

**INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11
MPEG00/M6588
October 2000, La Baule**

Source: Weight DS Core Experiment participants
Status: Report
Title: CE report on Weight DS
Author: Alessandro Bugatti (University of Brescia)
Contributors: Riccardo Leonardi (University of Brescia)
Benoit Mory (Philips)
Yoshiaki Shibata (Sony)

1	INTRODUCTION	2
2	OUTLINE OF CE ACTIVITY.....	2
2.1	PARTICIPANTS AND FUNCTIONALITIES TO BE VALIDATED	2
2.2	METHODOLOGY.....	2
3	EXPERIMENTAL RESULTS	2
3.1	NEW SYNTAX OF ORDERING KEY DS	2
3.1.1	<i>Syntax and semantics definition of the Ordering Key DS</i>	<i>2</i>
3.1.2	<i>Description extraction</i>	<i>4</i>
3.1.3	<i>Description examples</i>	<i>4</i>
3.1.4	<i>Validation experiments.....</i>	<i>6</i>
4	CONCLUSIONS AND RECOMMENDATIONS	8
5	REFERENCES	8

1 Introduction

In this document, we report the result of our core experiment on the Ordering Key DS. In the last meeting a decision is made to specialize the Weight DS [m6227] in a new DS in order to achieve ordering functionalities. This new DS is the Ordering Key DS which indicates which descriptors can be used in order to obtain an ordering of generic DSs (Segment, Concept, Object, Event).

The main objectives of this CE are:

- an improvement of the syntax which is currently part of the XM document
- to clarify explicitly the semantics of this DS so that users can understand what they should obey in their application
- to demonstrate the benefit to have such component for ordering purposes.

2 Outline of CE activity

2.1 Participants and functionalities to be validated

The main activities and participants are summarized in the below table

Name	Company	Functionality
Yoshiaki Shibata	Sony	Definition of new syntax of Ordering Key DS and check the formal correctness of the instantiation of this DS
Alessandro Bugatti, Riccardo Leonardi	University of Brescia	Definition of new syntax of Ordering Key DS, instantiation of Ordering Key DS for ordering functionality and realization of a demo in order to demonstrate ordering functionalities
Benoit Mory	Philips	Cross-check of results of Brescia experiments

2.2 Methodology

The steps of this CE are summarized in the following points:

- a new Ordering Key DS syntax definition has been made based on that included in MDS XM 4.0, specifying the OrderingKey syntax in the same way as key/unique are defined.
- an instantiation of the OrderingKey DS was built to demonstrate its utility to obtain a semantic ordering applied to a segment set (video segment).
- a demo was built to show ordering keys functionalities. There is an XML file MDS compliant validated by XML 3.0 Spy for each instance in the demo.

3 Experimental results

3.1 New syntax of Ordering Key DS

3.1.1 Syntax and semantics definition of the Ordering Key DS

The OrderingKey DS syntax contained in MDS XM 4.0 was improved in order to obtain a better mechanism to individuate the ordering key field inside the set of objects which have to be ordered.

```
<!-- xpath datatype definition -->
```

```
<simpleType name="xpath" base="string"/>
```

```

<!-- ##### -->
<!--      OrderingKey DS as a basic element      -->
<!-- ##### -->

<element name="OrderingKey" type="mpeg7:OrderingKeyType"/>

<complexType name="OrderingKeyType">
  <element name="selector" type="mpeg7:xpath"/>
  <element name="field" type="mpeg7:xpath" minOccurs="1"
maxOccurs="unbounded"/>
  <attribute name="id" type="ID" use="optional"/>
  <attribute name="name" type="string" use="optional"/>
  <attribute name="semantics" type="string"/>
  <attribute name="direction" use="default" value="descending">
    <simpleType base="string">
      <enumeration value="descending"/>
      <enumeration value="ascending"/>
    </simpleType>
  </attribute>
</complexType>

```

where the semantic of each component is summarized as

Semantics of the datatypes:

<i>Name</i>	<i>Definition</i>
xpath	The syntax of OrderingKey DS is basically the same as <unique> or <key> except the main element, i.e, the key for the ordering which is specified by <selector> and <field> elements based on XPath [XPath]. This datatype is only a "tag" in order to clarify that selector and field elements have to contain an XPath expression.

Semantics of OrderingKey:

<i>Name</i>	<i>Definition</i>
OrderingKey	Type of OrderingKey element in this scheme. It allows to indicate which can be a useful descriptor in order to obtain ordering functionalities.
Selector	An XPath expression which indicates the objects (segments, objects, events) that have to be ordered.
Field	An XPath expression which indicates the ordering key, i.e. the descriptor used to obtain an order between the object indicate from Selector element. The descriptors used as ordering keys have to be either strings or single numeric values: in this way an alphabetical or magnitude criterion is used to obtain the order. It is not possible to use complex descriptor as ordering key to avoid ordering ambiguity (e.g. the color histogram is not a valid ordering key because it is a vector of integer and it would not be clear for the applications how to order)
id	Identifier for each instance of the OrderingKey DS. Optional
Name	A label for the instance of the OrderingKey DS. Optional
Semantics	Element describing the semantics of the ordering. It can be a simple label indicating the descriptor used (e.g. if it has been used the id of video segments as ordering key semantics could be "accordingtoSceneID") or a more complex semantic description of the ordering results (e.g. if PanSpeed inside video segments of a

	soccer match is used as ordering key a possible semantics could be "Shots containing a close-up of a player")
direction	The direction of the ordering (ascending or descending)

3.1.2 Description extraction

N.A.

3.1.3 Description examples

The OrderingKey DS is used to obtain an ordering between objects of the same type (segments, objects, events).

In the first example a simple ordering has obtained based on the *id* of the scene that have to be ordered. In the second one some camera motion descriptors are used in order to obtain useful ordering.

Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<Mpeg7Unit xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:schemaLocation="http://www.mpeg7.org/2001/MPEG-7_Schema
c:\users\bugatti\mpeg7\WeightDS\mds4CE.xsd">

  <VideoSegment id="program">
    <SegmentDecomposition DecompositionType="temporal">
      <VideoSegment id="scene1">
        <MediaTime>
          <MediaTimePoint/>
        </MediaTime>
      </VideoSegment>
      <VideoSegment id="scene2">
        <MediaTime>
          <MediaTimePoint/>
        </MediaTime>
      </VideoSegment>
      <VideoSegment id="scene3">
        <MediaTime>
          <MediaTimePoint/>
        </MediaTime>
      </VideoSegment>
    </SegmentDecomposition>
    <MediaTime>
      <MediaTimePoint/>
    </MediaTime>
  </VideoSegment>

  <OrderingKey name="SceneOrdering" semantics="accordingToSceneId">
    <selector>/VideoSegment/SegmentDecomposition/VideoSegment
    </selector>
    <field>@id</field>
  </OrderingKey>
</Mpeg7Unit>
```

In this case the ordering will be Scene1 → Scene2 → Scene3 according to the alphabetical order applied to the selected ordering key (the direction is descending because it is the default direction).

Example 2

```
<?xml version="1.0" encoding="UTF-8"?>
<MPEG7Unit xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:schemaLocation="http://www.mpeg7.org/2001/MPEG-7_Schema
C:\users\Bugatti\UPMSAnnotator\mds4CE.xsd">

<VideoSegment id="id1">
  <SegmentDecomposition DecompositionType="temporal">
    <VideoSegment id="id2">
      <MediaTime>
        <MediaTimePoint/>
      </MediaTime>
      <CameraMotion NumSegmentDescription="3" DescriptionMode="1">
        <CameraMotionSegment>
          <FractionalPresence>
            <TRACK_LEFT_F>0.23</TRACK_LEFT_F>
            <ZOOM_IN_F>0.56</ZOOM_IN_F>
          </FractionalPresence>
          <Amount/>
        </CameraMotionSegment>
      </CameraMotion>
    </VideoSegment>
    <VideoSegment id="id3">
      <MediaTime>
        <MediaTimePoint/>
      </MediaTime>
      <CameraMotion NumSegmentDescription="3" DescriptionMode="1">
        <CameraMotionSegment>
          <FractionalPresence>
            <TRACK_LEFT_F>0.1</TRACK_LEFT_F>
            <ZOOM_IN_F>0.8</ZOOM_IN_F>
          </FractionalPresence>
          <Amount/>
        </CameraMotionSegment>
      </CameraMotion>
    </VideoSegment>
    <VideoSegment id="id4">
      <MediaTime>
        <MediaTimePoint/>
      </MediaTime>
      <CameraMotion NumSegmentDescription="3" DescriptionMode="1">
        <CameraMotionSegment>
          <FractionalPresence>
            <TRACK_LEFT_F>0.67</TRACK_LEFT_F>
          </FractionalPresence>
          <Amount/>
        </CameraMotionSegment>
      </CameraMotion>
    </VideoSegment>
  </SegmentDecomposition>
</MediaTime>
```

```

        <MediaTimePoint/>
    </MediaTime>
</VideoSegment>
<OrderingKey name="ActionOrdering" semantics="The shots containing more
action">
    <selector>/VideoSegment/SegmentDecomposition/VideoSegment
    </selector>
    <field>/VideoSegment/SegmentDecomposition/VideoSegment/Camer
aMotion/CameraMotionSegment/FractionalPresence@TRACK_LEFT_F
    </field>
</OrderingKey>

<OrderingKey name="Close-up Ordering" semantics="The shots containing
close-up of a player">
    <selector>/VideoSegment/SegmentDecomposition/VideoSegment
    </selector>
    <field>/VideoSegment/SegmentDecomposition/VideoSegment/Camer
aMotion/CameraMotionSegment/FractionalPresence@ZOOM_IN_F
    </field>
</OrderingKey>

</MPEG7Unit>

```

In this case it is possible to use two camera motion descriptors (the track left and zoom in descriptor) in order to obtain some useful ordering.

3.1.4 Validation experiments

Two demos were built to show ordering key functionalities and instantiations of these two examples are contained in files SoccerMatch.xml and News.xml. Both the files was validated using XML Spy 3.0 and a MDS WD and XM schema from Yoshiaki with adds in order to validate the above examples (mds4CE.xsd).

In the first demo is possible to order the shots based on their camera movement Pan speed obtaining in first positions the shots containing close-up of players.

In the second one a cluster of the shots containing the anchor man is obtained using the dominant color of the key frame representing the shots as ordering key.

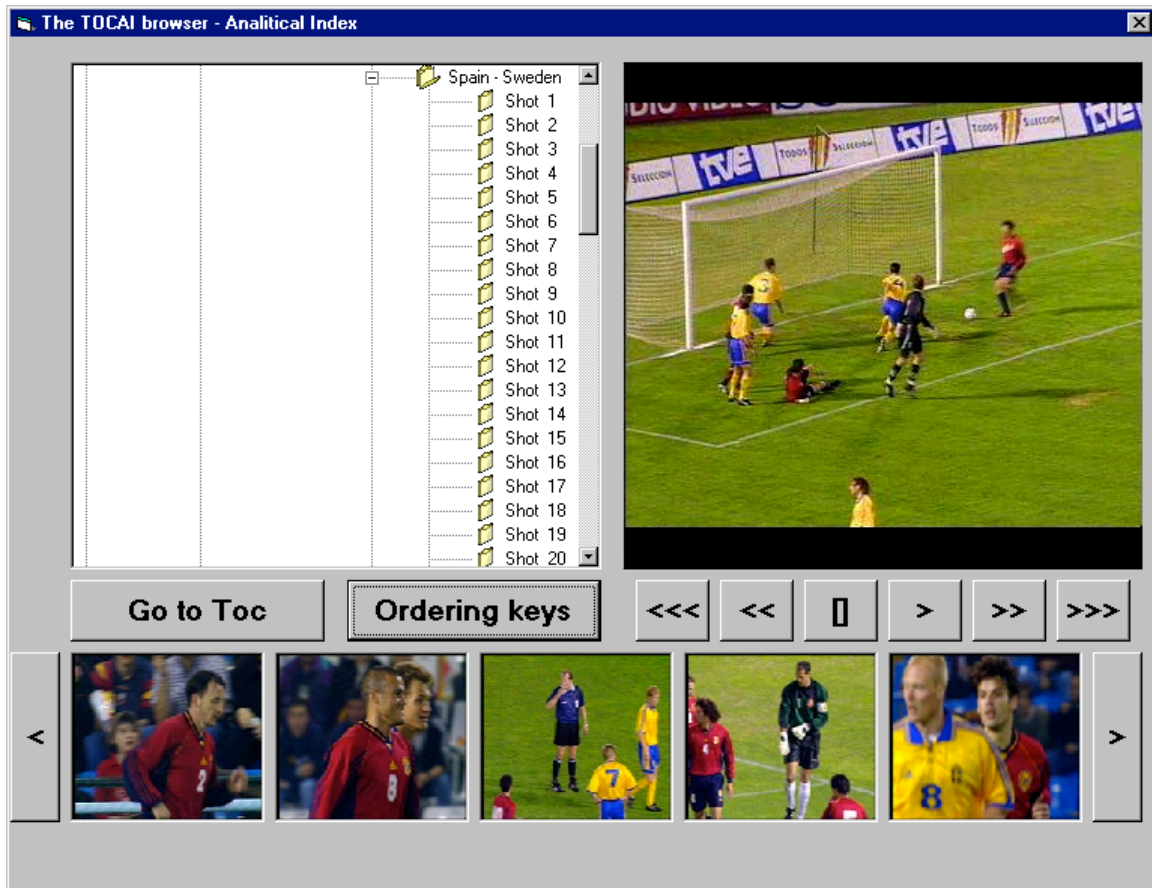


Figure1 - The shots contained in the video of the soccer match Spain-Sweden are ordered according on Pan speed ordering key. As you see the first shots contain close-up of players.

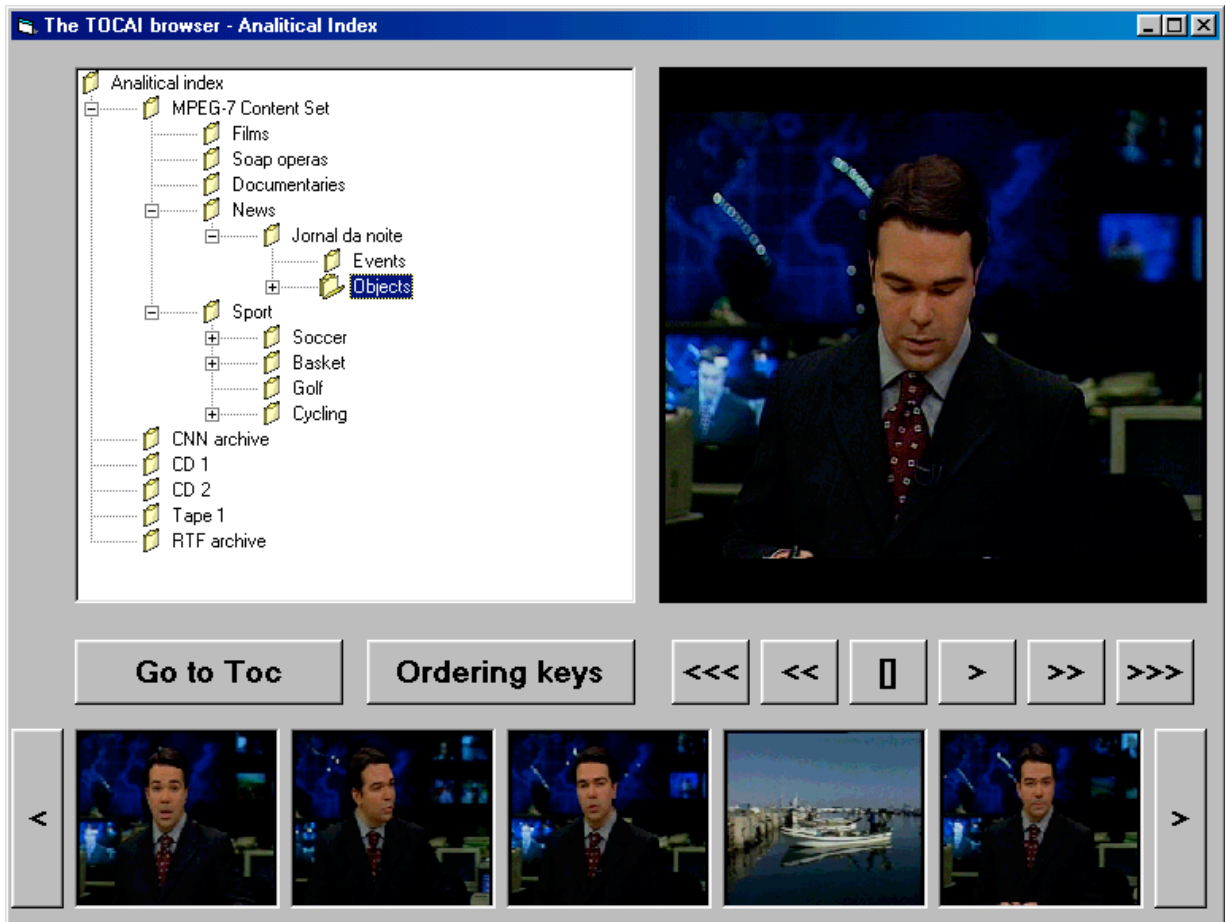


Figure 2 - In this case it is possible to order the segment according to their dominant color associated to the key frame representing the shot. This ordering key can be applied on shots of 'Jornal da Noite' portuguese news: in this way it is possible for example to cluster all the shots containing the speaker in spite of their temporal position in the multimedia stream, as you see in the figure above.

4 Conclusions and recommendations

Based on the previous CE on the Weight DS, we have improved the syntax and semantics in order to avoid ambiguity to the applications that will use this DS. A nex mechanism to indicate the ordering key and the objects that have to be ordered has been experimented: it is based on the XPath and thanks to the XML Schema convention, the OrderingKey (supported by MPEG-7) will be understood easily by the MPEG-7 users.

Finally we have demonstrated through a demo that this new syntax of OrderingKey DS can support ordering functionalities.

5 References

[m6227] ISO/IEC JTC1/SC29/WG11/N2996, CE report on Weight DS, A. Bugatti et al., Beijing, July 2000.

[XPath] XML Path Language (XPath) - Version 1.0 - W3C Recommendation - 16 November 1999