INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO/IEC JTC1/SC29/WG11/M9849
Trondheim, Norway
July, 2003

Title: **Report of CE on Metadata Adaptation – Integration**

Source: **Nicola Adami, Sergio Benini, Riccardo Leonardi**
(University of Brescia)
Hirofumi Nishikawa (Mitsubishi Electric Corp.)

Status: **Report**

1 Abstract

In this document the Metadata Adaptation tool presented at the Pattaya meeting has been validated, with a slightly adjusted syntax and semantics. An alternative syntax has also been proposed so as to be combined with the adaptation hint DS. This tool supports adaptation operations such as metadata integration.
The presented scheme includes two descriptors, named *AverageValue* and *InvariantProperties*, that can be efficiently used by an integration engine in order to improve the quality of the integrated final description and to speed up the whole integration process.
Examples and experiment results are given to support the validation of the proposed tool.

2 Introduction

The diffusion of recently introduced standards for the description of Multimedia content such as for example MPEG-7 will probably originate a large and sparse amount of multimedia content description instances. This means that in the context of MPEG-21 DIA particular attention has to be paid to develop tools in order to support the adaptation of metadata. These tools have to be developed according to the particular nature and characteristics of metadata that cannot be considered as generic resources. Currently the MetadataAdaptationHint DS is the only one tool of this type proposed in the MPEG-21 DIA SoCD v5.0 (see document N5612 [9]). This DS aims to provide useful information about the structure and the information that characterise a given multimedia content description XML instance. As shown in a previous CE (see document m9024 [1]) this tool cannot cover completely one aspect concerning metadata adaptation; the integration of multiple content description instances of a same document. Therefore a new tool has been proposed in order to improve the DIA support of metadata adaptation: the MetadataIntegration DS. The scheme includes two descriptors that can be efficiently used by an integration engine in order to improve the quality of the integrated final and to speed up the integration process. The first descriptor, named "*AverageValue"*, provides information concerning the average length of a segment in a segment (temporal and/or spatial) decomposition. The main advantage of using this D is for example the possibility to compare two or more tree like segment decompositions by only looking at the *AverageValues* associated to each level of the tree, without the need to fetch all the information contained in the description instances. The second one, named "*InvariantProperties*", tells whether or not a D or DS instance associated to a segment is invariant with respect to spatio-temporal cropping. When this property is true a D or DS can be directly propagated to a sub-segment (without having to access the original multimedia information in order to recalculate the new D or DS value on sub-portions of the content). Two advantages characterise the use of this information: the possibility to build richer descriptions without accessing the original multimedia content (not always available) and no need to recalculate a new D or DS value. In the next section, the concept and related issues to metadata combination and transformation are presented. A complete description of the information provided by the proposed tool, its syntax and semantic are presented in Section 4. In Section 5, an integration example is provided showing the efficiency that can be achieved by using the proposed tool. In Section 6 a possible solution for the merging of syntax of MetadataAdaptationHint DS and of MetadataAdaptationIntegration DS is proposed. Finally in Sections 7 and 8, the CE results and recommendations are reported.

3 Metadata Adaptation

The use of tools for describing multimedia content such as those provided by MPEG-7 will allow application to generate, exchange and consume metadata in an interoperable manner. Besides, as when dealing with content, the side effect will be a production of a large and sparse number of content description instances even for a single piece of material. In the context of MPEG-21 DIA several metadata adaptation processes have been identified on the basis of DIA requirements [2] [5]:

- **Filtering**, to discard the metadata of a description that do not satisfy a certain query. Example: keep all the DominantColor D only if the associated region is bigger than hundred pixels.
- **Scaling**, to reduce the number and the size of metadata according to some constraints. Example: in a tree like segment decomposition with five levels, to prune all the leaves at the third level (in order to reduce for example storage needs)

- **Integration,** to merge two or more descriptions of a given document.
  Example: considering two different spatial decomposition of a same image and the associated descriptors, a unique description can be obtained by merging, according to some criterion or objective, the available information.
- **Segmentation**, to build a description for a portion of a given content using, when it is possible, only the description of the original content (i.e. without the need to reprocess the content)
  Example: considering an image, we can use the corresponding color histogram as a descriptor for a sub-image of the original one. Clearly this operation is possible only under certain conditions (for example the image has a regular pattern and the original image support is not too small). Note that a segmentation could be necessary during the integration process, for example if we have two different spatial decompositions with associated descriptors for a same image.

As reported in the previous CEs (see documents m8589, m8576, m8560) the MPEG-21 DIA tool MetadataAdaptationHint DS can be used to accomplish the first two operations (filtering and scaling), while it is not completely effective in the case of metadata integration (see document m9024).

Considering the general integration scheme of Figure 1 (a more complete description of the method is reported in [10]), it can be noticed how the process can be divided into three main tasks:

- *phase 1, coarse Description Schemes integration.* The structures of two descriptions are compared and integrated.
- *phase 2, Descriptors integration.* Integration of the descriptors.
- *phase 3, fine DSs integration.* The result given by phase 1 is refined on the basis of the consistency of descriptor values (obtained after phase 2) attached to contiguous segments.

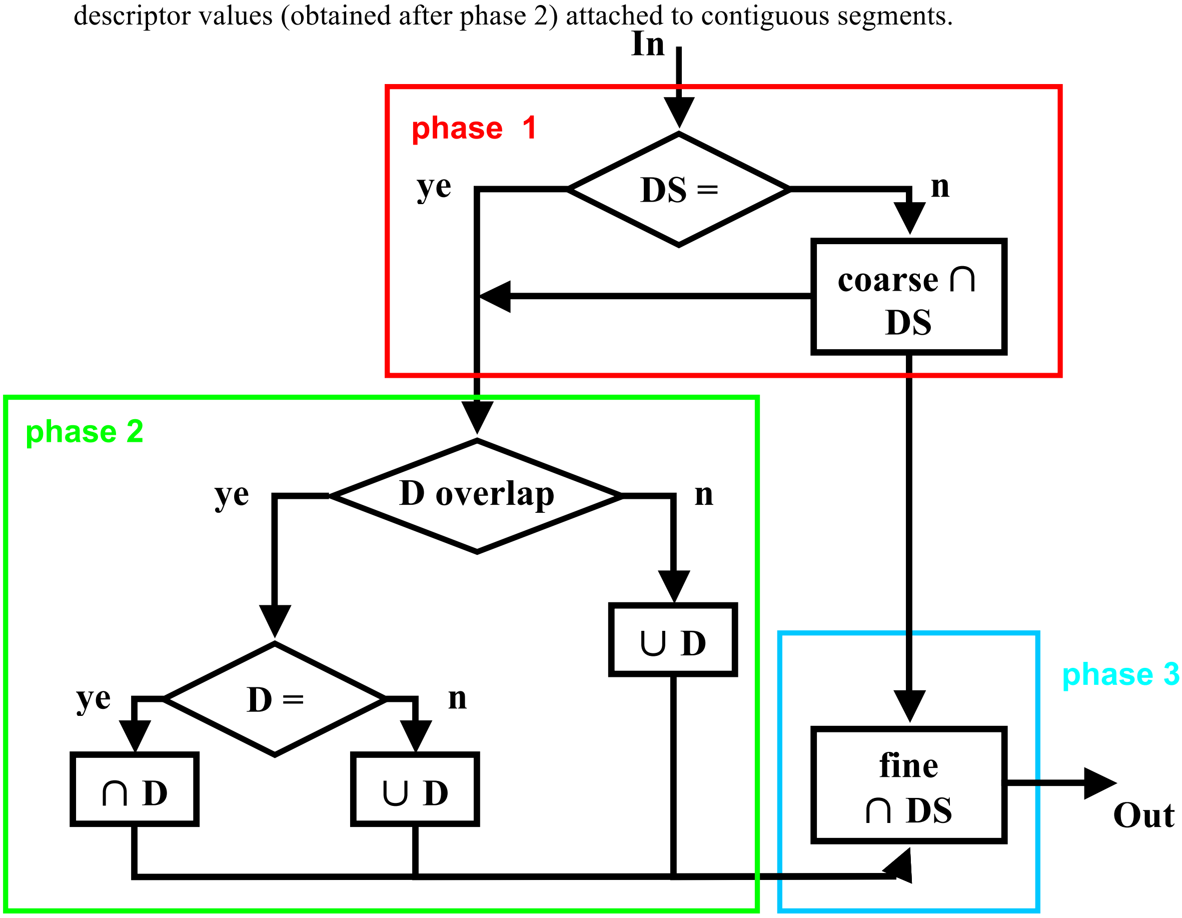


Figure 1 Complete integration process.

- *phase 1 (coarse DSs integration)*
- *phase 2 (Ds integration)*
- *phase 3 (fine DSs integration).*

To better understand how integration process works and the related problems, let us consider an example. The input consists of two different one level spatial decomposition of the same image, where a single segment can hold the ColorHistogram D calculated on the corresponding area. What we want to obtain is a single description with all the original information.

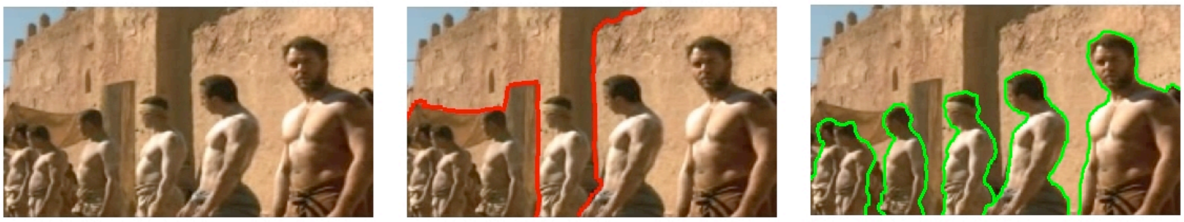


Figure 2: Two different one level spatial decompositions of the same image

Following the scheme of Figure 1 the process starts with the coarse integration of the two spatial decompositions. Basically this can be implemented performing the following operations:

1. find the intersection between the two descriptions (ex: $I = SptS1 \cap SptS2$),
2. add to the first description the part of the second one that does not overlap (ex: $R = SptS1 \cup (SptS2 - I)$), or vice versa.

where the operation corresponding to the union and the intersection depends on the engine implementation. In our case we assumed the following rules:

1. *Intersection*: two segments are considered to be the same if they overlap each other at least for 95%.

2. *Union*: segments have to be nested in a tree like structure. If a segment in totally included in another one it will be considered as child segment.

In order to find the intersection between *SptS1* and *SptS2*, the engine has to calculate the best matching between all the possible pairs of segments. It has to be noticed that this operation presents a complexity of M*N, where M and N are the number of segments respectively in *SptS1* and *SptS2*. Looking at Figure 3 it can be noticed that *SptS2* is a finer decomposition with respect to *SptS1,* and in this case the intersection is empty (no segments overlap for at least 95%). This is because the two decompositions present a different level of partitioning.

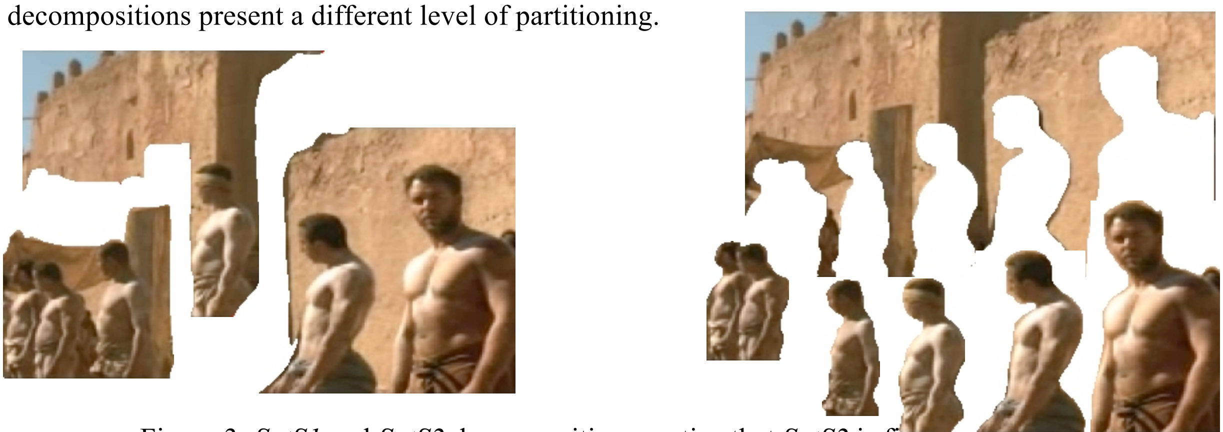


Figure 3: *SptS1* and *SptS2* decompositions: notice that *SptS2* is finer than *SptS1*

In Figure 4 we can see the integration result after *phase I.*

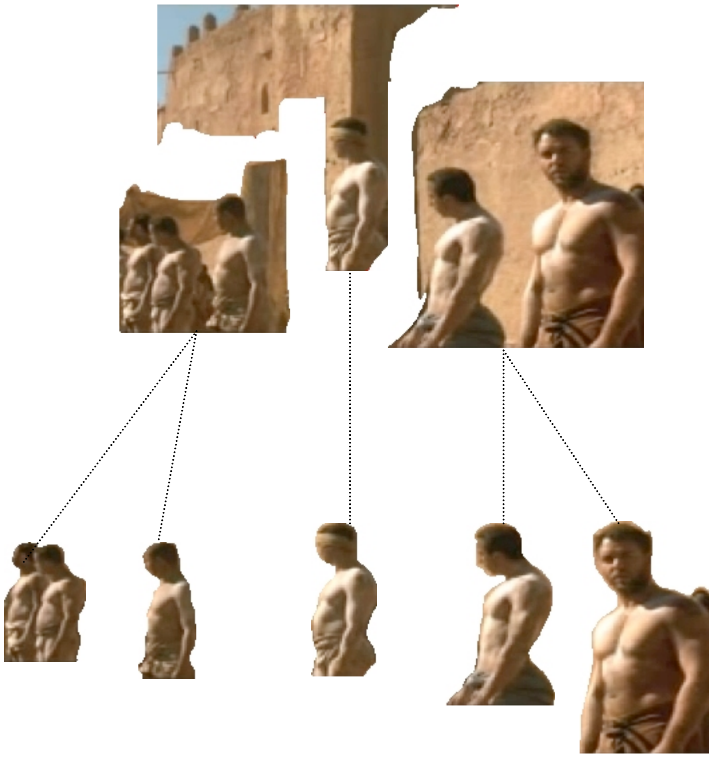


Figure 4: Integration result after Phase I

A way to speed up the integration process is to consider some a priori knowledge of the descriptions that have to be integrated. In our example, such information can be the average value of segment dimension (number of pixels) associated to a given level. With this *AverageValue* information, the engine is able to automatically conclude that segments associated to the first level of SptS2 are unlikely to have the same granularity of segments at the same level in SptS1, and hence they cannot overlap. A numerical example is given considering the MPEG-7 representation of the above image decompositions and the associated AverageValue tools (see Doc. 1, Doc. 2 and Doc. 3.).


```
<DescriptionUnit xsi:type="ImageType">
 <Header xsi:type="Spatial2DCoordinateSystemType"
         id="CDR000" xRepr="1" yRepr="1">
      <Unit>pixel</Unit>
 </Header>
 <Image>
 <SpatialDecomposition>
  <StillRegion>
   <SpatialMask>
    <SubRegion>
      <CoordRef refid="CRD000"/>
      <Polygon>
        <Coords mpeg7:dim="2 6"> 0 288 0 144 90 134
                                 90 110  120 110 120 288</Coords>
      </Polygon>
    </SubRegion>
    <SubRegion>
      <CoordRef refid="CRD000"/>
      <Polygon>
       <Coords mpeg7:dim="2 8"> ... </Coords>
      </Polygon>
      </SubRegion>
```

```
      <SubRegion>
       <CoordRef refid="CRD000"/>
       <Polygon>
        <Coords mpeg7:dim="2 7"> ... </Coords>
       </Polygon>
      </SubRegion>
     </SpatialMask>
    </StillRegion>
   </SpatialDecomposition>
  </Image>
</DescriptionUnit>
```

Doc. 1: mpeg-7 representation of the spatial decomposition SptS1.

```
<Description xsi:type="MetadataAdaptationIntegrationType"
                     instanceSchema="urn:mpeg:mpeg7:schema:2001"

instanceLocator="../../repository/descriptions/SptS1.xml">
      <Component name="SubRegion">
            <AverageValue value="33792" unitOfMeasure="CoordRef">
                  <Location type="SpatialDepth">1</Location>
            </AverageValue>
      </Component>
</Description>
```

Doc. 2: XDI associated to SptS1.

```
<Description xsi:type="MetadataAdaptationIntegrationType"
                     instanceSchema="urn:mpeg:mpeg7:schema:2001"

instanceLocator="../../repository/descriptions/SptS2.xml">
      <Component name="SubRegion">
            <AverageValue value="9876" unitOfMeasure="CoordRef">
                  <Location type="SpatialDepth">1</Location>
            </AverageValue>
      </Component>
</Description>
```

Doc. 3: XDI associated to SptS2.

Another a priori information useful for the integration process could be the knowledge whether or not a D or DS instance associated to a segment is invariant with respect to spatio-temporal cropping. When this property (which we call *InvariantProperties*) is true a D or DS can be directly propagated to a sub-segment without accessing the original data in order to recalculate the new D or DS value. For example starting from a color segmented object in a frame in a video sequence, if motion parameters associated to the frame are invariant, the same motion field can be attached to anyone object in a segmentation of the frame.

Thanks to these initial considerations it is easy to understand how the use of *AverageValue* and *InvariantProperties* can speed up the integration process of compatible descriptions of a given content. For this purpose, a first prototype of DI metadata integration engine has been developed according to the guideline reported in the CE workplan document.

4 Metadata Adaptation Integration DS

This DS can be considered as a complement of the currently available DIA Metadata Adaptability tools. It can be used to describe structural features of a description instance such as, for example, an MPEG 7 Segment DS instance. As it can be seen in Figure 5 the DS has two main elements. The first one is useful to describe an average feature value associated to a set of components; for example the average length of a group of VideoSegment DS. This value can be efficiently used by an integration engine for comparing tree like segment decompositions without the need to parse and compare every pairs of segments. The *InvariantProperties* element is used to specify whether the value of a descriptor associated to a segment is invariant with respect to a segment cropping. With respect to the syntax currently defined in the MPEG-21DIA AM v5.0 some minor modifications have been introduced according to the comments of reviewers. The modifications mainly concern the InvariantProperties element where all the information provided by the attributes spatial, temporal have been embedded in a new child element named “InvariantDimension”.

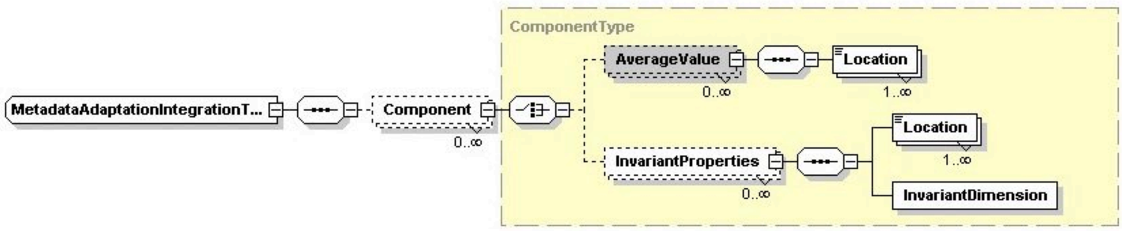


Figure 5: Graphical view of MetadataAdaptationIntegration type

Syntax of the MetadataAdaptationIntegrationType:

```
<xs:complexType name="MetadataAdaptationIntegrationType">
   <xs:complexContent>
      <xs:extension base="DIADescriptionType">
         <xs:sequence>
            <xs:element name="Component" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:complexContent>
                     <xs:extension base="ComponentType">
                        <xs:attribute name="name" type="xs:QName" use="required"/>
                     </xs:extension>
                  </xs:complexContent>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
         <xs:attribute name="instanceSchema" type="xs:anyURI" use="required"/>
         <xs:attribute name="instanceLocator" type="xs:anyURI" use="required"/>
      </xs:extension>
   </xs:complexContent>
```

```
</xs:complexType>

<xs:complexType name="ComponentType">
   <xs:complexContent>
      <xs:extension base="DIABaseType">
         <xs:choice>
            <xs:element name="AverageValue" minOccurs="0" maxOccurs="unbounded">
               <xs:complexType>
                  <xs:complexContent>
                     <xs:extension base="DIABaseType">
                        <xs:sequence>
                           <xs:element name="Location" maxOccurs="unbounded">
                              <xs:complexType>
                                 <xs:simpleContent>
                                    <xs:extension base="xs:string">
                                       <xs:attribute name="type" type="xs:string"
use="required"/>
                                    </xs:extension>
                                 </xs:simpleContent>
                              </xs:complexType>
                           </xs:element>
                        </xs:sequence>
                        <xs:attribute name="unitOfMeasure" type="xs:string"
use="required"/>
                        <xs:attribute name="value" type="xs:string" use="required"/>
                     </xs:extension>
                  </xs:complexContent>
               </xs:complexType>
            </xs:element>
            <xs:element name="InvariantProperties" minOccurs="0"
maxOccurs="unbounded">
               <xs:complexType>
                  <xs:complexContent>
                     <xs:extension base="DIABaseType">
                        <xs:sequence>
                           <xs:element name="Location" maxOccurs="unbounded">
                              <xs:complexType>
                                 <xs:simpleContent>
                                    <xs:extension base="xs:string">
                                       <xs:attribute name="type" type="xs:string"
use="required"/>
                                    </xs:extension>
                                 </xs:simpleContent>
                              </xs:complexType>
                           </xs:element>
                           <xs:element name="InvariantDimension">
                              <xs:complexType>
                                 <xs:attribute name="dimension" type="xs:string"
use="required"/>
                                 <xs:attribute name="invariantFactor"
type="mpeg7:zeroToOneType" use="required"/>
                              </xs:complexType>
                           </xs:element>
                        </xs:sequence>
                     </xs:extension>
                  </xs:complexContent>
               </xs:complexType>
            </xs:element>
         </xs:choice>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

Semantics of the MetadataAdaptationIntegrationType:

Name	*Definition*
`MetadataAdaptationIntegrationType`	Provides auxiliary information about an XML instance, useful in order to decrease the computational time and to improve the output quality in a metadata integration process.
`instanceSchema`	Describes the schema of the XML instance. (required)
`instanceLocator`	A pointer to the description instance the Metadata Adaptation Match is describing. (required)
`Component`	Describes the information about a component that is used in an XML instance.
`Name`	Identifies the name of a description tool that represents the component. (required)
`AverageValue`	Describes an average feature (element or attribute) of a set of description tools components (D or DS).
`value`	Indicates the average value . (required)
`unitOfMeasure`	Identifies the feature (element or attribute) from which the unit of measure can be derived. (required)
`InvariantProperties`	Gives information about invariant properties of a description tool.
`InvariantDimension`	Gives information about the dimension with respect the D or DS it is invariant to cropping.
`dimension`	Indicates with respect to wich dimension the D or DS value is invariant (required). This attribute can assume one the following values: • *Spatial,* indicates the instancs value is invariant to spatial cropping • *Temporal,* indicates the instancs value is invariant to temporal cropping • *SpatioTemporal,* indicates the instancs value is invariant to spatio-temporal cropping
`ivariantFactor`	Indicates the cropping factor. A D or DS instance value can be considered ivariant if the considered subsegmet size is biger than the size of the original segment multiplyed by this factor (required). It can assume values from 0 to 1 where 0 means fully invariant and 1 not invariant.
`Location`	Describes the location where the description tool can be described in the XML instance.
`type`	Indicates the type of location. The following values shall be used: (required) • *TemporalDepth and SpatialDepth*: In case of tree like decomposition, identifies the relative spatio/temporal position, with respect to the root, of the considered set of components, to locate a set of description tools in a spatio/temporal decomposition • *listOfID*: `Location` can be represented as a list of ID.

In the following example, the MetadataAdaptationIntegration DS is used to describe the average duration of a set of VideoSegment DS instances.

```
<DIA>
 <Description xsi:type="MetadataAdaptationIntegrationType"
             instanceSchema="urn:mpeg:mpeg7:schema:2001"
             instanceLocator="../../repository/descriptions/VS1.xml">
  <Component name="VideoSegmentDS">
     <AverageValue value="PT0M10S" unitOfMeasure="MediaDuration">
         <Location type="listOfID">shot#1 shot#2... shot#10</Location>
     </AverageValue>
     <AverageValue value="PT0M10S" unitOfMeasure="MediaDuration">
         <Location type="TemporalDepth">1</Location>
     </AverageValue>
  </Component>
 </Description>
</DIA>
```

Aim: Propagation
of semantic

The following description has been used to describe the *InvariantProperties* of a ColorHistogram DS. The description says that the instance value is invariant to spatial variation of the associated media until the variation is smaller than the 50%. This can be the case when for example, the global histogram associated to an image is considered.

```
<DIA>
 <Description xsi:type="MetadataAdaptationIntegrationType"
              instanceSchema="urn:mpeg:mpeg7:schema:2001"
             instanceLocator="../../repository/descriptions/VS1.xml">
  <Component name="ColorHistogramDS">
      <InvariantProperties>
         <Location type="listOfID">ColH#1 ColH#2</Location>
         <InvariantDimension dimension="spatial" invariantFactor="0.5"/>
      </InvariantProperties>
  </Component>
 </Description>
</DIA>
```

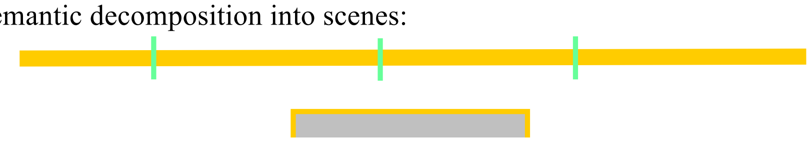
5 Integration examples

In this section a comprehensive example of the integration process performed by an engine is reported. As an example of Digital Item Integration in which *InvariantProperties* are applied, we show how semantic labels of XML schema can be propagated between two different temporal segmentations of the same video sequence.

Suppose to have a first supervised segmentation result (that we call S1), which divides the video sequence into semantic coherent segments. Moreover consider a complete automatic shot-based segmentation of the same video sequence called S2, in which no semantic is associated to each shot.

In Figure 6 the considered situation is shown:

Semantic decomposition into scenes:



S1 manual segmentation

S2 automatic segmentation

Shot decomposition (no semantic associated to shots)

Figure 6: Two different segmentation S1 and S2 of the same video sequence

In our example we consider a video sequence extracted from the MPEG-7 content set sequence *jornaldonoite.mpg,* subdivided into three semantic scenes in a manual segmentation S1 (with the associated labels -A: soccer; -B: anchorman; -C: marathon) and into 29 shots in an automatically obtained segmentation S2.
A possible MPEG-7 XML description of one video segment (semantic scene) of the S1 segmentation is given by:

```
…
<VideoSegment id="Scene C">
      <MediaTime>
<MediaTimePoint>T00:00:30</MediaTimePoint>
                  <MediaDuration>PT1M49S</MediaDuration>
      </MediaTime>
      <Semantic>
                    <Label>
                  <Name>marathon news</Name>
                    </Label>
                          <SemanticBase xsi:type="EventType">
                  <Label>
                              <Name>marathon</Name>
                  </Label>
                          </SemanticBase>
      </Semantic>
</VideoSegment>
…
```

Figure 7: MPEG-7 XML description of scene "marathon" belonging to S1 segmentation

in which the semantics associated to the scene (C: marathon) is highlighted.
On the other side, regarding S2 segmentation, the associated MPEG-7 XML description of one shot is given by:

```
…
<VideoSegment id="Shot 6">
          <MediaTime>
                    <MediaTimePoint>T00:00:37</MediaTimePoint>
                    <MediaDuration>PT0M3S</MediaDuration>
          </MediaTime>
</VideoSegment>
<VideoSegment id="Shot 7">
…
```

Figure 8: MPEG-7 XML description of one shot belonging to S2 segmentation

where no semantic type is associated to shots (note that the shot in this XML example is included the "marathon" scene).
Metadata Integration tool allows to merge the two MPEG-7 XML descriptions combining data as showed in Figure 9 below:

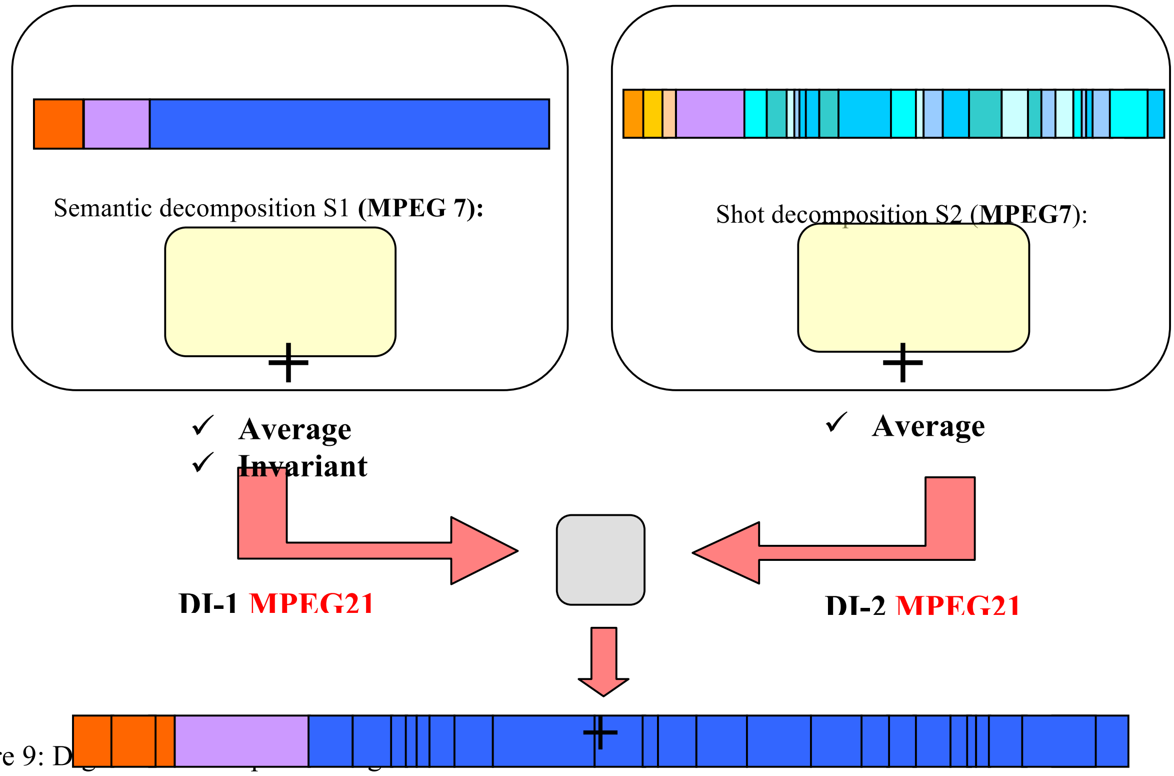


Figure 9: D

In this way, starting from the two MPEG-7 schema we obtain, thanks to the application of *InvariantProperties* and *AverageValue*, two MPEG-21 compliant Digital Items, which are named DI-1 and DI-2. After the merging process of Digital Item Adaptation the obtained result is a new Digital Item (DI-3) which contains the integrated MPEG-21 XML schema.
The XML description of the single shot of our example obtained as a result of the metadata adaptation-integration process becomes:

```
…
<VideoSegment id="Shot 6">
          <MediaTime>
                    <MediaTimePoint>T00:00:37</MediaTimePoint>
                    <MediaDuration>PT0M3S</MediaDuration>
          </MediaTime>
          <Semantic>
                    <Label>
                               <Name>marathon news</Name>
                     </Label>
                     <SemanticBase xsi:type="EventType">
                               <Label>
                                           <Name>marathon</Name>
                               </Label>
                     </SemanticBase>
          </Semantic>
</VideoSegment>
<VideoSegment id="Shot 7">
 …
```

Figure 10: XML description resulting from metadata adaptation process

in which, besides the temporal description already provided by the S2 description, the associated semantic of S1 schema has been included.

Since *InvariantProperties* expresses the possibility to apply the same descriptor value to sub-parts of a description (and if the property is valid there is no need to recalculate the descriptor value!), starting from the supervised scene labelling (S1), the semantics of the video segments of S1 has propagated to each single shot of S2.
Moreover the use of *AverageValue* allowed to speed up the adaptation process.

Associating semantics to each single shot could seem rather trivial in some cases, for example in a soccer game scene (semantic A in our example) in which the semantic of every single shot can be easily fixed at a glance (Figure 11 presents two key-frames of the scene decomposition).

Figure 11: Key-frames of two “soccer” shots in which semantic is clear

This is also the case of the second semantic scene (B: anchorman), in which only one shot is present and the semantic is rather obvious to any observer.

Figure 12: Key-frame of an “anchorman” shot in which semantic is clear

But in general, and this comprises most real cases, the semantics of a single shot can be hardly inferred only by its view or the view of some key-frames. For instance, considering the third semantics scene in our example (C: marathon) the “marathon” concept can hardly be associated to shots without a supervised labelling and without observing the whole sequence.
In Figure 13 some key-frame of some “marathon” shots are presented: it is clear that the semantics “marathon” cannot be easily inferred from a single key-frame (except may be from the third key-frame).

Figure 13: Key-frames of three "marathon" shots in which semantic is NOT clear

In these cases, the application of *InvariantProperties* is an effective method to propagate the semantic labels, obtaining in this way a reacher description, avoiding the need to reprocess the original content and reducing potential information redundancy.

As an additional benefit, it is worth indicating that the integration engine works as an effective method for boundary refinement. In fact if manual segmentation presents inaccurate time boundaries between segment, for example not starting (or ending) exactly from a between-shot transition, the integration process can easily recover the exact temporal boundaries relying on the derived finer segmentation (as shown below).

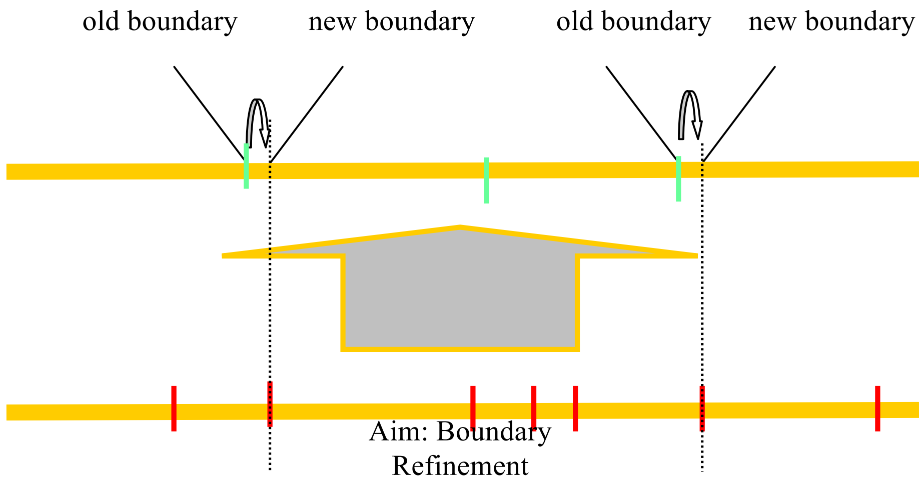


Figure 14: Automatic boundary refinement relying on finer segmentation

6 Possible unified syntax for Hint and Integration DS

MetadataAdaptation Hint and Integration DS are both describing characteristics of an XML instance and they use similar mechanisms for component referencing. Looking at the two syntax and semantics we suggest as an alternative a possible solution for merging the two description schemes syntaxes as in Figure 15.

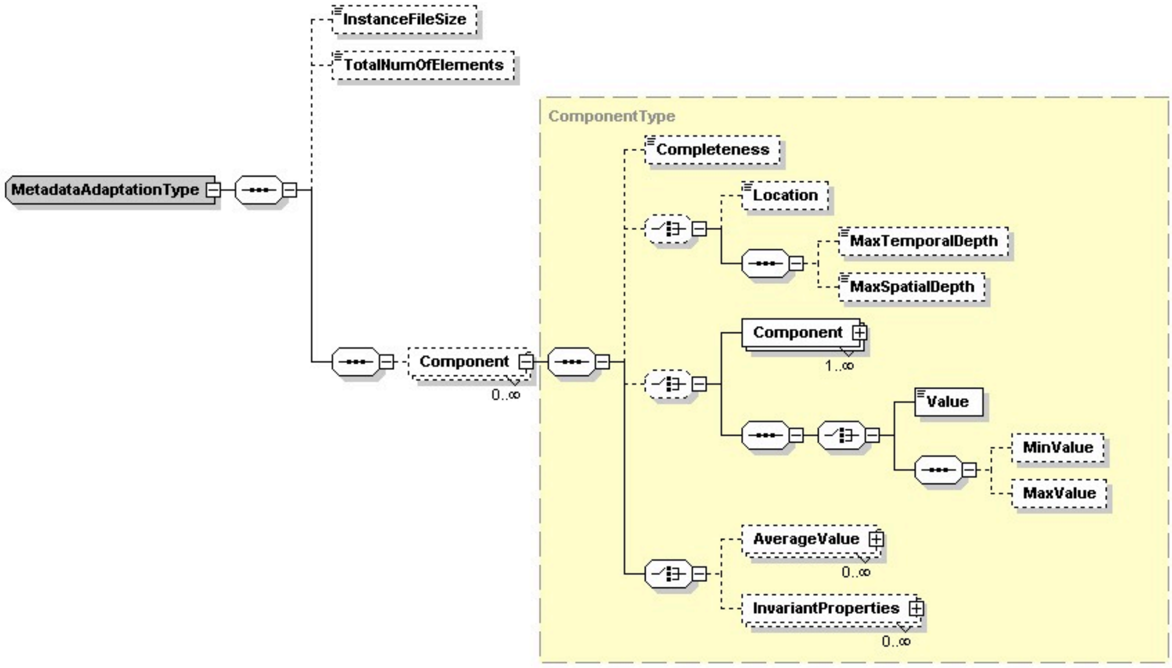


Figure 15: A possible unified syntax for MetadataAdaptation Hint and Integration

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- Digital Item Adaptation ISO/IEC 21000-7 -->
<!-- MetadataAdaptationHint schema -->
<xs:schema targetNamespace="urn:mpeg:mpeg21:dia:schema:2003"
xmlns="urn:mpeg:mpeg21:dia:schema:2003" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:include schemaLocation="DIA.xsd"/>
   <!-- ######################################### -->
   <!--  Definition of MetadataAdaptationHint        -->
   <!-- ######################################### -->
   <xs:complexType name="MetadataAdaptationType">
      <xs:complexContent>
         <xs:extension base="DIADescriptionType">
            <xs:sequence>
               <xs:element name="InstanceFileSize" type="xs:nonNegativeInteger"
minOccurs="0"/>
               <xs:element name="TotalNumOfElements" type="xs:nonNegativeInteger"
minOccurs="0"/>
               <xs:sequence>
                  <xs:element name="Component" type="ComponentType" minOccurs="0"
maxOccurs="unbounded"/>
               </xs:sequence>
            </xs:sequence>
            <xs:attribute name="instanceSchema" type="xs:anyURI" use="required"/>
            <xs:attribute name="instanceLocator" type="xs:anyURI" use="required"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
   <xs:complexType name="ComponentType">
      <xs:complexContent>
         <xs:extension base="DIABaseType">
            <xs:sequence>
               <xs:element name="Completeness" type="xs:boolean" minOccurs="0"/>
               <xs:choice minOccurs="0">
                  <xs:element name="Location" minOccurs="0">
```

```
                    <xs:complexType>
                       <xs:simpleContent>
                          <xs:extension base="xs:string">
                             <xs:attribute name="type" type="xs:string"/>
                          </xs:extension>
                       </xs:simpleContent>
                    </xs:complexType>
                 </xs:element>
                 <xs:sequence>
                    <xs:element name="MaxTemporalDepth"
type="xs:nonNegativeInteger" minOccurs="0"/>
                    <xs:element name="MaxSpatialDepth" type="xs:nonNegativeInteger"
minOccurs="0"/>
                 </xs:sequence>
              </xs:choice>
              <xs:choice minOccurs="0">
                 <xs:element name="Component" type="ComponentType"
maxOccurs="unbounded"/>
                 <xs:sequence>
                    <xs:choice>
                       <xs:element name="Value">
                          <xs:complexType>
                             <xs:simpleContent>
                                <xs:extension base="xs:string">
                                   <xs:attribute name="type" type="xs:string"/>
                                </xs:extension>
                             </xs:simpleContent>
                          </xs:complexType>
                       </xs:element>
                       <xs:sequence>
                          <xs:element name="MinValue" minOccurs="0"/>
                          <xs:element name="MaxValue" minOccurs="0"/>
                       </xs:sequence>
                    </xs:choice>
                 </xs:sequence>
              </xs:choice>
              <xs:choice>
                 <xs:element name="AverageValue" minOccurs="0"
maxOccurs="unbounded">
                    <xs:complexType>
                       <xs:complexContent>
                          <xs:extension base="DIABaseType">
                             <xs:sequence>
                                <xs:element name="Location" maxOccurs="unbounded">
                                   <xs:complexType>
                                      <xs:simpleContent>
                                         <xs:extension base="xs:string">
                                            <xs:attribute name="type"
type="xs:string" use="required"/>
                                         </xs:extension>
                                      </xs:simpleContent>
                                   </xs:complexType>
                                </xs:element>
                             </xs:sequence>
                             <xs:attribute name="unitOfMeasure" type="xs:string"
use="required"/>
                             <xs:attribute name="value" type="xs:string"
use="required"/>
                          </xs:extension>
                       </xs:complexContent>
                    </xs:complexType>
                 </xs:element>
                 <xs:element name="InvariantProperties" minOccurs="0"
maxOccurs="unbounded">
                    <xs:complexType>
                       <xs:complexContent>
                          <xs:extension base="DIABaseType">
                             <xs:sequence>
                                <xs:element name="Location" maxOccurs="unbounded">
```

```
                                    <xs:complexType>
                                       <xs:simpleContent>
                                          <xs:extension base="xs:string">
                                             <xs:attribute name="type"
type="xs:string" use="required"/>
                                          </xs:extension>
                                       </xs:simpleContent>
                                    </xs:complexType>
                                 </xs:element>
                                 <xs:element name="InvariantDimension">
                                    <xs:complexType>
                                       <xs:attribute name="dimension"
type="xs:string" use="required"/>
                                       <xs:attribute name="invariantFactor"
type="xs:float" use="required"/>
                                    </xs:complexType>
                                 </xs:element>
                              </xs:sequence>
                           </xs:extension>
                        </xs:complexContent>
                     </xs:complexType>
                  </xs:element>
               </xs:choice>
            </xs:sequence>
            <xs:attribute name="name" type="xs:QName" use="required"/>
            <xs:attribute name="number" type="xs:nonNegativeInteger"/>
         </xs:extension>
      </xs:complexContent>
   </xs:complexType>
</xs:schema>
```

7 CE Results

With respect to the overall CEs the following objectives have been achieved:

- Conceptual descriptions (by example) of *AverageValue* and *InvariantProperties*
- Precise Syntax and Semantics definition for the MetadataAdaptationIntegration tool.
- Experimental demonstration that the *AverageValue* information can help the engine in order to speed up the integration Process (previous CE Report)
- Example of an application of *InvariantProperties*, for semantic propagation.

8 Conclusion and recommendations

The results evaluation lead us to the conclusion that the DIA Metadata Adaptability tools defined in the current version of MPEG21 DIA CD are useful, but not exhaustive in order to support adaptation operations such as metadata integration. For this reason a new Metadata Adaptability tool has been validated, with a precise definition of its syntax and semantics. Hence the final recommendation of the CE is to include the *MetadataAdaptationIntegration DS in the MPEG-21 DIA FCD* and use it as an additional scheme for Metadata Adaptability DIA tools.

9 References

[1] N. Adami, M. Corvaglia, R. Leonardi, “CE on Metadata Adaptation – Integration of multiple descriptions”; ISO/IEC JTC1/SC29/WG11 M9024, Shanghai, October 2002.

[2] H.Nishikawa, “Report of CE on Metadata Adaptation”, ISO/IEC JTC1/SC29/WG11 M8589, Klagenfurt, Austria, July 2002

[3] K. Kazui: “Result of CE on Metadata Adaptation”, ISO/IEC JTC1/SC29/WG11 M8576, Klagenfurt, July 2002

[4] M. Sasaki: “Report and Study of CE on Metadata Adaptation”, ISO/IEC JTC1/SC29/WG11 M8560, Klagenfurt, July 2002

[5] N. Adami, R. Leonardi, “Content and Description Scalability Issues in a Home Network Environment”; ISO/IEC JTC1/SC29/WG11 M8693, Klagenfurt, July 2002.

[6] Description for Metadata Adaptation Hint, Mitsubishi, ISO/IEC JTC1/SC29/WG11 m8324, Fairfax, May 2002

[7] MPEG-21 Requirements on Digital Item Adaptation, ISO/IEC/JTC1/SC29/WG11 N4684, Jeju, March 2002

[8] Requirements, MPEG-21 Requirements on Digital Item Adaptation, ISO/IEC/JTC1/SC29/WG11 N4684, Jeju, March 2002

[9] MDS, Study of ISO/IEC CD 21000-7 Digital Item Adaptation, ISO/IEC/JTC1/SC29/WG11/N5612, Pattaya , March 2003

[10] N. Adami, M. Corvaglia and R. Leonardi, "Comparing the quality of multiple descriptions of multimedia documents", *MMSP 2002 Workshop*, St. Thomas (US Virgin Islands), Dec. 2002

[11] MDS, “Work plan for CE on Metadata Adaptation – Integration”; ISO/IEC JTC1/SC29/WG11 N5182, Shanghai, October 2002.

[12] MDS, “Work plan for CE on Metadata Adaptation – Integration”; ISO/IEC JTC1/SC29/WG11 N5362, Awaji, December 2002.