertaining to consumption of multimedia content of a particular user.
- UserIdentifier → Identifies a particular preference description of a user.
- UsagePreferences → Describes user preferences.

### **UserIdentifier datatype**

- UserIdentifierType → Identifies a particular preference description of a user.
- UserName → Name associated with the user, or name given to a particular set of user preferences.
- Protected → Indicates whether the user desires to keep the identifier information private. The values allowed are defined as follows.
  - *True*: indicates that the identifier information may only be accessed by the user and may not be communicated to external parties.
  - *False*: indicates that the identifier information does not have to be kept private and may be communicated to external parties, such as service providers and trusted software agents.
  - *User*: indicates that the user must be asked for approval on a case by case basis.

### **UsagePreferences DS**

- UsagePreferencesType → A container DS that contains two types of user preferences:
  - FilteringAndSearchPreferences DS.
  - BrowsingPreferences DS.
- FilteringAndSearchPreferences → Describes user's preferences for filtering of multimedia content or searching for preferred multimedia content.
- BrowsingPreferences → Describes user's preferences for multimedia content navigation and browsing.
- AllowAutomaticUpdate → Indicates whether the user permits automatic update of the usage preferences information, e.g., by a software agent. The values allowed are defined as follows.
  - *True*: indicates that associated preferences may be automatically updated, e.g. by a trusted software agent.
  - *False*: indicates that associated preferences may not be automatically updated, i.e., only the user is allowed to update the preference information.
  - *User*: indicates that the user should be asked for permission on a case by case basis.

### **FilteringAndSearchPreferences DS**

- FilteringAndSearchPreferencesType → Describes user's preferences for filtering of multimedia content or searching for preferred multimedia content. Preferred content is determined by matching individual components or combinations of components of a

FilteringAndSearchPreferences description against descriptions of multimedia content. First level preference components are CreationPreferences, ClassificationPreferences and SourcePreferences. Each of these elements in turn contains second level preference components. A FilteringAndSearchPreferences element may optionally contain other FilteringAndSearchPreferences elements as its children, to specify hierarchically structured preferences. In this case, the filtering and search preferences of the children elements apply on the condition that the preferences contained in their ancestor nodes are satisfied by matching multimedia content.

- CreationPreferences → Describes the user's preference related to media creation descriptions.
- ClassificationPreferences → Describes the user's preference related to media classification descriptions.
- SourcePreferences → Describes the user's preference for a particular source of media.
- PreferenceCondition → Identifies the usage condition(s) for a particular filtering and search preference description in terms of time and place. If multiple PreferenceCondition elements are present, the preferences apply under all conditions indicated.
- FilterAndSearchPreferences → Describes child preferences in a hierarchy of user preferences. The FilteringAndSearchPreferences of children nodes are conditioned on the preference items and preferences values described by the current node. In other words, when one or more children FilteringAndSearchPreferences nodes are present, each child node describes preferences that would apply under the condition that the preferences contained in its parent node are satisfied.
- Protected → Indicates whether the user desires to keep the filtering and search preferences information private. The values allowed are defined as follows.
  - *True*: indicates that the filtering and search preferences information may only be accessed by the user and may not be communicated to external parties.
  - *False*: indicates that the filtering and search preferences information does not have to be kept private and may be communicated to external parties, such as service providers and trusted software agents.
  - *User*: indicates that the user must be asked for approval on a case by case basis.
- PreferenceValue → Describes the relative priority or weight assigned to a particular FilteringAndSearchPreferences description, in case multiple filtering and search preference descriptions are present. When combinations of CreationPreferences, ClassificationPreferences and/or SourcePreferences are used, the preferenceValue attribute of FilteringAndSearchPreferences gives the relative preference value of the particular filtering and search preference description of the combined preferences, not each individual sub preference.

### ***CreationPreferences DS***

- CreationPreferencesType → Specifies user preferences related to the creation of the content.

- Title → Describes user's preference for the title of the content. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Title.
- Creator elements → Describes user's preference for the creator of the content (e.g., director or actor). A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Creator elements.
- Keyword → Describes a textual keyword that indicates user's preferred content. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Keyword elements.
- Location → Describes user's preference for the location where the content is created. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Location elements.
- DatePeriod → Describes user's preference for a period of time when the content was created. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other DatePeriod elements.
- preferenceValue → Describes the relative priority or weight assigned to a creation preference description, in case multiple preference descriptions are present.

### *ClassificationPreferences DS*

- ClassificationPreferencesType → Specifies user preferences related to the class of the content.
- Country → Describes user's preference for country of origin of the content. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Country elements.
- DatePeriod → Describes user's preference for a period of time when the content was first released. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other DatePeriod elements.
- Language → Describes user's preference for language of origin of the content. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Language elements.
- Genre → Describes user's preference for the genre of the content. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Genre elements.
- Subject → Describes user's preference for the subject of the multimedia content. The subject classifies multimedia content from a point of view of types of content, without considering genre classification. A preferenceValue attribute may be attached to each element to indicate its relative priority with respect to other Subject elements.

- **MediaReview** → Describes user's preference with respect to reviews of the content. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **MediaReview** elements.
- **ParentalGuidance** → Describes user's preference for parental rating of the multimedia content. The parental rating is specified according to a rating scheme, which may vary from one organization to another, or from one country to another. **ParentalGuidance** includes the specification of the country, parental rating scheme and the parental rating value under that particular scheme. It also supports rating on the basis of age classification for targeted audiences. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **ParentalGuidance** elements.
- **preferenceValue** → Describes the relative priority or weight assigned to a particular classification preference description, in case multiple preference descriptions are present.

### *SourcePreferences DS*

- **SourcePreferencesType** → Specifies preferred source of content that is available for consumption (i.e. published).
- **PublicationType** → Describes user's preference for publication medium or delivery mechanism of content, e.g. terrestrial broadcast, web-cast, streaming, CD-ROM, etc. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **PublicationType** elements.
- **PublicationSource** → Describes user's preference for a particular source of content, referring to for example a broadcast channel or a server. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **PublicationSource** elements.
- **PublicationPlace** → Describes user's preference for the location from where the content is distributed. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **PublicationPlace** elements.
- **PublicationDate** → Describes user's preference for time and date (period) of the content availability, e.g., time and date of the broadcast, or time and date of availability for an on-demand content. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **PublicationDate** elements.
- **Publisher** → Describes user's preference for a publisher of the content, i.e. the person, group or organization that makes the content available for consumption, e.g. a broadcaster or distributor. A **preferenceValue** attribute may be attached to each element to indicate its relative priority with respect to other **Publisher** elements.
- **MediaFormat** → Describes user's preference on the format of the content, e.g. video format, audio format, aspect ratio, etc. A **preferenceValue** attribute may be attached to

each element to indicate its relative priority with respect to other MediaFormat elements.

- noRepeat → Indicates whether the user prefers content that is not a repeat of content made available before (optional).
- noEncryption → Indicates whether the user prefers content that is not encrypted (optional).
- noPayPerUse → Indicates whether the user prefers content that is not available on a pay-per-use basis (optional).
- preferenceValue → Describes the relative priority or weight assigned to a particular SourcePreferences description, in case multiple source preference descriptions are present.

### ***BrowsingPreferences DS***

- SummaryPreferencesType → Describes user's preferences regarding to media summaries and their visualization and sonification.
- SummaryTypePreference → Describes the type of the preferred media summary and corresponds to the type of components in a media summary description. A preferenceValue attribute may be attached to each SummaryTypePreference element, to indicate the relative priority or weight of each summary type.
- PreferredSummaryTheme → Describes the preferred summary names or names/themes for media segments where the names/themes are included in summary and segment media descriptions. A preferenceValue attribute may be attached to each PreferredSummaryTheme element, to indicate the relative priority or weight of each summary theme.
- NumOfKeyFrames → Describes the preferred number of keyframes in a visual summary.
- MinNumOfKeyFrames → Describes the minimum number of keyframes in a visual summary.
- MaxNumOfKeyFrames → Describes the maximum number of keyframes in a visual summary.
- SummaryDuration → Describes the preferred duration for an AV summary of media.
- MinSummaryDuration → Describes the minimum duration for an AV summary of media.
- MaxSummaryDuration → Describes the maximum duration for an AV summary of media.

- NumOfChars → Describes the preferred length, in number of characters, of a textual summary.
- MinNumOfChars → Describes the minimum length, in number of characters, of a textual summary.
- MaxNumOfChars → Describes the maximum length, in number of characters, of a textual summary.
- preferenceValue → Describes the relative priority or weight assigned to a particular SummaryPreferences description, in case multiple summary preference descriptions are present.
- SummaryComponentType → A datatype that enumerates options for preferred types of summaries of AV content. The options are defined as follows.
  - *visual*: indicates preference for a visual summary.
  - *visual/keyFrames*: indicates preference for a visual summary that includes key-frames.
  - *visual/keyVideoClips*: indicates preference for a visual summary that includes key-video clips.
  - *visual/keyThemes*: indicates preference for a visual summary that includes key-themes.
  - *audio*: indicates preference for an audio summary.
  - *audio/keySounds*: indicates preference for an audio summary that includes key-sounds.
  - *audio/keyAudioClips*: indicates preference for an audio summary that includes key-audio clips.
  - *audio/keyThemes*: indicates preference for an audio summary that includes key-themes.
  - *textual*: indicates preference for a textual summary.
  - *textual/keyThemes*: indicates preference for a textual summary that includes key-themes.

### 3.1.2 Usage History

The *Usage History* DS describes the history of actions carried out by a user of a multimedia system. The usage history description can be exchanged between consumers, their agents, content providers and devices and may, in turn, be used to determinate the user's preferences with regard to AV content.

#### UsageHistory DS

- UsageHistoryType → Specifies user's multimedia content consumption history.
- UserIdentifier → Identifies the individual for whom the usage history is provided. This element is of type UserIdentifierType and contains the protected attribute. Thus the identity of the user shall not be disclosed unless this attribute is set to false.

- **UserActionHistory** → Describes history of the actions the user has carried out during the observation period.
- **AllowCollection** → Indicates whether the user permits usage history of his/her actions to be collected. The values allowed are defined as follows.
  - *True*: indicates that usage history data may be collected.
  - *False*: indicates that usage history data may not be collected.
  - *User*: indicates that the user should be asked for permission on a case by case basis.

### **UserActionHistory DS**

- **UserActionHistoryType** → Specifies a history of the actions carried out by the user.
- **ObservationPeriod** → Describes the time period(s) during which the associated history items have been recorded. Multiple instance of ObservationPeriod can be used to represent discontinuous time periods.
- **UserActionList** → Describes a list of actions of the same type, i.e. all actions in the UserActionList carry the same ActionType value.
- **protected** → Indicates whether the user desires to keep the user action history information private. The values allowed are defined as follows.
  - *True*: indicates that the user action history information may only be accessed by the user and may not be communicated to external parties.
  - *False*: indicates that the user action history information does not have to be kept private and may be communicated to external parties, such as service providers and trusted software agents.
  - *User*: indicates that the user must be asked for approval on a case by case basis.

### **UserActionList DS**

- **UserActionListType** → Describes a list of user actions, all of the same type.
- **ActionType** → Specifies the type of action performed by the user, e.g., "View," "Pause," "Play," etc. All UserAction elements in an ActionList have the same ActionType. An example of a classification schemes with terms for ActionType is MPEG7ActionTypeCS.
- **UserAction** → Specifies a single user action in a list of actions. Each action is associated with a single multimedia program.
- **numInstances** → Specifies the number of UserAction elements in a UserActionList. (e.g. 21 "Record" actions; 5 "View" actions; etc.).
- **totalDuration** → Specifies the total time spent by the user during the observation period performing a specific action (e.g. 32 minutes for a "Record" action).

*UserAction DS*

- UserActionType → Describes a single user action.
- ActionTime → Specifies the time that the action took place and, if applicable, its duration (e.g. for "Play," "Pause," etc.). The time of occurrence of the action can be described in two ways: by ActionMediaTime and/or by ActionGeneralTime. The duration of a UserAction refers to the duration in terms of the media time, which is identical to the duration in UTC for a large number of action types, but may be different for such action types as "Repeat" or "FastForward."
- ActionMediaTime → Action time relative to the time reference established for the given media. This time referencing method is useful for such action items as "Repeat" or "FastForward," and for manipulating content on the user's local system (such as personal CDs or DVDs).
- ActionGeneralTime → Action time relative to Coordinated Universal Time (UTC) in the Gregorian date/time format.
- ProgramIdentifier → Unique identifier of the program that is associated with the given action. Each Action is associated with a single program and, consequently, a single ProgramIdentifier.
- ActionDataItem → Refers to a specific part of the description of the multimedia content, or to other material related to the action (e.g. the URL the user chooses to follow in an enhanced TV application). Shall refer to a multimedia content description instance.

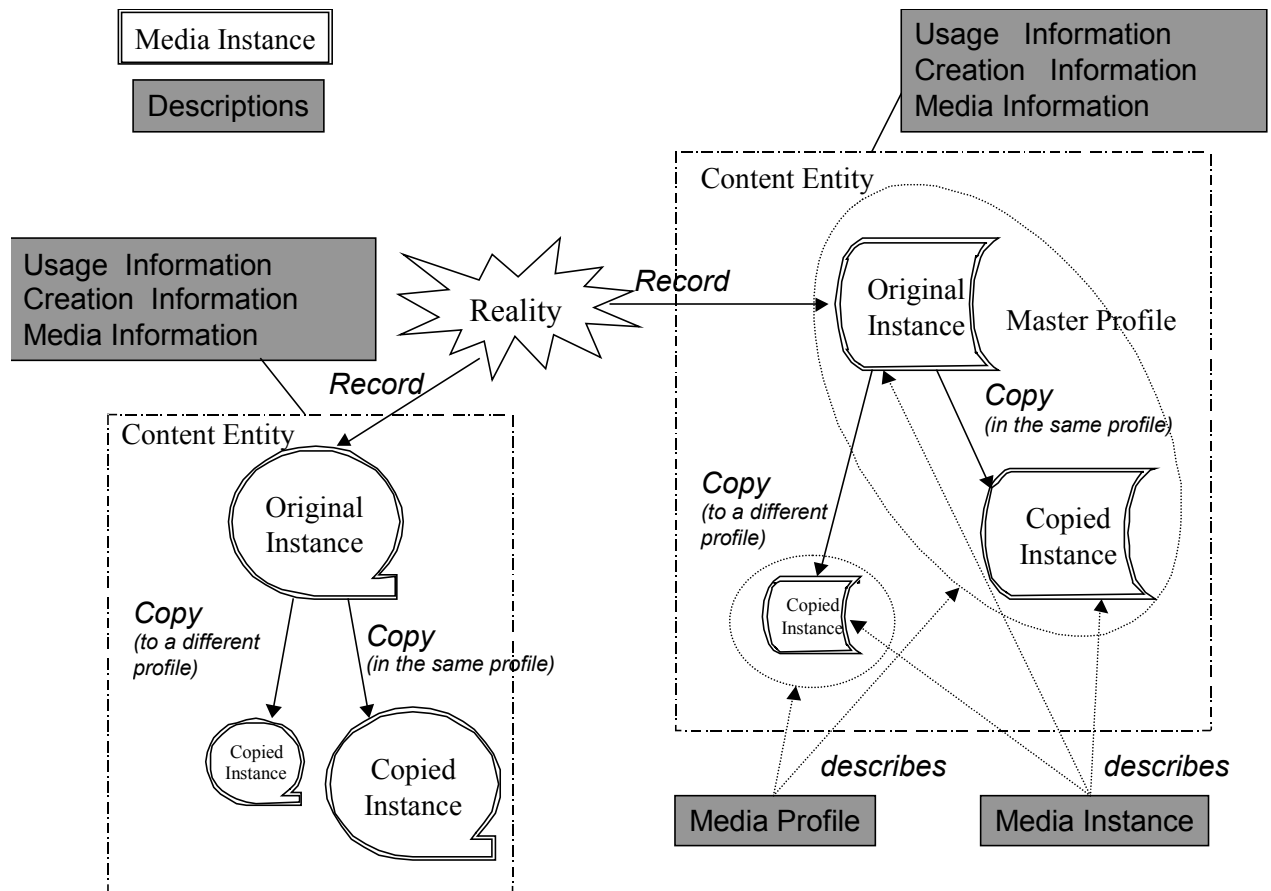
## **3.2 Content Management**

*Content Management* defines the description tools to describe the content of a single multimedia document from the point of view of its creation and production, its media and its usage.

### **3.2.1 Media description tools**

This clause specifies the description tools for describing the media features of the coded multimedia data. The multimedia data described by descriptions can be available in different modalities, formats, coding versions, and there can be multiple instances. The description tools defined in this clause can handle all of these different aspects.

For example, a concert can be recorded in two different modalities: audio and AV. Each of these modalities can be encoded by different encoding algorithms. This creates several media profiles. Finally, several instances of the same encoded multimedia content may be available. These concepts of modality, profile and instance are illustrated in Figure 4.



**Figure 4.** Model for content entity, profile and instance.

Referring to Figure 4:

- Content entity: One reality such as a concert in the world can be represented as several types of media, e.g. audio media, AV media. A content entity has a specific structure to represent the reality.
- Media Information: Physical format of a multimedia content is described by the MediaInformation DS. One instance of the MediaInformation DS is attached to one content entity to describe it. The DS includes an identifier of the content entity and a set of descriptors for the format of the entity.
- Media Profile: One content entity can have one or more media profiles that correspond to different coding versions of the entity. One of the profiles is the original one, called master profile. It corresponds to the initially created or recorded profile. The others are transcoded from the master. If the multimedia content is encoded with the same encoding software or hardware but with different parameters, different media profiles are created.
- Media Instance: A content entity can be instantiated as physical entities called media instances. An identifier and a locator specify a media instance.
- CreationInformation (see subclause 3.2): The information about the creation process of a content entity is described by the CreationInformation DS. One description of the DS may be attached to one content entity to describe it.
- UsageInformation (see subclause 3.3): The UsageInformation DS describes the usage-related information of a content entity. One description of the DS may be attached to one content entity to describe it.

#### **MediaInformation DS**

- MediaInformationType → Describes the physical format of the multimedia data.
- MediaIdentification → Identifies the multimedia content entity representing a reality. This D is characterized by a set of tools: EntityIdentifier, AudioDomain, VideoDomain, ImageDomain.
- MediaProfile → Describes one Media Profile of the multimedia content; the different Media Profile instances allow the description of the different sets of coding parameters values available for different coding profiles. This DS is composed of other Ds and DSs: MediaFormat, MediaTranscodingHints, MediaQuality, MediaInstance, ComponentMediaProfile.

### 3.2.2 Creation & production description tools

This clause specifies the description tools for describing author-generated information about the generation/production process of the multimedia content. This information is related to the AV multimedia content but it is not explicitly depicted in the actual multimedia content and, usually, cannot be extracted from the multimedia content itself.

The meta information describing the creation and production of the content are, typically, features such as title, creator, classification, purpose of the creation, etc.

#### **CreationInformation DS**

The CreationInformation DS describes author-generated information about the creation/production process of a multimedia content. The CreationInformation DS contains:

- Information about the creation process not perceived in the material (e.g, the author of the script, the director, the character, the target audience, etc.) and information about the creation process perceived in the multimedia content (e.g, the actors in the video, the players in a concert).
- Classification related information (target audience, style, genre, rating, etc.).

The tools specified in this subclause are:

- CreationInformationType → Describes creation features of the multimedia content.
- Creation → Describes by whom, when, and where, etc, the multimedia content was created.
- Classification → Describes user oriented and service oriented classification of the multimedia content. The resulting descriptions facilitate searching and filtering of multimedia content based on user preferences (e.g, language, style, genre, etc.) and service-oriented classifications (e.g, purpose, parental guidance, market segmentation, media review, etc.).
- RelatedMaterial → Describes additional information about the material related to the multimedia content.

### 3.2.3 Usage description tools

This clause specifies the description tools describing information about the usage of the multimedia content. Meta information related to the usage of the content: typical features involve rights holders, access right, publication, and financial information. This information may very likely be subject to change during the lifetime of the multimedia content.

#### **UsageInformation DS**

The UsageInformation DS describes information about the usage process of the multimedia content (e.g. an AV program, a video, an image, an audio, ..).

The UsageInformation DS contains:

- Information about the rights for using the multimedia content.
- Information about the ways and means to use the multimedia content (e.g. edition, emission, etc.) and the results of the usage (e.g. audience).
- Financial information including the financial results of the production (in the Financial D within the UsageInformation DS), and of the publication (in the Financial D within each Availability DS UsageRecord DS) of the multimedia content.

It is important to note that new descriptions may be added each time the content is used (e.g. UsageRecord, Income), or when there are new ways to access to the multimedia content for some kind of applications, if required. In such case, the size of the description might significantly increase. This information should be taken into account when designing the systems and applications using this descriptions.

The tools specified in this subclause are:

- UsageInformationType → Describes usage features of the multimedia content.
- Rights → Describes information about the owners of the rights corresponding to the multimedia content, and how the multimedia content can be used. This D include also the RightsId D that identifies the link to the current Rights Owner information.
- FinancialResults → Describes the cost of the creation of the multimedia content and the income the multimedia content has generated. The incomes vary with time. This D is composed of other Ds and DSs: AccountItem, EffectiveDate, CostType, IncomeType, Currency, Value.
- Availability → Describes where, when, how, and by whom the multimedia content can be used. This DS includes: InstanceRef, PublicationType, OriginPlace, Distributor, Financial, Rights, AvailabilityPeriod.
- UsageRecord → Describes where, when, how, and by whom the multimedia content was used. This Ds includes: AvailabilityRef, Audience, Financial.

### 3.3 Content Organization

This clause specifies tools for describing the organization and modeling of multimedia content. The tools used for describing collections and models are listed in Table 2.

<i>Tools</i>	<i>Functionality</i>
Collection (section 3.3.1)	These tools describe unordered sets of multimedia content, segments, descriptors, concepts, or mixed sets of the above.
Model (section 3.3.2)	These tools describe parameterized models of multimedia content, descriptors or collections.
ProbabilityModel (section 3.3.3)	These tools describe the association of statistics or probabilities with the attributes of multimedia content, descriptors or collections.
AnalyticModel (section 3.3.4)	These tools describe the association of labels or semantics with multimedia content or collections.
CollectionModel (section 3.3.5)	The CollectionModel DS describes the association of semantics with a Collection
Cluster Model (section 3.3.6)	These tools describe the association of labels or semantics, and statistics or probabilities with multimedia content collections.
Classification Model (section 3.3.7)	These tools describe information about known collections of multimedia content in terms of labels, semantics and models that can be used to classify unknown multimedia content.

**Table 2.** List of content organization tools.

#### 3.3.1 Collections

This subclause specifies tools for describing collections related to multimedia content. The tools specified in this subclause are:

- CollectionType → Describes a collection related to multimedia content, for example a collection of multimedia data, descriptors, or semantic concepts.
- CreationInformation → Describes information related to the creation of the Collection.
- UsageInformation → Describes information related to the usage of the Collection.
- TextAnnotation → Describes a TextAnnotation of the Collection.
- Summarization → Describes a Summarization of the Collection.
- Collection → Describes a child Collection that is nested within the Collection. The child Collection shall be of the same type as the parent Collection.
- CollectionRef → Describes a reference to a child Collection that is nested within the Collection. The child CollectionRef shall refer to a Collection of the same type as the parent Collection.

- Name → Identifies the name of the Collection.

In this subclause other tools are defined: ContentCollection DS, SegmentCollection DS, DescriptorCollection DS, ConceptCollection DS, MixedCollection DS, StructuredCollection DS.

### **ContentCollection DS**

The ContentCollection DS describes collections of multimedia data, such as video, audio and multimedia material, images, sounds, and so forth. The ContentCollection DS provides the basic functionality of describing a grouping of multimedia data into an unordered, unbounded, nested set. The tools referring to this DS are:

- ContentCollectionType → Describes a collection of multimedia content, such as video, audio, images, sounds, and so forth. The ContentCollection may contain a mix of different types of multimedia content within a single description.
- VisualFeature → Describes an aggregated visual feature of the content collection. For example, the VisualFeature can describe the aggregated color of the images in an image collection using a Group of Pictures (GoF/GoP) color descriptor. The VisualFeature applies only in the case of a homogeneous collection of visual content such as a collection of images or video.
- AudioFeature → Describes an aggregated audio feature of the content collection. The AudioFeature applies only in the case of a homogeneous collection of audio content such as a collection of songs or speech documents.
- Content → Describes content that makes up the ContentCollection.
- ContentRef → Describes a reference to content that makes up the ContentCollection.

### **SegmentCollection DS**

The SegmentCollection DS describes collections of segments, such as video segments, audio segments, still regions, and so forth, possibly from different multimedia content. The SegmentCollection DS provides the basic functionality of describing a grouping of segments into an unordered, unbounded, nested set.

The tools referring to this DS are:

- SegmentCollectionType → Describes a collection of segments, such as video segments, audio segments, still regions, and so forth, possibly from different multimedia content. The SegmentCollection may contain a mix of different types of segments within a single description.
- Segment → Describes a segment that makes up the ContentCollection.
- SegmentRef → References an existing description of a segment that makes up the ContentCollection.

### **DescriptorCollection DS**

The DescriptorCollection DS describes collections of descriptors of multimedia content. For example, these tools can be used to describe a collection of color feature or shape descriptions. In general, the elements of the DescriptorCollection can be instances of descriptions of color, texture, shapes, motion, and so forth. The DescriptorCollection DS provides the basic functionality of describing a grouping of descriptor instances into an unordered, unbounded, nested set.

The tools referring to this DS are:

- DescriptorCollectionType → Describes a collection of instances of a particular type of descriptor. The DescriptorCollection is limited to a single type of descriptor within a single description.
- Descriptor → Describes the instance of the multimedia content descriptor by giving the value of the Descriptor. The DescriptorCollection shall be limited to one type of Descriptor per instance.

### **ConceptCollection DS**

The ConceptCollection DS describes a collection of semantic concepts related to multimedia content, such as multimedia objects and events. The ConceptCollection DS provides the basic functionality of describing a grouping of concepts into an unordered, unbounded, nested set.

The tools referring to this DS are:

- ConceptCollectionType → Describes a collection of semantic concepts related to multimedia content, such as multimedia objects and events. The ConceptCollection may contain a mix of different types of concepts within a single description.
- Concept → Describes a semantic concept belonging to the ConceptCollection.
- ConceptRef → Describes a reference to a semantic concept belonging to the ConceptCollection.

### **MixedCollection DS**

The MixedCollection DS subclause specifies tools for describing mixed collections of multimedia content, descriptors and semantic concepts related to multimedia material. The MixedCollection DS provides the basic functionality of describing a grouping of these items into an unordered, unbounded, nested set.

The tools referring to this DS are:

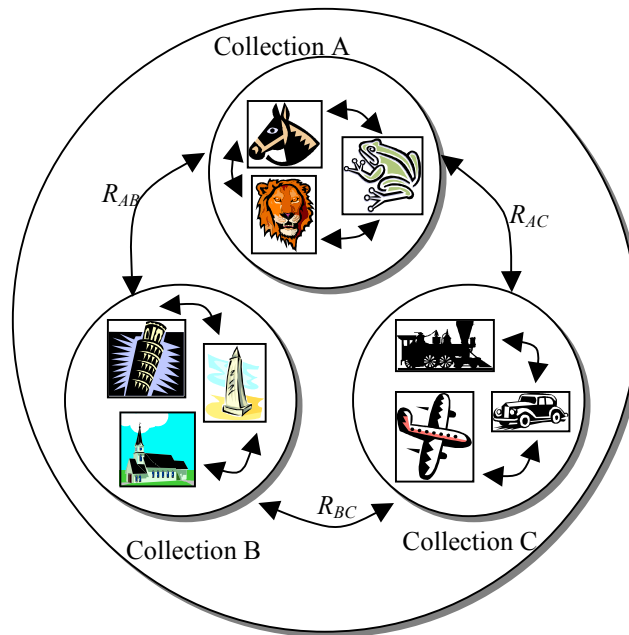
- MixedCollectionType → Describes a mixed collection of multimedia content, descriptors and semantic concepts related to multimedia material. The MixedCollection may contain a mix of different types of multimedia content and concepts within a single description. However, the MixedCollection shall be limited to a single type of descriptor per description.
- Content → Describes content that makes up the ContentCollection.
- ContentRef → Describes a reference to content that makes up the ContentCollection.
- Descriptor → Describes the instance of the multimedia content descriptor by giving the value of the Descriptor. The DescriptorCollection shall be limited to one type of Descriptor per instance.
- Concept → Describes a semantic concept belonging to the ConceptCollection.
- ConceptRef → Describes a reference to a semantic concept belonging to the ConceptCollection.

### **StructuredCollection DS**

The StructuredCollection DS describes the association relationships among Collections, CollectionModels and ClusterModels. An example of collection structure is depicted in Figure 5 .

The tools referring to this DS are:

- StructuredCollectionType → Describes the association relationships among Collections, CollectionModels and ClusterModels.
- Collection → Describes an unbounded set of collections.
- CollectionRef → Describes an unbounded set of references to collections.
- CollectionModel → Describes an unbounded set of collection models.
- CollectionModelRef → Describes an unbounded set of references to collection models.
- ClusterModel → Describes an unbounded set of cluster models.
- ClusterModelRef → Describes an unbounded set of references to cluster models.
- Graph → Describes a graph that gives an unbounded set of relations among the Collections, CollectionModels or ClusterModels.



**Figure 5.** The StructuredCollection DS describes collections in terms of the relationships (i.e.,  $R_{AB}$ ,  $R_{BC}$ ,  $R_{AC}$ ) among collections.

### 3.3.2 Models

This subclause specifies tools for describing models related to multimedia content. The models provide parameterized descriptions of collections or classes of multimedia content. The models can be expressed in terms of statistics or probabilities associated with the attributes of collections of multimedia content, or can be expressed through examples or exemplars of the multimedia content classes.

The tools belonging this subclause are:

- ModelType → Describes a model related to multimedia content.
- Confidence → Identifies the confidence of the Model. Confidence indicates the degree of confidence in the model parameters. For example, in the case of a ProbabilityModel, if the model parameter values are far away from their corresponding statistical mean or outside a corresponding confidence interval, the model shall be assigned a lower confidence score. Likewise if the model parameters values are considered to be unusual, then the confidence in the values may be low. Otherwise, if the values are usual or within an expected range, then the confidence may be high. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
- Reliability → Identifies the reliability of the Model. Reliability refers to the accuracy of the values of the model parameters. Reliability may take into account the method of extraction. For example, in some cases if model parameter values are extracted automatically, then a reliability score may be assigned that is lower than those

extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.

### 3.3.3 Probability models

The ProbabilityModels DS describes the characterization of multimedia content using probabilities and statistics. The ProbabilityModel DS forms the base type of the specialized probability models that characterize the attributes and features of multimedia content.

The tools defined in this part are:

- ProbabilityModelType → Describes the characterization of multimedia content using probabilities and statistics.
- ProbabilityDistributionType → Describes a probability distribution. This is characterized by the following tools: Mean, Variance, Min, Max, Mode, Median, Moment, Value (of the Moment around a point specified by Center), Order (of the Moment), Center (referring the Moment), Cumulant, Value (of the Cumulant around a point specified by Center), Order (of the Cumulant), Center (referring the Cumulant), Dim.
- DiscreteDistributionType → Describes a discrete probability distribution. DiscreteDistributionType extends ProbabilityDistributionType. In this DS the most important distributions, with the relative parameters, are defined: HistogramProbabilityType, BinomialDistributionType, HyperGeometricDistributionType, PoissonDistributionType, GeometricDistributionType, DiscreteUniformDistributionType.
- ContinuousDistributionType → Describes a continuous multi-dimensional probability density function. In this DS the most important distributions, with the relative parameters, are defined: GaussianDistributionType, GeneralizedGaussianDistributionType, ExponentialDistributionType, LognormalDistributionType, GammaDistributionType, ContinuousUniformDistributionType, GaussianMixtureModelType.
- FiniteStateModelType → Describes a finite state model. Also, other DSs are defined with the relative parameters: StateTransitionModelType, DiscreteHiddenMarkovModelType, ContinuousHiddenMarkovModelType.

### 3.3.4 Analytic models

Analytic models describe the association of labels or semantics with collections of multimedia content. The collections may be specified by enumerating the members of the collection, such as by using ContentCollections or DescriptorCollection, or may be specified using parameterized representations of the collections in the form of ProbabilityModels. This allows

the Analytic models to characterize different semantic concepts by models in terms of clusters of multimedia data, example collections of multimedia content descriptions, or probability models.

The tools defined in this subclause are:

- AnalyticModelType → Describes the association of labels or semantics with collections of multimedia content.
- Label → Describes a semantic label for the model given as a text description. Multiple labels may be supplied.
- Relevance → Identifies the relevance of the semantic label to the Analytic Model. Relevance is specified on a zero-to-one scale, where zero indicates no relevance and one indicates highest relevance.
- Confidence → Identifies the confidence of the semantics label. Confidence indicates the degree of confidence of the semantic label considering the likelihood of the label. For example, if a label occurs rarely, it may be assigned a low confidence. Whereas, if a label occurs commonly, it may be assigned a high confidence. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
- Reliability → Identifies the reliability of the semantics label. Reliability refers to the accuracy of the values of the semantic label, which may take into account the method of extraction. For example, in some cases the semantic label values that are produced automatically may be assigned lower reliability score than those that are extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.
- Semantics → Describes a semantics for the model given as a Semantics description. Multiple Semantics descriptions may be supplied.
- Relevance → Identifies the relevance of the Semantic entity to the Analytic Model. Relevance is specified on a zero-to-one scale, where zero indicates no relevance and one indicates highest relevance.
- Confidence → Identifies the confidence of the Semantic entity. Confidence indicates the degree of confidence of the Semantic entity considering the likelihood of the Semantic entity. For example, if a Semantic entity occurs rarely, it may be assigned a low confidence. Whereas, if a Semantic entity occurs commonly, it may be assigned a high confidence. A value of 1.0 indicates highest confidence. A value of 0.0 indicates lowest confidence.
- Reliability → Identifies the reliability of the Semantic entity. Reliability refers to the accuracy of the values of the Semantic entity, which may take into account the method of extraction. For example, in some cases the Semantic entity values that are produced automatically may be assigned lower reliability score than those that are extracted manually. A value of 1.0 indicates highest reliability. A value of 0.0 indicates lowest reliability.

- **Function** → Describes whether the Semantics or Label is being described by the model, or if the Semantics or Label is describing the model. For example, in the describing case, the Label="Sunsets" can be associated with a collection of images to describe or annotate the images. For example, in the described case, a collection of images can also be associated with the Label="Sunsets" to indicate that the images describe or annotate the concept represented by the Label. In this case, the images may be thought of as example images of sunsets.
- **ModelStateType** → Describes the state of a source of a finite state model.

In this subclause other tools are defined: CollectionModel DS, DescriptorModel DS, ProbabilityModelClass DS.

### 3.3.5 CollectionModel DS

The CollectionModel DS describes the association of semantics with a Collection. The CollectionModel allows the association of the same semantics with multiple Collections by allowing an unbounded set of Collections to be specified.

The tools referring to this DS are:

- **CollectionModelType** → Describes the association of semantics with a collection or multiple collections. The CollectionModel may contain a mix of different types of collections of multimedia content and concepts within a single description. However, the CollectionModel shall be limited to a single type of descriptor per description.
- **Collection** → Describes the Collections that are assigned the labels or semantics of the CollectionModel.
- **CollectionRef** → Specifies references to the Collections that are assigned the semantic labels of the CollectionModel.

### DescriptorModel DS

The DescriptorModel DS describes a template or model of a known descriptor by selecting elements from the descriptor. The model is formed by describing the mapping of elements of the descriptor type to the fields of the descriptor model. The DescriptorModel is used by ProbabilityModelClass and ClusterModel to allow the description of a class or cluster of descriptors using probability models or statistics computed over the aggregation of multiple descriptors.

The tools referring to this DS are:

- **DescriptorModelType** → Describes a template or model of a known descriptor by selecting elements from the descriptor. The model is formed by describing the mapping of elements of the descriptor to the fields of the descriptor model.

- **Descriptor** → Describes an example instance of the Descriptor in order to allow its elements to be modeled. The Descriptor example shall provide a valid description of the Descriptor including all required attributes and elements. Any number of the elements of the Descriptor can be used to form the descriptor model by specifying the mapping of elements of the descriptor to the fields of the descriptor model. Any elements of the example Descriptor that are not mapped to fields of the descriptor model are assumed to be constant in the descriptor model, taking the values specified in the example Descriptor. All attributes of the example Descriptor are also assumed to be constant in the descriptor model, taking the values specified in the example Descriptor.
- **Field** → Describes the sequence of elements from the example Descriptor that are being modeled. The fields of the model, which correspond to elements of the Descriptor, are described by specifying the xpath expression to the elements of the Descriptor. The xpath locator shall be used only to locate elements in the immediately preceding Descriptor example. Multiple fields can be described, in which case, the elements of the Descriptor are concatenated in the order given by the specification of the fields.

### **ProbabilityModelClass DS**

The ProbabilityModelClass DS describes the association of semantics with classes of descriptions of multimedia content, which are described using ProbabilityModels. The ProbabilityModels provide an aggregate statistical representation of the descriptions in a class. The ProbabilityModelClass allows the association of the same semantics with multiple ProbabilityModels by allowing an unbounded set of ProbabilityModels to be specified.

The tools referring to this DS are:

- ProbabilityModelClassType → Describes the association of semantics with probability models of multimedia content.
- **DescriptorModel** → Describes an example instance of the Descriptor in order to allow the elements of the Descriptor to be modeled using a ProbabilityModel. The DescriptorModel describes the sequence of fields that are concatenated to form a vector. The aggregation of instances of these vectors are then modeled using the ProbabilityModel.
- **ProbabilityModel** → Describes the ProbabilityModel of the aggregation of Descriptions of type of the DescriptorModel that form the class.

### **3.3.6 Cluster Models**

Cluster Models describe Analytic models and their associated Probability models. For example, this allows the description of the centroid of a collection of descriptor instances in order to represent a semantic class.

The tools defined in this subclause are:

- ClusterModelType → Describes the association of labels or semantics, and statistics or probabilities with collections.
- Collection → Describes the collection that provides the multimedia content collection, instance or class information.
- CollectionRef → Describes the reference to the associated collection that provides the multimedia content collection, instance or class information.
- ClusterModel → Describes an unbounded set of child ClusterModels of the current ClusterModel. This allows nesting of ClusterModels within ClusterModels to describe hierarchies. The specification of child ClusterModels is optional. If not specified, then the current ClusterModel terminates the ClusterModel hierarchy.
- ClusterModelRef → Describes an unbounded set of references to child ClusterModels of the current ClusterModel.
- DescriptorModel → Describes an example instance of the Descriptor in order to allow the elements of the Descriptor to be modeled using a ProbabilityModel. The DescriptorModel describes the sequence of fields that are concatenated to form a vector. The aggregation of instances of these vectors are then modeled using the ProbabilityModel.
- ProbabilityModel → Describes the associated ProbabilityModel that provides a parameterized description of the AnalyticModel.
- DescriptorName → Describes the name of the Descriptor or Description Scheme type as a QName that gives the name of the complexType or simpleType associated with the D or DS. The descriptorName is mandatory since it is needed to understand how to interpret the ProbabilityModel that characterizes the aggregation of description elements in the class.

### 3.3.7 Classification Models

Classification Models describe information about known multimedia content that can be used to classify unknown multimedia content. The output of the classifiers is the association of labels or semantics with the unknown content. The information the classifier uses involves the association of labels or semantics with the known content.

The tools referring to this DS are:

- ClassificationModelType → Describes information about known collections of multimedia content in terms of labels, semantics and models that can be used to classify unknown multimedia content.
- Complete → Indicates whether the set of AnalyticModels that comprises ClassificationModel is sufficient for completely covering the semantic space of the unknown data. For example, given a ClassificationModel formed from a model of

"Sunsets" and a model of "Nature Scenes", the ClassificationModel completely covers the semantics of photographs of sunsets and nature scenes, but does not completely cover the semantics of photographs of soccer games. By default, complete has a value of "true".

- Redundant → Indicates whether the set of AnalyticModels that comprises ClassificationModel overlap in the semantic space. For example, given a ClassificationModel formed from a model of "Sunsets" and a model of "Sunsets over Water", the ClassificationModel redundantly covers the semantics of photographs of sunsets. By default, complete has a value of "false".

In this subclause other tools are defined: ClusterClassificationModel DS and ProbabilityClassificationModel DS.

### **ClusterClassificationModel DS**

The ClusterClassificationModel DS describes a classifier by specifying a set of ClusterModels that characterize the semantics of the multimedia content. The ClusterModel describes the association of semantics with Clusters of multimedia content.

The tools referring to this DS are:

- ClusterClassificationModelType → Describes a classifier by specifying a set of models that characterize the semantics of the multimedia content.
- ClusterModel → Describes the set of ClusterModels. Each ClusterModel describes the association of semantics with Clusters of multimedia content.
- ClusterModelRef → Describes the set of ClusterModels in terms of references to ClusterModel descriptions. Each ClusterModel describes the association of semantics with Clusters of multimedia content.

### **ProbabilityClassificationModel DS**

The ProbabilityClassificationModel DS describes a classifier for unknown multimedia content by specifying a set of ProbabilityModels that characterize the semantics of the descriptions of multimedia content. Each ProbabilityModel describes the association of semantics with ProbabilityModels of multimedia content descriptions.

The tools referring to this DS are:

- ProbabilityClassificationModelType → Describes a classifier by specifying a set of ProbabilityModels that characterize the semantics of the descriptions of multimedia content.
- ProbabilityModelClass → Describes the set of ProbabilityModelClasses. Each ProbabilityModelClass describes the association of semantics with ProbabilityModels of multimedia content descriptions.

## 4 MPEG-21 tools

Today, many elements exist to build an infrastructure for the delivery and consumption of multimedia content. There is, however, no 'big picture' to describe how these elements, either in existence or under development, relate to each other.

The aim for MPEG-21 is to describe how these various elements fit together, in fact, the vision for MPEG-21 is to define a multimedia framework to enable “*transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities*”.

The general title of MPEG-21 is: *Information Technology – Multimedia framework*.

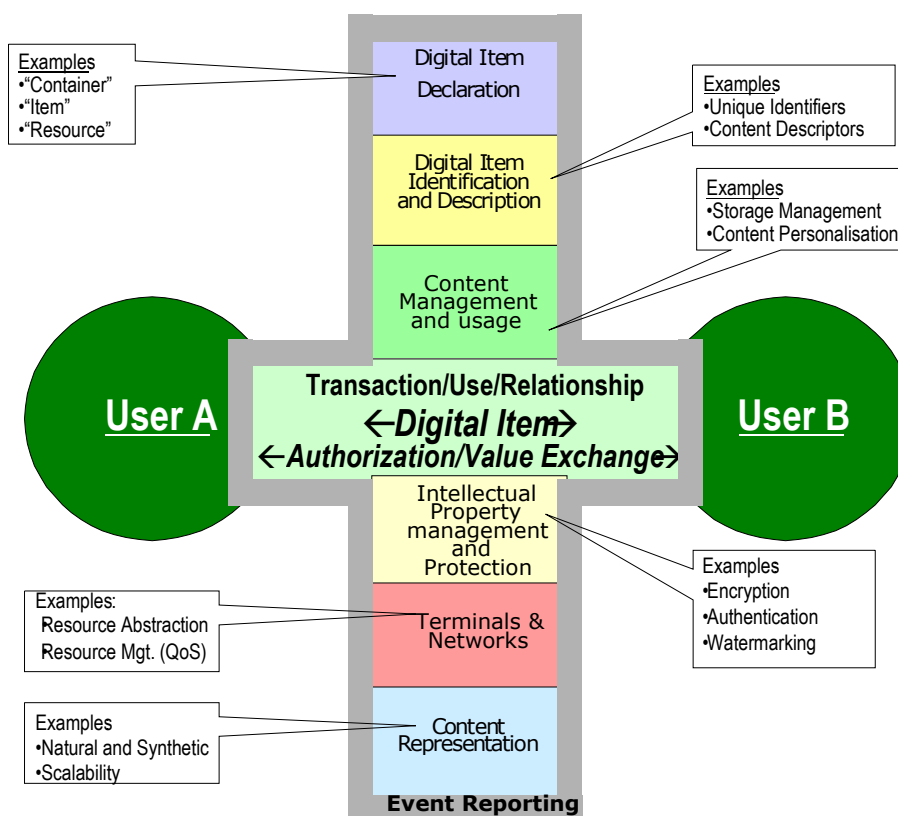
### 4.1 User model

A *User* is any entity which interacts in the MPEG-21 environment or makes use of a Digital Item.

A *Digital Item* (DI) is **a structured digital object with a standard representation, identification and metadata within the MPEG-21 framework; this entity is also the fundamental unit of distribution and transaction within this framework.**

As shown in Figure 6, users interact using the keys elements defined by MPEG-21:

- *DI Declaration*: a uniform and flexible abstraction and interoperable schema for declaring DI;
- *DI Identification and Description*: a framework for identification and description of any entity regardless of its nature, type or granularity;
- *Content Handling and Usage*: provide interfaces and protocols that enable creation, manipulation, search, access, storage, delivery, and (re)use of content across the content distribution and consumption value chain;
- *Intellectual Property Management and Protection*: the means to enable content to be persistently and reliably managed and protected across a wide range of networks and devices;
- *Terminals and Networks*: the ability to provide interoperable and transparent access to content across networks and terminals;
- *Content Representation*: how the media resources are represented;
- *Event Reporting*: the metrics and interfaces that enable Users to understand precisely the performance of all reportable events within the framework.



**Figure 6.** Event reporting, by creating and interfaces, further describes specific interactions.

## 4.2 MPEG-21 Parts

MPEG-21 consists of seven parts, as described in the following paragraphs.

### 4.2.1 Part 1: Vision, Technology and Strategy

As mentioned in the previous section, the ‘vision’ for MPEG-21 is to define a multimedia framework to enable transparent and augmented use of multimedia resources across a wide range of networks and devices used by different communities.

The ‘technology’ is needed to facilitate the creation, management, transport, manipulation, distribution, and consumption of digital items.

The 'strategy' is defined for achieving a multimedia framework by the development of specifications and standards based on well-defined functional requirements through collaboration with other bodies.

### 4.2.2 Part 2: Digital Item Declaration

Within MPEG-21, which proposes to facilitate a wide range of actions involving DI, there is a need for a very precise description for defining exactly what constitutes such an “item”. Clearly there are many kinds of content, and probably just as many possible ways of

describing it to reflect its context of use. This presents a strong challenge to lay out a powerful and flexible model for DI which can accommodate the myriad of forms that content can take. In this sense, the purpose of the *Digital Item Declaration* (DID) specification is to describe a set of abstract terms and concepts to form a useful model for defining DI.

The DID is described in three sections:

1. *model*: definition of abstract terms and concepts to form a useful model for defining DI;
2. *representation*: normative description of the syntax and semantics of each of the Digital Item Declaration elements, as represented in XML;
3. *schema*: normative XML schema comprising the entire grammar of the Digital Item Declaration representation in XML.

#### **4.2.3 Part 3: Digital Item Identification**

The scope of *Digital Item Identification* (DII) is to uniquely identify the DIs, the IP of DIs, the DSs and to use identifiers to link DIs with related information

#### **4.2.4 Part 4: Intellectual Property Management Tool Representation and Communication System**

This part of MPEG-21 defines an interoperable framework for *Intellectual Property Management and Protection* (IPMP).

#### **4.2.5 Part 5: Right Expression Language**

A *Rights Expression Language* (REL) is seen as a machine-readable language that can declare rights and permissions using the terms as defined in the Rights Data Dictionary.

The REL is intended to provide flexible, interoperable mechanisms to support transparent and augmented use of digital resources. Also, the REL is intended to support specification of access and use controls for digital content in cases where financial exchange is not part of the terms of use, and to support exchange of sensitive or private digital content. Finally, the REL is intended to provide a flexible interoperable mechanism to ensure personal data is processed in accordance with individual rights and to meet the requirement for users to be able to express their rights and interests in a way that addresses issues of privacy and use of personal data.

#### **4.2.6 Part 6: Right Data Dictionary**

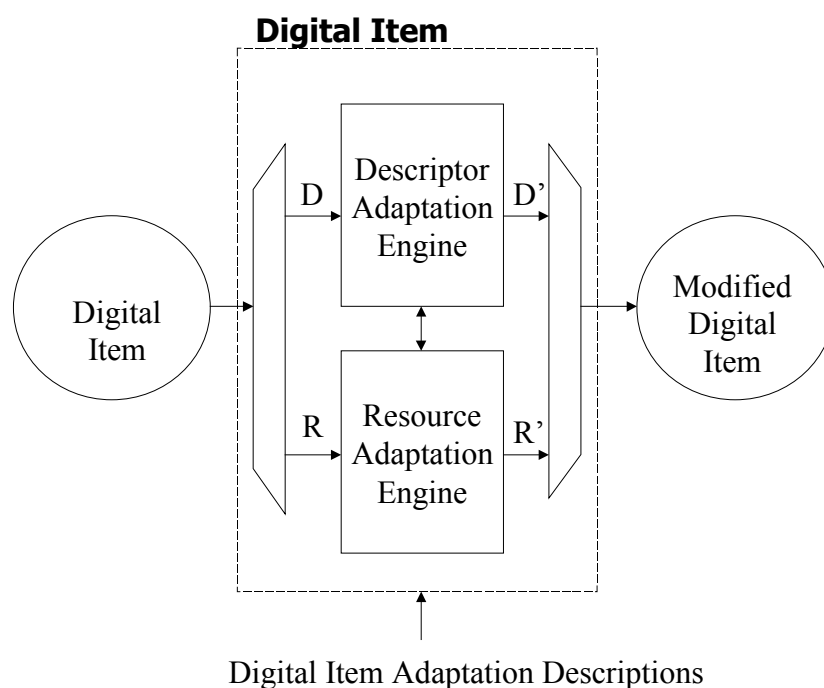
The scope of *Rights Data Dictionary* (RDD) is to provide a set of clear, consistent, structured and integrated definitions of terms for use in REL.

Terms in RDD are categorized as *Standardised*, *Native*, *Adopted*, *Mapped* and *Isolated*.

#### **4.2.7 Part 7: Digital Item Adaptation**

The goal of the Terminals and Networks key element is to achieve interoperable transparent access to (distributed) advanced multimedia content by shielding users from network and

terminal installation, management and implementation issues. This will enable the provision of network and terminal resources on demand to form user communities where multimedia content can be created and shared, always with the agreed/contracted quality, reliability and flexibility, allowing the multimedia applications to connect diverse sets of users, such that the quality of the user experience will be guaranteed. In this sense, a *Digital Item Adaptation* (DIA) is required. The conceptual architecture of DIA is shown in Figure 7.



**Figure 7.** Illustration of Digital Item Adaptation.

The specific items targeted for standardization are outlined below.

- *User Characteristics*: description tools that specify the characteristics of a user, including preferences to particular media resources, preferences regarding the presentation of media resources, and the mobility characteristics of a user. Additionally, description tools to support the accessibility of DI to various users, including those with audio-visual impairments, are being considered.
- *Terminal Capabilities*: description tools that specify the capability of terminals, including media resource encoding and decoding capability, hardware, software and system-related specifications, as well as communication protocols that are supported by the terminal.
- *Network Characteristics*: description tools that specify the capabilities and conditions of a network, including bandwidth utilization, delay and error characteristics.
- *Natural Environment Characteristics*: description tools that specify the location and time of a user in a given environment, as well as audio-visual characteristics of the natural environment, which may include auditory noise levels and illumination properties.

- *Resource Adaptability*: tools to assist with the adaptation of resources including the adaptation of binary resources in a generic way and metadata adaptation. Additionally, tools that assist in making resource-complexity trade-offs and making associations between descriptions and resource characteristics for Quality of Service are targeted.
- *Session Mobility*: tools that specify how to transfer the state of DI from one User to another. Specifically, the capture, transfer and reconstruction of state information will be specified.

### 4.3 Digital Item Adaptation (Part 7)

In this part of MPEG-21, the clause ‘*usage environment description tools*’ is defined. In turn, in the ‘*usage environment descriptions tools*’, there is a sub-clause called *user characteristics*.

#### 4.3.1 User characteristics

This subclause specifies tools for describing the user characteristics. Here, the following tools are defined:

- UserCharacteristicsType → tool for describing characteristics of a user.
- Reference → references an external description of user characteristics; either this reference is used as a description, or the user characteristics are specified inline according to the specified elements.
- User → describes general characteristics of a user such as name and contact information; a user can be a person, a group of persons, or an organization.
- ContentPreferences → describes preferences of an end user related to the type and content of DI.
- PresentationPreferences → describes preferences of an end user related to the presentation of DI.
- Accessibility → describes accessibility-related characteristics of a user.
- Mobility → describes mobility-related characteristics of a user.

The last four tools (ContentPreferences, PresentationPreferences, Accessibility, Mobility) are described in details subsequently.

#### 4.3.2 Content preferences

- ContentPreferencesType → describes preferences of an end user related to the type and content of DI.

- UserPreferences → describes preferences of an end user related to the type and content of DI; the syntax and semantics of UserPreferencesType is specified in ISO/IEC 15938-5 (MPEG-7).
- UsageHistory → describes history of actions on DI by an end user; the syntax and semantics of UsageHistoryType is specified in ISO/IEC 15938-5 (MPEG-7).

#### 4.3.3 PresentationPreferences

- PresentationPreferencesType → tools that describes the audio-visual presentation preferences of a user.
- Audio → describes the audio presentation preferences of a user. The users preference are described through the following tools:
  - AudioPower (loudness of audio),
  - Mute (the possibility of mute the audio),
  - FrequencyEqualizer (specific equalizing scheme in terms of frequency ranges and attenuation values),
  - Period (attribute of FrequencyEqualizer: the lower and the upper corner frequency for an equalization),
  - Level (attribute of FrequencyEqualizer: the attenuation or amplification of a frequency),
  - PresentEqualizer (specific equalizing scheme in terms of a verbal description for a equalizer preset),
  - AudibleFrequencyRange (for a specific frequency range, the lower and the upper corner frequency),
  - AudibleLevelRange (for a specific level range, the lower and the upper level value).
- Display → describes the display presentation preferences of a user.

#### 4.3.4 AccessibilityCharacteristics

- AccessibilityCharacteristicsType → tools that describes accessibility-related characteristics of a user.
- Audio → describes auditory impairments of a user. The users preference are described through the following tools:
  - AuditoryImpairmentType (impairment of a user's auditory system for the left and the right ear).
  - Audiogram (value for the derivation from the threshold in quite measured in decibel for  $N$  Hz for one ear).
  - FreqNHz (attribute of audiogram: threshold at  $N$  Hz).

- Visual → describes visual impairments of a user. The users preference are described through the following tools:
- - VisualImpairmentType (visual impairments of a user).
  - ColorVisionDeficiency (color vision deficiency of a user).  
This description tool is characterized, in turn, by other tools: ColorVisionDeficiencyType, DeficiencyType, Red-Deficiency, Green-Deficiency, Blue-Deficiency, CompleteColor-Blindness, DeficiencyDegree, TextualDegree, NumericDegree, RightSight, LeftSight, Illuminance.
- HasColorVisionDeficiency (attribute of color vision deficiency: indicator whether the user has color vision deficiency or not)

#### 4.3.5 MobilityCharacteristics

This part is still in its definition phase.

## 5 SPATION's user Interface and MPEG-7/21 tools

In this section the MPEG tools previously described will be revised with respect to the SPATION scenario constraints. More specifically the following aspects will be analyzed:

1. To identify which standard tools can support a given user functionality.
2. To identify which tools are useful to deliver content and metadata across the multimedia home network.
3. To identify which tools are needed to support the two previous operations (terminal and network capabilities, adaptation, etc.)

### 5.1 User Interface functionalities support

In SPATION Deliverable 5 we have defined the functionalities that will be accesible through the user interface. These functions will be now analyzed looking at the main information needed to support their implementation. According to the these requirements, a standard MPEG tool will be selected and its use will be discussed taking into account the SPATION constrains and objectives.

#### 5.1.1 Device browser

The device browser will be used to control devices physically available in different rooms of the home. In principle, this application will need to know the information regarding functionalities supported by each device and how a given function can be requested or pushed.

An elegant way for storing and distributing this information can be obtained by using a Digital Item. This general tool can carry both content and metadata, and in this case it would be possible to store the device information provided by the UPnP middleware plus the picture

of the considered device. Moreover a similar approach can be used to describe a single room of the house and the overall house.

To give an example of how this can be implemented, we can consider the picture represented in Figure 8 of the living room, where there is a mini sound system. The image can be segmented in order to isolate the main elements of the system. Then, this decomposition and the relationship between elements can be described by using the StillRegion DS, part of MPEG-7.

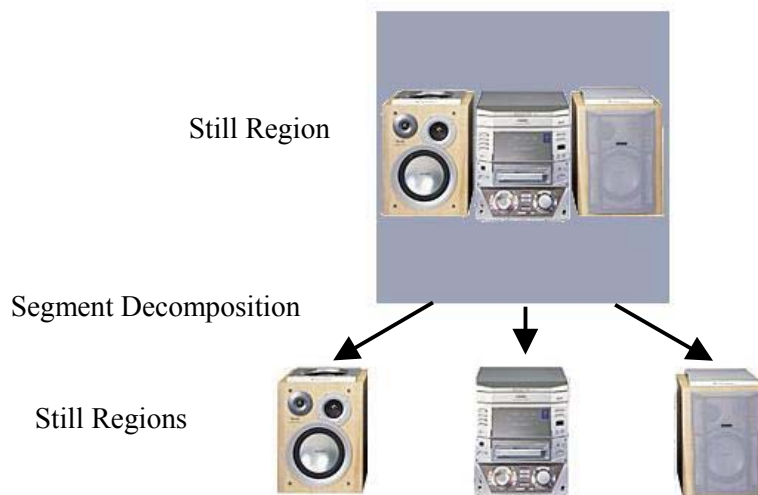


Figure 8: Image decomposition information.

```

<StillRegion id="MiniHiFi">
  <TextAnnotation>
    <FreeTextAnnotation> FW R88 Mini HiFi system with: Radio, Deck
      and CD changer.
    </FreeTextAnnotation>
  </TextAnnotation>
  <ContourShape > .. </ContourShape>
  <SpatialDecomposition gap="true" overlap="false">
    <StillRegion id="LeftLoudspeaker">
      <ContourShape> .. </ContourShape>
    </StillRegion>
    <StillRegion id="MainUnit">
      <ContourShape> ... </ContourShape>
    </StillRegion>
    <StillRegion id="RightLoudspeaker ">
      <ContourShape> ... </ContourShape>
    </StillRegion>
  </SpatialDecomposition>
</StillRegion>

```

### 5.1.2 Video, Music, and Photo browser

In the following sections we will list the DSs that can be used to support the SPATION content browsers. In the video browser we identify the following functions: Filter, Browse, Summary, Trailer, Device, and Box. In the Music and Photo browsers additionally albums for photographs and music collections are available.

- *Filter – provides functions to user to reduce the amount of returned results in a search*
  - MPEG-7, MDS, User Interaction, User Preferences
    - ✓ FilteringAndSearchPreferences DS
    - ✓ CreationPreferences DS → Title, Creator, Keyword, Location, DatePeriod
    - ✓ ClassificationPreferences DS → Country, DatePeriod, Language, Genre, Subject, Mediareview, ParentalGuidance
    - ✓ PreferenceCondition DS → PublicationSource, PublicationPlace, PublicationDate, Publisher, MediaFormat, noRepeat, noEncryption, noPayPerUse
  - MPEG-7, MDS, Content Management, Creation and Production tools
    - ✓ Creation DS → Title, TitleMedia (TitleVisual), Abstract, Creator (Role, Agent, AgentRef, Character, Instrument), CreationCoordinates, CreationLocation, CreationDate, CreationMaterial
    - ✓ Classification DS → Form, Genre, Subject, Purpose, Language, SubtitleLanguage, ClosedCaptionLanguage, SignLanguage, primary (referring to language), translation (referring to language), ExtendedLanguage, Release, Country, Date, Target, Market, Age, Country, ParentalGuidance (ParentalRating, MinimumAge, Country), MediaReview (Rating, FreeTextReview, ReviewReference, Review)
    - ✓ RelatedMaterial → PublicationType, MaterialType, MediaLocator, MediaInformation, MediaInformationRef, CreationInformation, CreationInformationRef, UsageInformation, UsageInformationRef
  - MPEG-7, MDS, Content Management, Usage Description tools
    - ✓ UsageInformation DS → Rights (RightsId), FinancialResults (AccountItem, EffectiveDate, CostType, IncomeType, Currency of the price, Value of the price), Availability (InstanceRef, PublicationType, OriginPlace, Distributor, Financial, Rights, AvailabilityPeriod), UsageRecords (AvailabilityRef, Audience, Financial)
- *Browse – browsing video content*
  - MPEG-7, MDS, User Interaction, User Preferences
    - ✓ BrowsingPreferences DS
    - ✓ PreferenceCondition DS
- *Summary – Provides a still picture overview of the content*
  - MPEG-7, MDS, User Interaction, User Preferences

- ✓ SummaryPreferences DS → NumOfKeyFrame, MinNumOfKeyFrame, MaxNumOfKeyFrame
- ✓ PreferredSummaryTheme DS
- *Trailers* – a trailer constructed using a playlist of segments of the original content
- *Device* - allows browsing devices that are detected in the network
- *Box*- provides a virtual grouping of arbitrary content
- MPEG-7, MDS, Content Organization, Collections
  - ✓ Collection DS → CreationInformation, UsageInformation, TextAnnotation, Summarization, Collection, CollectionRef
  - ✓ ContentCollection DS → VisualFeature, Content, ContentRef
  - ✓ SegmentCollection DS → Segment, SegmentRef
  - ✓ DescriptorCollection DS → Descriptor
  - ✓ ConceptCollection DS → Concept, ConceptRef
  - ✓ MixedCollection DS → Content, ContentRef, Descriptor, Concept, ConceptRef
  - ✓ StructuredCollection DS → Collection, CollectionRef, CollectionModel, CollectionModelRef, ClusterModel, ClusterModelRef, Graph
- MPEG-7, MDS, Content Organization, Models
  - ✓ ProbabilityModel DS → DiscreteDistribution, ContinuousDistribution, FiniteStateModel
  - ✓ AnalyticModel DS → CollectionModel, ProbabilityModelClass
  - ✓ ClusterModel DS
  - ✓ ClassificationModel DS → ClusterClassificationModel, ProbabilityClassificationModel
- *Album* – collection of images or songs
- MPEG-7, MDS, Content Organization, Collections
  - ✓ Collection DS → CreationInformation, UsageInformation, TextAnnotation, Summarization, Collection, CollectionRef
  - ✓ ContentCollection DS → VisualFeature, Content, ContentRef
  - ✓ SegmentCollection DS → Segment, SegmentRef
  - ✓ DescriptorCollection DS → Descriptor
  - ✓ ConceptCollection DS → Concept, ConceptRef
  - ✓ MixedCollection DS → Content, ContentRef, Descriptor, Concept, ConceptRef
  - ✓ StructuredCollection DS → Collection, CollectionRef, CollectionModel, CollectionModelRef, ClusterModel, ClusterModelRef, Graph

## 5.2 The UMA project

The Universal Multimedia Access (UMA) project aims to define solutions for the delivery of multimedia content such as images, video, audio and electronic format document, through a

network in different conditions. In the UMA philosophy the information transfer has to be realized taking into account both user and content provider preferences and terminals and network capabilities. A major motivation behind UMA is to enable terminals with limited communication, processing, storage and display capabilities to access rich multimedia content.

UMA presents a solution for wired and wireless clients to access the same content server, each receiving content targeted at their client's capabilities.

The UMA concept is visualized in Figure 9. It is currently proposed as an application for MPEG-7 and MPEG-21 [4]. The UMA application suits the next generation mobile and wireless systems, as seen in the developments of third generation systems such as the European Universal Mobile Telecommunications System (UMTS) and the efforts of the Third Generation Project Partnership (3GPP). For these applications, UMA will enable users access to future services independently from their choice of access technology, terminal equipment and usage preferences.



Figure 9: The UMA concept.

Universal Multimedia Access can be provided in two basic ways:

- **“Info-pyramid”**: by storing, managing, selecting, and delivering different versions of the media objects (images, video, audio, graphics and text).
- **“On-the-fly”**: by dynamically manipulating the media objects, such as by using methods for text-to-speech translation, image and video transcoding, media conversion and summarization.

This allows the multimedia content delivery to adapt to the wide diversity of client device capabilities in communication, processing, storage, and display. Both the above methods for providing Universal Multimedia Access are demonstrated in some practical situations at the UMA home page [3].

In MPEG-7, a number of specific Description Schemes (DSs) for describing multimedia content have been proposed that are designed for managing different variations of multimedia material and describe their different resource requirements.

The open problems related to the UMA concept are mainly:

- storage modality that can be distributed or centralized;
- network infrastructures and protocols, transcoding is required in the case of a heterogeneous system.

### 5.2.1 UMA demos

As UMA is a test bed to evaluate MPEG-7 and MPEG-21 tools, a number of applications have been proposed and tested.

Descriptions of some of the current UMA demos available on the website will be discussed.

## 6 Content and Metadata Transcoding

The purpose of this section is to describe how content and metadata transcoding can be realized, and which are the meta-information that can be useful to support this process. For both problems, solutions in terms of MPEG tools will be provided in order to support the resource adaptation processes.

### 6.1 Content Transcoding

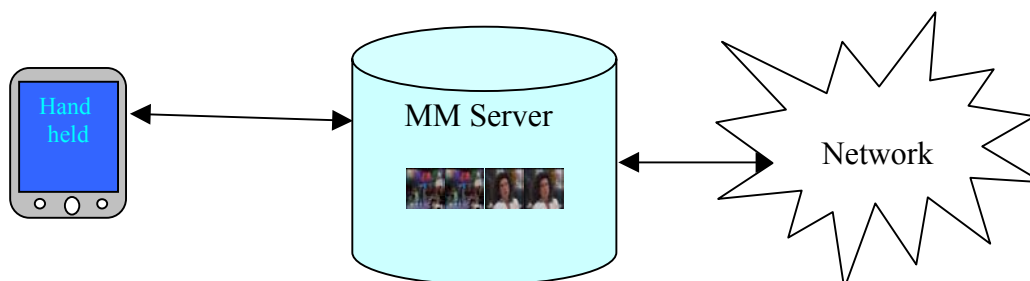
In the SPATION home network devices with different capabilities are connected. These capabilities affect the preferred format for content. Images and video can be scaled to fit a certain display size.. The amount of processing power that is available can also affect the coding format as some formats are more processing intensive than others. If a device has hardware support for decoding a certain format, but is not capable of decoding other formats a more powerful device in the network can transform the content to the required format. To perform these format conversions in a way that is transparent to the user the capabilities of devices and services in the network need to be described.

### 6.2 Metadata Adaptation

Besides adapting the format of the content meta-data adaption is needed. In this section we will provide an overview of the metadata adaptation process and provide a specification for the implementation of a metadata adaptation engine. A use case will be considered to better understand the problem we are trying to solve. While scaling and filtering of metadata have already been considered in other works (see document [14]) the integration of metadata is a quite new approach to the content description adaptation.

### 6.2.1 A use case

In Figure 10, a simple but sufficiently complex system, that will be used as an example, is depicted. As can be seen, the multimedia network is composed of a handheld and a Personal Computer communicating by means of a wireless connection.



**Figure 10.** A simple MultiMedia sub-network composed of a handheld and a content/descriptions server.

Assuming the PC is hosting a multimedia content/descriptions repository, we want to access this information from the handheld. More specifically, the objective is to understand the content of a specific document, like a video sequence, using the associated MPEG-7 description.

As originally proposed in the Universal Multimedia Access project, the access to any multimedia resource from any terminal and network has to be easy and transparent to the User. In the use case described here, we focus the attention on the word “easy”. This means that, due to the limited visualization capabilities of the handheld, some adaptation of the content/descriptions has to be performed.

If, for example, the user wants to explore the content by using the k-frames associated to the shots of the sequences, at least we need to adapt the k-frame size and color palette in order to fit the display characteristics. However, in general, more than one kind of description can be available for a given sequence, for example, obtained by using different extraction methods to create a different temporal segmentation of the AV document. In this case, the user has to select which part of the description he/she wants to use. However the user may not have enough information to make an optimal selection or he/she does not want to choose.

Consequently, one may require to adapt the various descriptions in order to obtain a unique one, generally, richer than the initial ones and lighter than the description obtained by simple aggregation. For example, if several k-frame series, relative to a specific video, are present in the network a unique description can be generated by using for example the technique described in [13].

Assuming now that we are referring to the above mentioned MPEG-7 descriptions by means of Digital Items it would be useful to have some mechanisms to support their integration.

### 6.2.2 Experiment description

Considering the above mentioned scenario, we will now define a specific case where integration can be useful.

Our user would like to watch the goals and the replay of goals of the soccer game Spain-Sweden. He/She is lying down on his/her bed and he/she does not want to go downstairs and use the PC to browse the video. However, he/she is holding a handheld connected to the

domestic network that, through a gateway, can provide an access to Internet. So he/she decides to do a query using a search engine in order to retrieve the MPEG-7 description of the soccer game, in which he/she is interested. After a query across the multimedia network, four MPEG-7 descriptions of the considered soccer game are retrieved.

In this situation, four descriptions should be downloaded and browsed by the user who, in principle, has no a priori knowledge of the overall metadata content.

As mentioned in the previous section, before sending the retrieved information to the user, it can be useful to process it in order to achieve the following objective:

- Reduce the metadata memory occupancy in order to better fit network and user terminal characteristics.
- Reduce, whenever it is possible, the metadata information redundancy.

This operation is relevant especially in the considered case where the user terminal has reduced capabilities.

In the following simulated experiment, it will be shown how these objectives can be supported by using the AdaptationHint DS as metadata adaptation tool.

### Description of Metadata

A video sequence, concerning the soccer game Spain-Sweden from the MPEG-7 Content Set (CD#18), has been used in the experiments. The sequence has been temporally segmented and a VideoSegment DS instance has been generated on the basis of the decomposition information.

For each shot, camera motion (pan) and audio volume values have been extracted in order to help the instantiation of three descriptions regarding salient event such as “goal” and “goal replay”.

The events have been detected using three different methods:

1. by manual extraction of the “goal” and “goal replay”;
2. using a detection algorithm based on audio loudness: the first ten shots characterized by the higher audio loudness are labeled as “goals” (Ordering Key concept [11]);
3. using an algorithm based on audio loudness and pan (camera motion): the first ten shots characterized by the higher loudness and pan are labeled as “goals”.

In the simulation three Event DSs instances have been considered, labeled respectively E1, E2, E3. While the segment decomposition information is common to all three descriptions, the positions where events have been detected are different depending on the used algorithm.

The resulting events are:

- E1 (description n° 1) detects: “goal” at shots #10, #38 and “replay of a goal” #16, #44, #45, #46, #53.
- E2: “goal” at shots #1, #2, #9, #10, #12, #16, #25, #32, #39, #43.
- E3: “goal” at shots #14, #31, #35, #39, #43, #54, #58, #61, #62, #68.

All the descriptions are composed by a list of events as shown in Example 1.

An event has been associated to each shot where an id equal to: “goal#N” (N is the shot number) has been used to identify a goal, “replay#N” a replay of a goal and “not-classified#N” for a shot with a content not classified.

```
..
<Event id="goal#02">
  <Label>
    <Name> Goal n.2 </Name>
  </Label>
  <SemanticTime id="shot#02">
    <Time>
      <TimePoint>T00:07:48</TimePoint>
      <Duration>PT0M9S</Duration>
    </Time>
  </SemanticTime>
</Event>

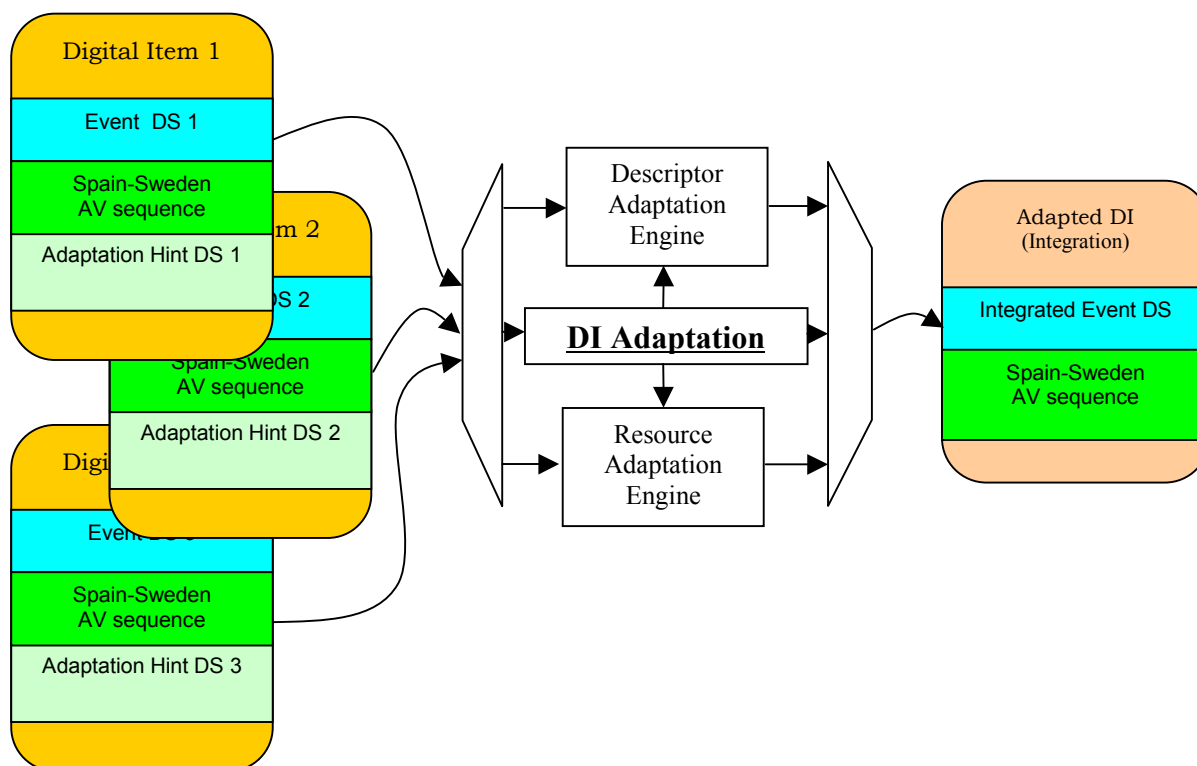
<Event id="not-classified#03">
  <SemanticTime id="shot#03">
    <Time>
      <TimePoint>T00:07:57</TimePoint>
      <Duration>PT0M15S</Duration>
    </Time>
  </SemanticTime>
</Event>
..
```

**Example 1.** Structure of E1, E2 and E3.

### Metadata Integration

The integration of E1, E2 and E3 has been performed in two different ways considering the “intersection” and the “union” of the events detected.

In both cases, two kinds of information have been used: the “Event id” and “Semantic Time id”.



**Figure 11.** Metadata scaling and Integration by using AdaptationHint DS and EventIntegration DS.

### Event Intersection

In this modality, the events are integrated separately according to the Event id “goal” or “replay”. The following rules have been set for the integration:

- In order not to lose information, for a given description, an event with a specific id is compared to elements of the same type in all other descriptions. This means that if an event with id=“my\_id..” is present only in one description nothing happens and the output corresponds to the list of events with id=“my\_id..” coming from the description considered initially.
- Events of the same type are compared considering a temporal window of three shots.

Applying the above rules we have obtained the following results:

- “goal” at shot #39.
- “goal replay” at shot #16, #44, #45, #46 and #53.

### Union method

The union method considers the common and non common events (“goal” and “goal replay”) given by the three algorithms.

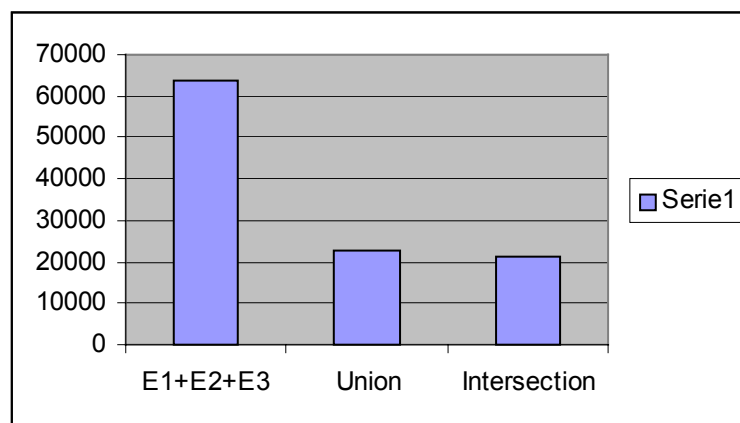
- “goal” at shot #1, #2, #9, #10, #12, #14, #16, #25, #31, #32, #35, #38, #39, #43, #54, #58, #61, #62 and #68
- “goal replay” at shot #16, #44, #45, #46 and #53

It has to be mentioned that this is one of the possible choices about integration methods that can be implemented, and more studies have to be conducted using this approach [11].

### Results evaluation

In Table 3, the comparison between the size of descriptions before and after the integration process is presented. It can be noticed how integration can simultaneously reduce the number of description instances (consequently the number of DI) and the total memory occupation.

Initial size (E1+E2+E3)	Integration by Union	Integration by Intersection
63819 B	22604 B	21200 B



**Table 3.** Memory occupation comparison(byte): before and after integration.

From the user side, the most evident benefit is the possibility to browse only one segment decomposition instead of three, with an immediate access to all the information he/she is interested in.

```
<AdaptationHint instanceSchema=.....>
  <instanceFileSize> 19920 </instanceFileSize>
  <totalNumOfElements> 78 </totalNumOfElements>
  <Component name="EventDS" number=7>
    <Location type="listOfID"> goal#1, goal#2, replay#1a, replay#2a,
      replay#2b, replay#2c, replay#2d
    </Location>
  </Component>
</AdaptationHint>
```

**Example 2.** AdaptationHint DS for Event DS 1 (E1).

```
<AdaptationHint instanceSchema=.....>
  <instanceFileSize> XXXX </instanceFileSize>
  <totalNumOfElements> 78 </totalNumOfElements>
  <Component name="EventDS" number=10>
    <Location type="listOfID"> goal#1, goal#2, goal#9, goal#10, goal#12,
                                goal#16, goal#25, goal#32, goal#39, goal#43
    </Location>
  </Component>
</AdaptationHint>
```

**Example 3.** AdaptationHint DS for Event DS 2 (E2).

For what concerns the use of Adaptation Hint DS we can say that it is useful in order to speed up the integration process if the “intersection” modality is used. Looking at Adaptation Hint DSs associated to the descriptions seen in Example 2 and Example 3, it can be noticed that it is easy to derive that no “replay” events are in E2 and E3. This eliminates the need to perform the intersection step for this type of event.

Even if Adaptation Hint DS can be useful to speed up description integration more adaptation tools can be identified in order to support a metadata adaptation engine as explained in the following sections.

### 6.2.3 Extension of MPEG-7/21 Tools

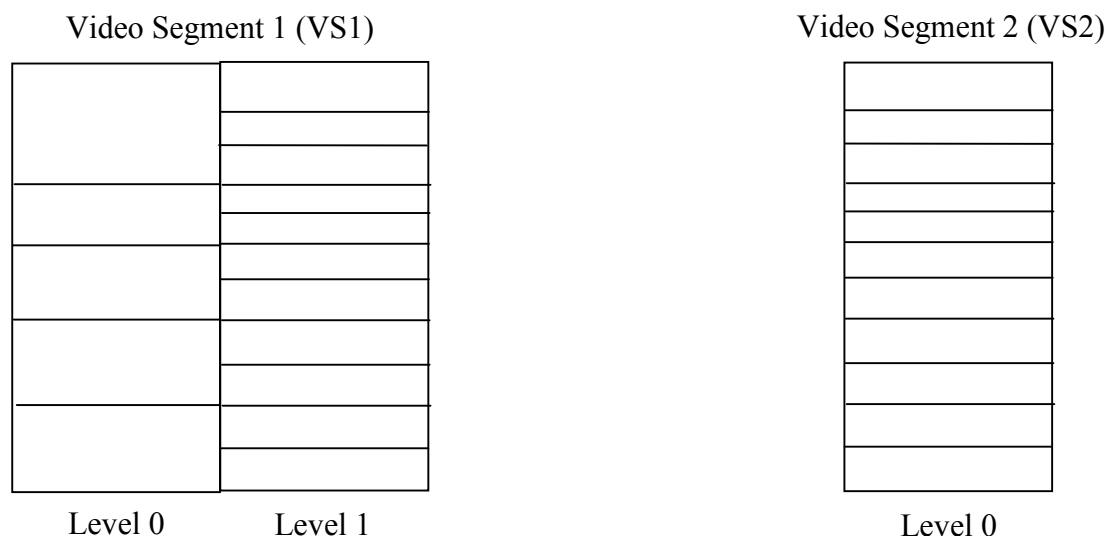
#### Integration Tools

One of the major problems related to the integration of metadata concerns the elementary unit that has to be considered during the process. In the experiment described above, we considered the integration of entities with the same granularity (an event associated to a shot).

What happens if, for example, we want to compare the two Video Segment instances, associated to the same video (described in Figure 12) ?

As it is shown, each of them has a different number of levels, and it is not trivial for an adaptation engine to automatically establish which are the elements that need to be compared during the integration process.

In this case, a way that would help in the identification of the level of granularity is to assign to each level a number representing the ratio between the number of segments over the total length of the video.



**Figure 12.** Segment comparison: segments of VS2 have to be compared with segment at level 1 of VS1.

### Metadata Adaptation Preferences

In the previous sections it has been shown how metadata integration can be supported by the use of Adaptation Hint DS. While audiovisual content transcoding is sometimes necessary to fit for example the size of the user display, metadata adaptation can also be an option. Considering for example a tree structured video segment decomposition, it is possible that a user does not want to get a scaled version, even if he has a display with limited display capabilities.

In this case, to scale the Video Segment according to a depth constraint can be a waste of adaptation resources (memory occupation and scaling time) while providing an unsatisfactory result to the User.

Assuming that the operations involving metadata adaptation such as scaling, integration, etc. are relevant in the above described context, adaptation tools have to be defined in order to establish if the metadata transcoding has to be performed or not according to the specific User preferences.

Here after, a preliminary definition of an Adaptation tool is proposed that is useful to determine User preferences about metadata adaptation. Whenever an adaptation engine can support functionalities such as metadata scaling, integration, etc., it can be established if this time consuming operations have to be performed or not.

```

<!-- #####-->
<!-- Definition of UserCharacteristics -->
<!-- #####-->
<complexType name="UserCharacteristicsType">
  <choice>
    <element ref="Reference"/>
  </choice>
</complexType>

```

```

    <sequence>
      ..
      <element name="MetaAdaptationPreferences"
        type="dia: ContentAdaptationPreferencesType"
        minOccurs="0" maxOccurs="1"/>
      ..
    </sequence>
  </choice>
  <attribute name="id" type="ID" use="optional"/>
</complexType>

```

### New UserCharacteristics semantics

Semantics of the UserCharacteristicsType:

<i>Name</i>	<i>Definition</i>
UserCharacteristicsType	Tool for describing characteristics of a User.
Reference	References an external description of User characteristics. Either this reference is used as a description, or the User characteristics are specified inline according to the specified elements.
User	Describes general characteristics of a User such as name and contact information. A User can be a person, a group of persons, or an organization.
ContentPreferences	Describes preferences of an End User related to the type and content of Digital Items.
PresentationPreferences	Describes preferences of an End User related to the presentation of Digital Items.
Accessibility	Describes accessibility-related characteristics of a User.
<b>MetaAdaptationPreferences</b>	<b>Describes Preferences of a User concerning metadata adaptation such as: scaling, integration ..</b>

#### 6.2.4 Metadata Adaptation Engine Specification

Coming soon.

## 7 Extraction Methods

In this section some of the recently developed techniques for metadata extraction and manipulation are presented.

A method for the detection of salient events in a soccer game has been developed and used in order to produce the MPEG-7 description part of the experiment on metadata integration.

The general methodology for the integration of metadata has been also included, providing a more detailed explanation for the integration engine used in the adaptation-integration experiments.

## 7.1 Detection of salient events in soccer games

To face the semantic indexing problem, man uses its cognitive skills, while an automatic system can face it by adopting a two-step procedure: in the first step, some low-level indices are extracted in order to represent low-level information in a compact way; in the second step, a decision-making algorithm is used to extract a semantic index from the low-level indices [16].

The problem of low-level descriptors extraction is widely discussed in literature [17], whereas only a few contributions address the decision-making issue [18]. Moreover, the solution to this issue seems to depend on the considered specific program category.

we have considered soccer video sequences. For this program category, the semantic content can be related to the occurrence of interesting events such as, for example, goals, shots to goal, and so on. These events can be found at the beginning or at the end of the game actions. Therefore a good semantic index of a soccer video sequence could be the list of all game actions, each one characterized by its beginning and ending event. Such a summary could be very useful to satisfy various types of semantic queries.

The problem of automatic detection of semantic events in sport games has been studied by many researchers. In general the objective is to identify certain spatio-temporal segments that correspond to semantically significant events.

In [19], for example, a method that tries to detect the complete set of semantic events which may happen in a soccer game is presented. This method uses the position information of the player and of the ball during the game as input, and therefore needs a quite complex and accurate tracking system to obtain this information.

In [20] and [21] we have studied the correlation between low-level descriptors and the semantic events in a soccer game. In particular, in [20], it is shown that the low-level descriptors are not sufficient, individually, to obtain satisfactory results (i.e., all the semantic events detected with only a few false detections).

In [21] we have therefore tried to exploit the temporal evolution of the low-level descriptors in correspondence with semantic events, by proposing an algorithm based on a finite-state machine. This algorithm gives good results in terms of accuracy in the detection of the relevant events, whereas the number of false detections remains still quite large.

In this work we present a semantic video indexing algorithm using controlled Markov chains to model the temporal evolution of low-level descriptors.

### The Low-level Descriptors

In this sub-section we describe the three low-level binary descriptors adopted in the proposed algorithm. These descriptors, associated to each P-frame, represent the following characteristics: (i) lack of motion, (ii) camera operations (pan and zoom parameters), and (iii) the presence of shot-cuts, and are the same descriptors used in [21]. Each descriptor takes value in the set  $\{0, 1\}$ .

The descriptor "Lack of motion" has been evaluated by thresholding the mean value of motion vector module for each P-frame. The threshold value has been set equal to 4. The descriptor assumes value 0 when the threshold is exceeded.

Camera motion parameters, represented by horizontal "pan" and "zoom" factors, have been evaluated using a least-mean squares method applied to P-frame motion fields [TM95]. We have then evaluated the value of the descriptor "Fast pan" ("Fast zoom") by thresholding the pan value (zoom factor), using the threshold value 20 (0.002). In this case, the descriptors assume value 1 when the threshold is exceeded.

Shot-cuts have been detected using only motion information as well. In particular, we have used the sharp variation of the above mentioned motion parameters, and of the number of Intra-Coded Macroblocks of P-frames [CBSV97] [TS97].

Specifically, to evaluate the sharp variation of the motion field we have used the difference between the average value of the motion vectors modules associated to two adjacent P-frames. This measure is given by:

$$\Delta\mu(k) = \mu(k) - \mu(k-1)$$

where  $\mu(k)$  is the average value of the motion vectors modules of P-frame  $k$ .

This parameter will assume significantly high values in presence of a shot-cut characterized by an abrupt change in the motion field between the two considered shots.

This information regarding the sharp change in the motion field has been suitably combined with the number of Intra-Coded MacroBlocks of the current P-frames, as follows:

$$\text{Cut}(k) = \text{Intra}(k) + \beta \Delta\mu(k),$$

where  $\text{Intra}(k)$  is the number of the Intra-Coded MacroBlocks of the current P-frame, and  $\beta$  is a weighting factor set to 20. When this parameter is greater than a prefixed threshold set to 700, we say that a shot-cut has occurred [21].

In the next sub-section, we describe the proposed algorithm where the temporal evolution of these low-level descriptors is modelled by a controlled Markov chain.

### **The proposed algorithm based on Controlled Markov Chain Model**

In this sub-section, we briefly describe the controlled Markov chain modelling framework [MDP94], and then detail the controlled Markov chain model adopted in our context.

The components of a controlled Markov chain model are the state and input variables, the initial state probability distribution, and the controlled transition probability function. Here, we consider homogeneous models with state and input variables taking values in finite sets. Denote by  $s(t)$  the random variable representing the state of the controlled Markov chain at time  $t \in T := \{0, 1, 2, \dots\}$ . At each  $t \in T$ , the state  $s(t)$  takes value in a discrete set  $S$ . At time  $t=0$ , the initial state  $s(0)$  is described in terms of its probability distribution, say  $P_0$ , over the space set  $S$ .

The evolution of  $s(t)$  from time  $t \in T$  to time  $t+1$  is governed by a probability of transition. This probability is affected by an input signal, that we denote by  $u(t)$ , taking value in a discrete input set  $U$ . The probability of transition is only a function of the input  $u \in U$  applied at time  $t$ . By this we mean that  $s(t+1)$  is a random variable conditionally independent of all other random variables at times smaller or equal to  $t$ , given  $s(t)$ ,  $u(t)$ .

Here we assume a stationary transition probability, i.e.,

$$P(s(t+1) = s' \mid s(t) = s, u(t) = u) = p(s, s', u),$$

$\forall s, s' \in S, u \in U, t \in T$ , where  $S \times S \times U \rightarrow [0, 1]$  is the controlled transition probability function.

If the input applied to the system stays constant, say equal to  $u' \in U$ , irrespectively of the system evolution, then the controlled Markov chain reduces to a standard Markov chain.

In our context,  $u(t)$  is introduced to model the occurrence of a shot-cut event. The control set is in fact defined as  $U = \{0,1\}$ , and if a shot-cut event happens at time  $t$ , then  $u(t)=1$ , otherwise  $u(t)=0$ .

We suppose that the occurrence of a shot-cut event causes the system to change dynamics. In order to model this fact, we describe the state of the system as a two-component state, i.e.,  $s(t) = (x(t), q(t)) \in S = X \times Q$ , where  $q(t) \in Q := \{0,1\}$  is called the mode of the system.

Also, we impose a certain structure on the controlled transition probability function. Specifically, the controlled transition probability function is supposed to satisfy the condition that a shot-cut event forces the controlled Markov chain to change operating mode, whereas if no shot-cut event occurs, then the controlled Markov chain remains in the same mode.

Note that within a single mode, say  $q \in Q$ , the controlled Markov chain reduces to a standard homogeneous Markov chain with state space  $X$ .

We denote by  $\varphi_{x,q}$  the probability distribution of  $x(t+1)$  when  $x(t)$  and  $q(t)$  take values  $x \in X$  and  $q \in Q$ , respectively.

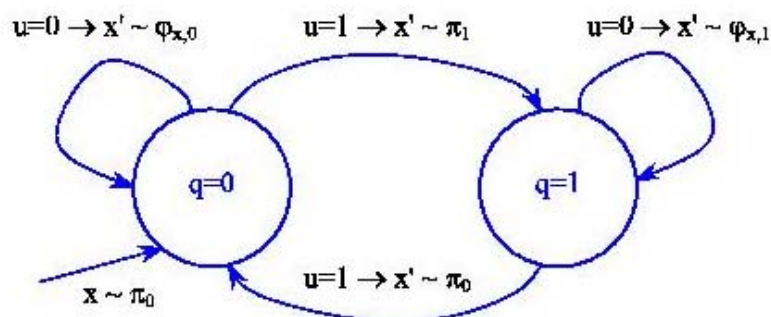
Here, we suppose that each one of the two homogeneous Markov chains admits a stationary probability distribution and we denote by  $\pi_q$  the one associated with mode  $q \in Q$ . Then,  $\pi_q(x)$  is the probability of  $x(t)$  being equal to  $x \in X$  in the long run, when the system remains in mode  $q$ .

We assume that at time  $t=0$ , when we start observing the system evolution, the system is in mode  $q=0$  and in stationary conditions, i.e.,  $P_0(s) = \pi_0(x)$ , if  $s=(x,0)$ , and 0, otherwise.

When a shot-cut event occurs, then the operating mode of the system changes. As for the state component  $x$  we suppose that it is reinitialised as a random variable with a certain fixed distribution. Specifically, we assume that:

$$P((x,q), (x',q'), 1) = \pi_{q'}(x'), \quad q \neq q'.$$

A schematic representation of the introduced model is given in Figure 13. In this figure, the symbol “ $\sim$ ” is used for “distributed according to”.



**Figure 13.** Controlled Markov chain model.

In our context,  $T$  represents the set of time instants associated with the P-frames sequence. As for  $x(t)$ , it is state of the P-frame observed at time  $t$ . In particular,  $x(t)$  can take the following values: "LM", "FP", "FZ", "FPZ", and "Other", hence the set  $X$  has cardinality 5.

The value taken by  $x(t)$  is evaluated by means of the low-level descriptors introduced in the previous sections.

Fix a time instant  $t$  and consider the corresponding P-frame. The state variable  $x(t)$  is said to take the value  $x = \text{"LM"}$  if the descriptor "Lack of motion" is equal to 1. If that is not the case, then,  $x(t)$  can take one of the other 4 values.

Specifically,  $x(t)$  is equal to  $x = \text{"FP"}$  if the value of the descriptor "Fast pan" is 1 and that of the descriptor "Fast zoom" is 0. In the opposite case, i.e., when "Fast pan" is equal to 0 and "Fast zoom" is equal to 1, then,  $x(t)$  takes the value  $x = \text{"FZ"}$ .

In the case when both the "Fast pan" and "Fast zoom" descriptors are equal to 1,  $x(t)$  assumes the value  $x = \text{"FPZ"}$ . In all the other cases,  $x(t)$  is said to take the value  $x = \text{"Other"}$ .

We suppose that each semantic event takes place over a two-shot block and that it can be modelled by a controlled Markov chain with the structure described above. Each semantic event is then characterized by the two sets of probability distributions over the state space  $X$ ,  $P_0$  and  $P_1$ , which govern the evolution of  $x(t)$  within mode  $q=0$  and  $q=1$ , respectively.

Specifically, we have considered 6 models denoted by A, B, C, D, E, and F, where model A is associated to goals, model B to corner kicks, and models C, D, E, F describe other situations of interest that occur in soccer games, such as free kicks, plain actions, and so on. For each event, we have determined the  $P_0$  and  $P_1$  sets of the corresponding model by selecting manually all the pairs of shots related to that event in a set of training sequences, then determining the values taken by  $x(t)$ , in the obtained P-frame sequences, and finally estimating the probabilities  $\varphi_{x,0}$ ,  $\varphi_{x,1}$ .

On the basis of the derived six Markov models, one can classify each pair of shots in a soccer game video sequence by using the maximum likelihood criterion. For each pair of consecutive shots (i.e., two consecutive sets of P-frames separated by shot-cuts), one needs to i) extract the sequence of low-level descriptors, ii) determine the sequence of values assumed by the state variable  $x$ , and iii) determine the likelihood of the sequence of values assumed by  $s=(x, q)$  (with  $q$  set equal to 0 before the shot-cut and to 1 after the shot-cut) according to each one

of the six admissible models. The model that maximizes the likelihood function is then associated to the considered pair of shots.

## Experimental Results

The performance of the proposed algorithm has been tested considering about 2 hours of MPEG sequences containing more than 800 shot-cuts. The sequences contain 9 goals and 16 corner kicks. The obtained results are the following: 8 goals out of 9, and 10 corner kicks out of 16 are detected.

The number of false detections could seem quite relevant. However, these results are obtained using motion information only, so these false detections could probably be reduced by using other type of media information (such as audio loudness). Also, we made the simplifying assumption that the Markov chain with state  $x$  associated to each mode  $q$  is in stationary conditions when entering that mode. One could instead introduce a "reset probability distribution" of  $x$  associated to mode  $q$ , which should then be estimated from data.

## 7.2 Metadata Integration

The "quality" of multimedia information nowadays available is not only an intrinsic property. The real value of a document is strongly related to how it can be retrieved and how one can rapidly browse toward it.

Focusing the attention to the subclass of audio-visual sequences, it can be pointed out that most work has been performed to define suitable frameworks for efficient browsing through this material and for retrieving relevant information according to user specific requirements. Tools that can automatically parse video sequences, classify each segment, and thereby provide non-linear access capability based on the semantic of the content can be provided not only to professional but also to generic users.

To support the above mentioned tools, the description of the content of multimedia documents will be based on the MPEG-7 standard. However, since the introduction of this open standard, mainly considered the objective to guarantee inter-operability between multimedia applications, additional problems need to be solved.

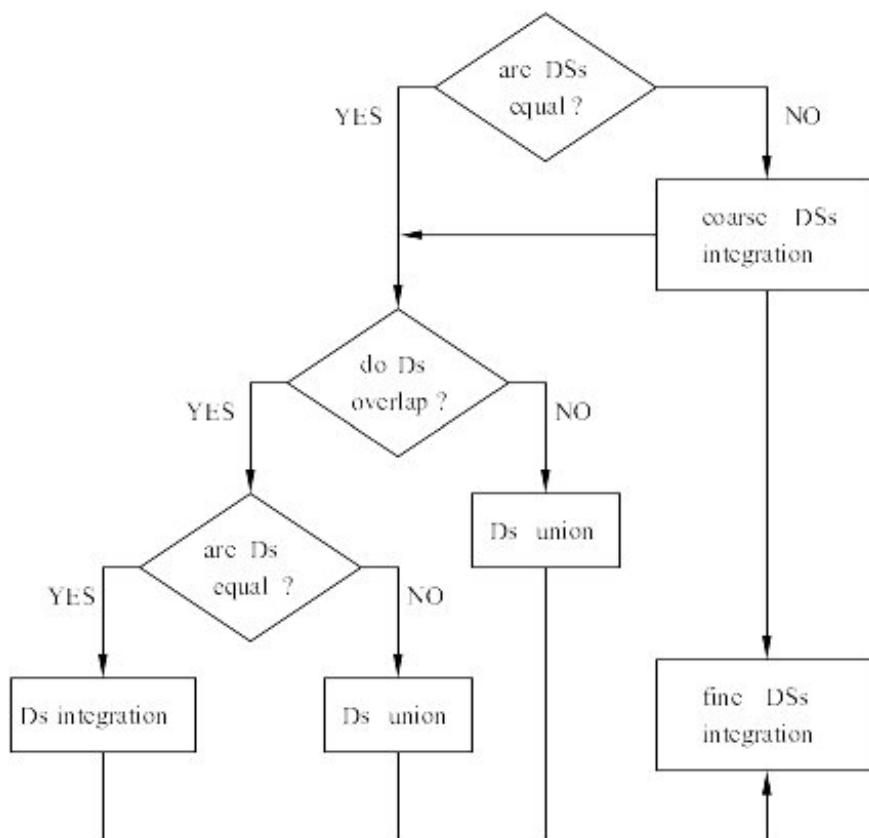
Considering that different descriptions of the same multimedia document will be available to a given application a new question can be raised: how such descriptions can be integrated together. Redundant information must be discarded while complementary ones must be integrated, in order to have a unique and richer description and possibly match a specific need. This would enable a better organization of such content allowing quality information to be retrieved for any specific purpose.

In this section, a general framework is proposed to compare and merge different Description Schemes (DS) and the associated Descriptors (D) describing the content of the same video document.

A specific case study is considered where the objective is to obtain a better temporal segmentation of a video sequence by integrating two separate segment decompositions (in the MPEG-7 sense) in a single partition with a more accurate representation of the shot boundaries. The processed information to reach this result uses two *Dominant Color* series associated to two shot decomposition obtained by applying different temporal segmentation algorithms to a video sequence.

## Integration of different descriptions

The general framework for the integration process is described in the flow chart of Figure 13. In order to show the applicability of the proposed method, the integration of two DSs will be considered. However it is important to stress that it can be applied to an arbitrary number of DSs.



**Figure 13.** Comparison and integration of Descriptions.

**Figure 14.** Example showing how the integration process takes place.

As can be seen in the diagram, if two given description structures are equal there is no need to integrate them at the structure level; only the associated descriptors have to be merged. Vice versa, if the structure of the descriptions do not perfectly match three cases may occur leading to the following steps: coarse DSs integration, Ds integration and fine DSs integration. Clearly the effective operations involved in each step are specialized with respect to the type of DSs and Ds that will be considered. The objective of the following subsections is to show how the result of a coarse description integration can be refined using only the information provided by the associated Ds without to reprocess the original data.

### Preliminary definitions and assumptions

It is assumed to have two description schemes  $DS_1$  and  $DS_2$  describing the structural decomposition of the same video sequence but obtained from different extraction algorithms. Each of them include two sub-DSs:

- ✓ one derived from the *Video Segment* DS [10] type, used to describe the temporal decomposition of a video;
- ✓ the other one derived from *Time Series* DS [9] type, used to associate the *Dominant Color* (DC) descriptor [9] to specified frames of a sequence.

These sub-DSs will be indicated respectively with  $SEG_{1/2}$  and  $DC_{1/2}$ .

Given a certain color space, DC represents a set of dominant colors (minimum number of colors: 1; maximum : 8) that characterize a frame or one of its arbitrarily-shaped regions; for any color in DC, three parameters are used in the computation of the distance measure: *variance*, *probability*, *coherence*. It will be assumed that all the DC instances have been extracted from the video sequences by using the same method in order to guarantee interoperability as specified by the MPEG-7 standard. Finally it is assumed that it is possible to have reliability indices, which can assume normalized values between 0 and 1, for the extraction method used to generate the descriptions. The indices can be used, for example, as indicators of which information has to be preserved when two descriptor instances that have to be merged overlap (spatially and/or temporally). For the DC extraction algorithm considered in this work (defined in the non normative part of the MPEG-7 standard) it is not necessary to define any reliability index because it would be the same for both descriptions. Regarding the temporal segmentation method two indices have been used, the probability of miss detection defined as

$$p^{miss} = \frac{N_{miss}}{N_{RT}} \quad \text{and} \quad p^{false} = \frac{N_{false}}{N_{CT} + N_{false}}$$

the probability of false detection.  $N_{miss}$  is the number of missed transitions and  $N_{RT}$  the number of real transition that characterize the video.  $N_{false}$  is the number of false alarms and  $N_{CT}$  the correctly estimated editing effects. To allow an effective comparison these parameters has to be estimated on the basis of the same ground truth.

### Coarse Segment Decomposition DSs integration

In this phase we estimate how two decompositions into shots  $SEG_{1/2}$  are related to each other.

The comparison can lead to two different results.

- ✓ A shot transition T is present in both the descriptions. In this case a transition has been recognized but especially for the gradual ones the associated shot boundaries could not perfectly match in the two descriptions. Assuming b and e represent respectively the beginning and the end of a shot a possible solution for the integration can given by:  $b = \frac{p_1 b_1 + p_2 b_2}{p_1 + p_2}$  and  $e = \frac{p_1 e_1 + p_2 e_2}{p_1 + p_2}$ , where  $p_{1/2} = 1 - p_{1/2}^{false}$  is the probability of correct recognition of the segmentation methods. If the hypothesis of statistical independence between the two methods is satisfied the new reliability values are updated according to  $p = 1 - p_1^{false} p_2^{false}$ .

- ✓ A shot transition  $T$  is present only in one decomposition. In this case a new interval is added to the temporary final segmentation with the following values:
  - $b=b_1$   $e=e_1$   $p=p_1$  if  $T \in \text{SEG}_1$
  - $b=b_2$   $e=e_2$   $p=p_2$  if  $T \in \text{SEG}_2$

The main problem with this approach is the determination of reliability values that have to be comparable. This means that the probability of miss and false detection have to be calculated applying the same procedure to all the temporal segmentation algorithms.

After this step a new DS for the segmentation into shots, that integrate the information of the starting ones, is obtained ( $\text{SEG}_{12}$ ). This description is characterized by a number of miss and false detections that in the worst case will be respectively the minimum of the miss values and the sum of the false values. However this coarse segmentation can be refined using the information available from individual DC series associated to each description. To facilitate the re-segmentation we start to integrate the DC series into a single description.

### Time Series DSs integration

The *Time Series* DS is used in this work to associate a DC to a specified frame; so the *Time Series* DSs integration is based on the DC integration (Ds integration).

It is possible to use different *Time Series*:

- ✓ *Regular Time Series*: where the DC descriptor is associated to frames that have been obtained by sub-sampling the original video by a fixed factor.
- ✓ *Irregular Time Series*: a DC descriptor is associated to a generic frame while specifying the single gap till the next frame which has an associated DC.

In general, the integration is given by the union of  $\text{DC}_1$  and  $\text{DC}_2$  ( $\text{DC}_{12}$ ) because the given Times Series do not

completely overlap. In the case of a frame described by two different DCs (overlapping) the integration has to consider the differences between the algorithms setup used in the extraction phase of  $\text{DC}_{1/2}$ .

### Fine Segment Decomposition DSs integration

After the integration of  $\text{DC}_{1/2}$  we can expect to have a more dense series of DC. This series  $\text{DC}_{12}$  can be used to refine the temporal segmentation  $\text{SEG}_{12}$  by evaluating the distance between consecutive DC across a transition point. This is a critical operation but as it is shown in the next section a better segmentation can be obtained when the density of DC is sufficiently high.

### Re-segmentation process

A real simulation of the previously described integration procedure has been implemented showing the effectiveness of the proposed method. In the first part of this section, a study on distance measures for establish robust correlation between instances of DC are reported. At the end of the experiment the fine integration of the two DSs is performed showing how the final number of misses and false detections can decrease with respect to the initial one and

how they are related to the DC Frame Gap. Let us first present in the next two subsection the components that need to be compared in order to reach the final description.

### **Dominant Color distance measure**

Two distance measures have been considered: a Euclidean distance and the Earth Mover's Distance.

#### **Euclidean distance**

In this case, the RGB color space has been selected. The distance between two DCs P and Q is defined by:

$$D(P, Q) = \sqrt{\sum_{k=1}^3 \sum_{i,j=1}^N (p_{ik} - q_{jk})^2} \quad (1)$$

where N indicates the number of dominant colors forming the D of each frame,  $p_{ik}$  and  $q_{jk}$  correspond to the i-th dominant color of P and the j-th one of Q, respectively (index k refers to the color component R,G,B).

We can use the Euclidean distance only if some hypothesis are satisfied:

- ✓ each DC must have the same number of dominant colors;
- ✓ the dominant colors are ordered in the same way for both D's, i.e., the first element of the first dominant color is compared with the first element of the second dominant color, ...

It can be observed that the RGB color space appears inadequate as it does not reflect any of the visual properties of the human visual system.

#### **Earth Mover's Distance**

The Earth Mover's Distance (EMD) [11] allows to establish a distance measure between two probability density functions. Since the DC represents a color distribution information, the EMD seems to be a good candidate instead of the Euclidean distance.

Defining a distance between two probability distributions requires a notion of distance between basic patterns of the distribution. This distance is called *ground distance*. For the DC, the ground distance is the distance between each color; the ground basic pattern distance used is the Euclidean distance in the CIE-Lab color space, since this color space is especially designed so that the Euclidean distance strongly correlates with the human ability to discriminate color information.

The two DCs can be seen as two color distributions:  $P = \{(p_1, w_{p1}), (p_2, w_{p2}), \dots, (p_{N_P}, w_{pN_P})\}$  is the first DC, where  $p_i$  is an element of the DC (a color),  $w_{p_i}$  its weight (probability),  $N_P$  the number colors;  $Q = \{(q_1, w_{q1}), (q_2, w_{q2}), \dots, (q_{N_Q}, w_{qN_Q})\}$  is the second DC; let  $\mathbf{D} = [d_{ij}]$  the *ground distance matrix* with  $d_{ij}$  the ground distance (Euclidean distance) between  $p_i$  and  $q_j$ :

$$d_{ij} = \sqrt{\sum_{k=1}^3 (p_{ik} - q_{jk})^2}. \quad (2)$$

The EMD is given by:

$$\text{EMD}(P, Q) = \frac{\sum_{i=1}^{N_p} \sum_{j=1}^{N_Q} d_{ij} f_{ij}}{\sum_{i=1}^{N_p} \sum_{j=1}^{N_Q} f_{ij}}$$

where  $f_{ij}$  represents the flow between between  $p_i$  and  $q_j$ , , that minimizes the overall cost subject to four defined constraints [11].

An alternative ground distance is proposed by [8]; this distance can be used only for the DC:

$$d_{ij} = \sqrt{\sum_{k=1}^3 (p_{ik} - q_{jk})^2 + \sum_{k=1}^3 (\sigma_{pik} - \sigma_{qjk})^2 + \sqrt{(ch_{pi} - ch_{qj})^2}} \quad (3)$$

where  $p_{ik}$  and  $q_{jk}$  are the  $k$ -th color component respectively of the  $i$ -th element of *dominant color* P and of the  $j$ -th one of Q,  $\sigma_{pi}$  and  $\sigma_{qj}$  the  $i$ -th and  $j$ -th element variance,  $ch_{pi}$  and  $ch_{qj}$  their respective coherence.

For simplicity, we shall denote by EMD to indicate the EMD using the Euclidean distance (2) and EMDdc to indicate the EMD using equation (3).

### Integration process

The distance measures introduced in the previous subsection are compared through the following experiment.

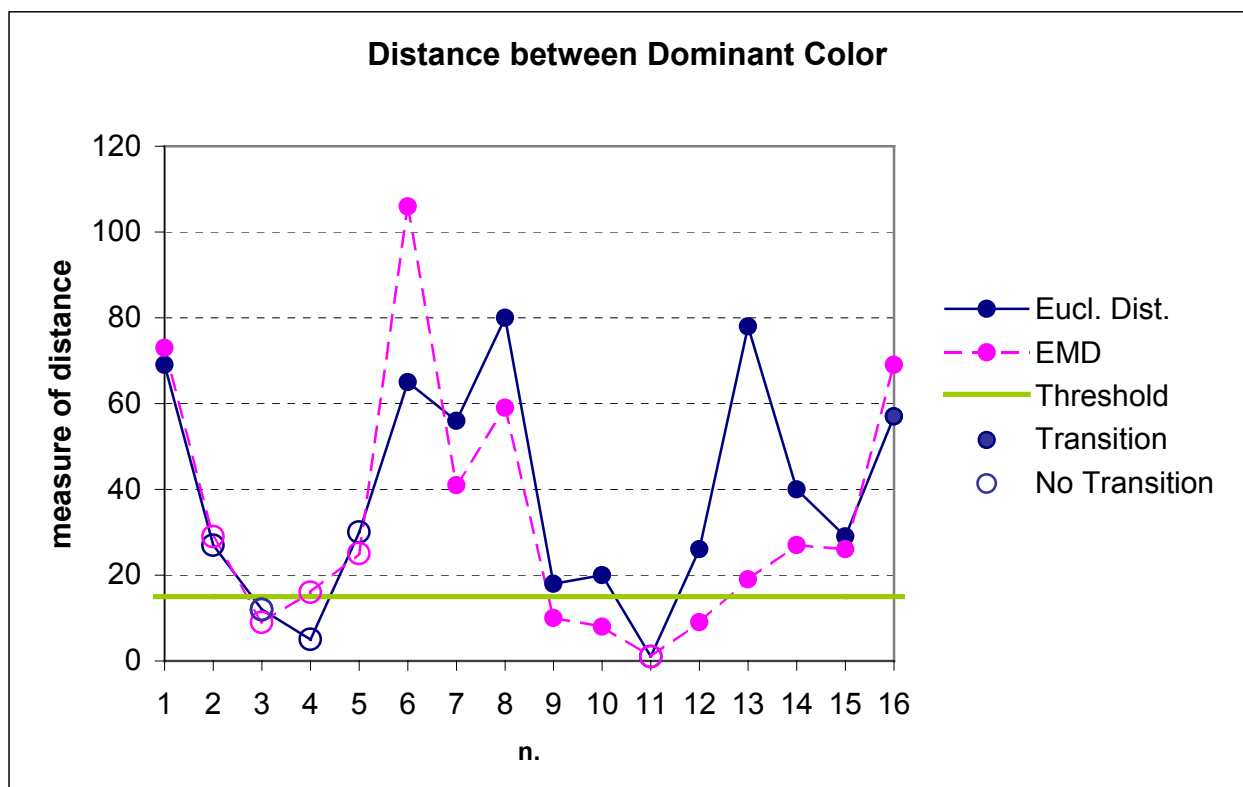
Two different shot boundaries segmentations of a video (1400 frames) are considered:  $SEG_a$  is a shot segmentation extracted by the algorithm proposed in [7], while  $SEG_h$  is a segmentation extracted by hand. Also, a DC with 8 elements is computed every 10 frames of the video sequence, that is at frame 10, 20, etc.

By, comparing  $SEG_a$  and  $SEG_h$ , we observe that  $SEG_a$  is characterized by some false shot transitions. We then try to use the information given by the DC to correct  $SEG_a$ . Specifically, for each shot transition of  $SEG_a$ , we compute the distance between the DC just before the transition and the one just after it. For instance, if there is a transition that starts at frame 51 and ends at frame 52, we compute the distance between the DC of frame 50 and the one of the frame 60. The results are reported in Table 4 (T=shot Transition, NT=No shot Transition).

n.	sega	Eucl. Dist.	EMD	EMD dc	segh
1	51-52	69	73	852	T
2	81-82	27	29	573	NT
3	150-151	12	9	392	NT
4	184-185	5	16	308	NT
5	217-218	30	25	474	NT
6	233-260	65	106	788	T
7	295-316	56	41	893	T
8	622-633	80	59	763	T
9	839-840	18	10	260	T
10	904-905	20	8	264	T

11	956-964	1	1	45	NT
12	1013-1014	26	9	237	T
13	1063-1064	78	19	518	T
14	1156-1157	40	27	380	T
15	1258-1259	29	26	261	T
16	1359-1375	57	69	651	T

**Table 4.** Comparison between distances.



**Figure 15.** Comparison between distances.

From the results shown in Table 4, we can see that, the EMDdc is not a good distance since it is difficult to fix a threshold that can discriminate a shot transition from a non existing shot transition.

As shown in Figure 15, the EMD and the Euclidean distance provide better results: for example, by setting the threshold to  $d_{th}=15$ , the Euclidean distance identifies 3 false shot transitions (n.3, n.4, n.11) while the EMD identifies 2 false shot transitions (n.3, n.11) and 3 false no shot transition (n.9, n.10, n.12). The Euclidean distance seems to offer the optimal trade-off, but it can be used only if there are two DCs with the same number of elements ( $N_p=N_Q$ ); if DCs with different number of elements ( $N_p \neq N_Q$ ) are compared, the EMD must be used.

## Performance Evaluation

Two different shot boundaries segmentations  $SEG_{1/2}$  of a same video (6000 frames) are compared and integrated. We assume that the integration result between  $DC_{1/2}$  ( $DC_{12}$ ) gives a DC with 8 elements every 10 frames. We use the EMD with Euclidean distance as the ground distance. In order to evaluate the performance, we created a ground truth by annotating by hand the correct shot boundaries.

It is thus possible to extract from  $SEG_1$  and  $SEG_2$  the number of missed shot transitions and the number of false transitions:

- ✓  $seg_1$ : 2 missed, 26 false;
- ✓  $seg_2$ : 5 missed, 35 false.

The integration performance is indicated in Figure 16.

As can be seen, for a large sub-sampling factor in the assignment of the DC information (more than 10), the number of missed shot boundaries goes to zero, but the number of false alarms remains approximately constant (to about 25). For a small sub-sampling factor (less than 10), the number of missed boundaries goes not to zero while the number of false ones is reduced to about 15. With a small sub-sampling factor the distance measure is smaller than the one that would have been obtained with a larger sub-sampling factor, since the distance is computed between nearer frames. Thus, in the first case, we tend to cancel more easily false transitions while not avoiding the identification of misses.

The number of false transition does not reduce to zero as:

- ✓ a boundary can be a false transition even if it is present in both  $seg_1$  and  $seg_2$ ;
- ✓ if a shot has a lot of motion activity, near frames are very different; so, the distance measure between the DCs associated to two consecutive frames could be above the selected threshold.

Therefore, a better performance can be expected only with the use of additional Ds. In some cases to reach satisfactory results one must process the original video material.

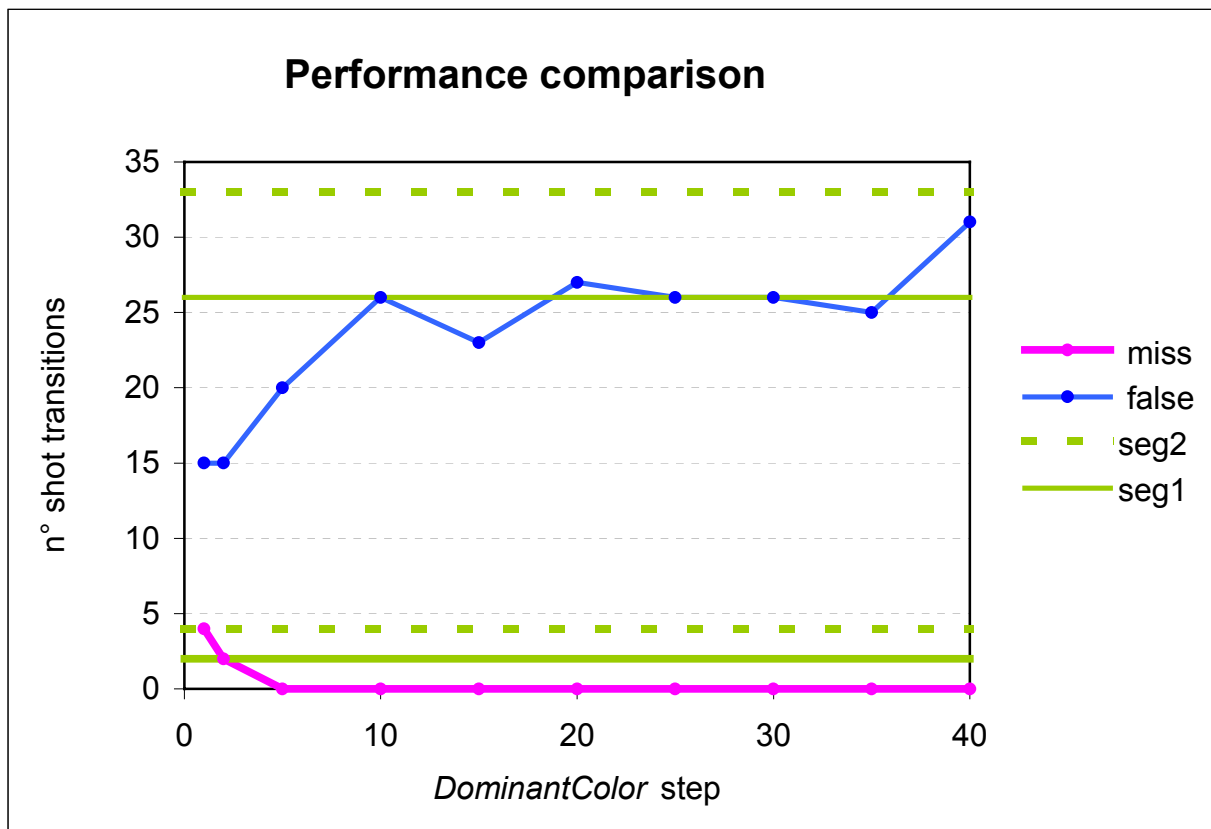


Figure 16 Missed and false detection VS DC Frame Gap

## 8 Conclusions

In this document we analyzed the requirements set by the SPATION scenarios and compared them to the possibilities offered by the MPEG7 and MPEG21 standards. A detailed listing of all applicable description schemes is given. Additionally, the experiments performed to provide input to the MPEG21 standardisation process have been described.

## 9 Bibliography

- [1] Deliverable D2: "Evaluation of network technology".
- [2] Deliverable D3: "Home network usage scenarios".
- [3] Universal Multimedia Access, <http://ltswww.epfl.ch/~newuma/>
- [4] UMA - Universal Multimedia Access from wired or wireless systems, <http://www.midgardmedia.net/UMA/IKT2010-UMA.html>
- [5] O. Steiger, "Smart camera for MPEG-7", Diploma project, Lausanne, EPFL, 2001
- [6] O. Steiger, A. Cavallaro and T. Ebrahimi, "MPEG-7 Description of Generic Video Objects for Scene Reconstruction", In VCIP 2002, Proc. of SPIE, vol. 4671, pp. 947-958, San Jose, CA, USA, Jan. 21-23, 2002.
- [7] N. Adami, R. Leonardi. *Identification of editing effects in image sequences by statistical modelling*. Proc. of PCS99, Picture Coding Symposium. Pages 157-160. Portland - OR, USA, April 1999
- [8] N. Adami, R. Leonardi, Y. Wang. *Evaluation of different descriptors for identifying similar video shots*. Proc. Of ICME2001, International Conference on Multimedia Expo. Pages 948-951, Tokyo, Japan, August 2001.
- [9] MPEG-7 Video Grup. *Multimedia content description interface - Part 3: Visual*. ISO/IEC JTC1/SC29/WG11/N4062. Singapore, March 2001.
- [10] MPEG-7 Video Grup. *Multimedia content description interface - Part 5: Multimedia Description Schemes*. ISO/IEC JTC1/SC29/WG11/N3966. Pisa, January 2001.
- [11] Y. Rubner, C. Tomasi, L.J. Guibas. *A metric for distributions with applications to image databases*. Pages 59-66, Bombay, India, January 1998. Proc. ICCV 1998.
- [12] N. Adami, A. Bugatti, R. Leonardi, P. Migliorati, & L.A. Rossi: "The ToCAI Description Scheme for Indexing and Retrieval of Multimedia Documents", *Multimedia Tools and Application*, Vol. 14 N. 2, 153-173, Kluwer Academic Press, 2001.
- [13] N. Adami, M. Corvaglia and R. Leonardi, "Comparing descriptions of multimedia data for simplification and integration", *In Proc. of IPMU 2002*, Annecy, France, July 2002.
- [14] H. Nishikawa, et al, "Description for Metadata Adaptation Hint", ISO/IEC JTC1/SC29/WG11/M8324, Fairfax, USA, May 2002.
- [15] MDS: "Workplan for CE on Metadata Adaptation", ISO/IEC JTC1/SC29/WG11 N4953, Klagenfurt, July 2002
- [16] R. Lagendijk, "A Position Statement for Panel 1: Image Retrieval", Proc. of the VLBV99, Kyoto, Japan, pp. 14-15, October 29-30, 1999.
- [17] Yao Wang, Zhu Liu, Jin-cheng Huang, "Multimedia Content Analysis Using Audio and Visual Information", IEEE Signal Processing Magazine, vol. 17, no. 6, pp. 12-36, Nov. 2000.
- [18] R. Zhao, W.I. Grosky, "From Features to Semantics: Some preliminary Results", Proc. of IEEE International Conference ICME2000, New York, NY, USA, 30 July - 2 August 2000.
- [19] V. Tovinkere, R. J. Qian, "Detecting Semantic Events in Soccer Games: Toward a Complete Solution", Proc. ICME'2001, pp. 1040-1043, August 2001, Tokyo, Japan.
- [20] A. Bonzanini, R. Leonardi, P. Migliorati, "Semantic Video Indexing Using MPEG Motion Vectors", Proc. EUSIPCO'2000, pp. 147-150, 4-8 Sept. 2000, Tampere, Finland.
- [21] A. Bonzanini, R. Leonardi, P. Migliorati, "Event Recognition in Sport Programs Using Low-Level Motion Indices", Proc. ICME'2001, pp. 920-923, August 2001, Tokyo, Japan.