

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

**ISO/IEC JTC1/SC29/WG11 MPEG2014/M33211
March 2014, Valencia, Spain**

Source Canon Research Centre France, Telecom ParisTech,
Status For consideration at the 108th MPEG Meeting
Title Dependency type signaling in DASH
Author Franck Denoual, Frédéric Mazé, Eric Nassor, Cyril Concolato, Jean Le Feuvre,

1 Introduction

At MPEG#107, two contributions ([2], [3]) proposed to associate metadata Representations to video Representations. It was suggested in [2] to distinguish between dependencies that would correspond to coding dependencies and dependencies that would reflect associations between Representations. This could apply to [3]. The DASH subgroup report [4](section 12.2) suggested investigating this point. We propose, in case a new attribute is created, to go further by enabling more precise description of the dependency between Representations.

2 Current Dependency signaling in DASH

The MPD schema provides the `Representation@dependencyId` to express dependencies between Representations.

Definition: specifies all complementary Representations the Representation depends on in the decoding and/or presentation process as a whitespace-separated list of values of `@id` attributes.

Constraints:

- For any dependent Representation X that depends on complementary Representation Y, the m-th Subsegment of X and the n-th Subsegment of Y shall be non-overlapping (as defined in 4.5.3) whenever m is not equal to n.
- For dependent Representations the concatenation of the Initialization Segment with the sequence of Subsegments of the dependent Representations, each being preceded by the corresponding Subsegment of each of the complementary Representations in order as provided in the `@dependencyId` attribute shall represent a conforming Subsegment sequence as defined in 4.5.4 conforming to the media format as specified in the `@mimeType` attribute for this dependent Representation.

During MPEG#107, limitations of `@dependencyId` were highlighted [2] and it was decided to investigate the possibilities of extending the dependency signaling [4].

3 Proposal: more flexible dependency signaling in DASH

In addition to existing `Representation@dependencyId`, we propose to add an optional `Representation@dependencyType` to qualify a dependency. For example, the dependency type could reflect the track reference type value from 'trf' in ISOBMFF and accept values registered for track reference types in [5]. This works well for the metadata use case presented in [3].

3.1 Green metadata use case

The example of [3] would become:

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:XXXX"
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:xxxx"
  type="dynamic"
  minimumUpdatePeriod="PT2S"
  timeShiftBufferDepth="PT30M"
  availabilityStartTime="2011-12-25T12:30:00"
  minBufferTime="PT4S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">

  <BaseURL>http://cdn1.example.com/</BaseURL>
  <BaseURL>http://cdn2.example.com/</BaseURL>

  <Period>
    <!-- Video -->
    <AdaptationSet
      id="video"
      mimeType="video/mp4"
      codecs="avc1.4D401F"
      frameRate="30000/1001"
      segmentAlignment="true"
      startWithSAP="1">
      <BaseURL>video/</BaseURL>
      <SegmentTemplate timescale="90000" media="$Bandwidth$/$Time$.mp4v">
        <SegmentTimeline>
          <S t="0" d="180180" r="432"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="v0" width="320" height="240" bandwidth="250000"/>
      <Representation id="v1" width="640" height="480" bandwidth="500000"/>
      <Representation id="v2" width="960" height="720" bandwidth="1000000"/>
    </AdaptationSet>
    <!-- English Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en" segmentAlignment="0">
      <SegmentTemplate timescale="48000" media="audio/en/$Time$.mp4a">
        <SegmentTimeline>
          <S t="0" d="96000" r="432"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="a0" bandwidth="64000" />
    </AdaptationSet>
    <!-- French Audio -->
    <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="fr" segmentAlignment="0">
      <SegmentTemplate timescale="48000" media="audio/fr/$Time$.mp4a">
        <SegmentTimeline>
          <S t="0" d="96000" r="432"/>
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="a0" bandwidth="64000" />
    </AdaptationSet>
    <!--AdapatationSet carrying Green Video Information for Video -->
    <AdaptationSet id="green_video" codecs="gvme.depr"/>
    <BaseURL>video green depr/</BaseURL>
    <SegmentTemplate timescale="90000" media="$id$/$Time$.mp4m">
      <SegmentTimeline>
        <S t="0" d="180180" r="432"/>
      </SegmentTimeline>
    </SegmentTemplate>
    <Representation id="gv0" bandwidth="1000" dependencyId="v0" dependencyType='cdsc' />
    <Representation id="gv1" bandwidth="1000" dependencyId="v1" dependencyType='cdsc' />
    <Representation id="gv2" bandwidth="1000" dependencyId="v2" dependencyType='cdsc' />
  </AdaptationSet>
</Period>

</MPD>
```

Using the @dependencyType attribute, the DASH client is informed that the metadata Representation provides description of the dependent video Representation (meaning of 'cdsc'). Without this new @dependencyType attribute, if the client selects one of the

metadata Representation, it would be led to download the dependent video Representation since the @dependencyId attribute described it as a complementary Representation on which the metadata Representation depends for decoding and/or presentation.

This can be used for MPEG-Green metadata or quality metrics metadata and provides richer information than a simple associateId as suggested in [2].

3.2 Describing multi-track tile encapsulation in MPD

The proposed @dependencyType attribute can be used in other use cases than metadata tracks, for example when a video is encapsulated into multiple tracks with some having dependencies.

This can be the case for scalable video:

- a first track containing the base layer track is described in MPD by a Representation with @id=1
- a second track providing the enhancement layer and depending on the base track is described in MPD by a Representation with @id=2.

Representation with @id=2 has a @dependencyId=1 to reflect the encapsulation and inter-track dependencies. To indicate explicit behavior to the DASH client regarding the download of this complementary Representation, the @dependencyType for Representation with @id=2 can be set to 'scal' to indicate a coding dependency (thus mandatory).

In multiview scenarios, @dependencyType could be used with value 'vdep' to qualify the dependency between a Representation describing auxiliary video depth and the Representation describing the related video.

3.3 Proposed modifications

In section 5.3.5.2, Table 7 – Semantics of Representation element, replace

@dependencyId	O	specifies all complementary Representations the Representation depends on in the decoding and/or presentation process as a whitespace-separated list of values of @id attributes. If not present, the Representation can be decoded and presented independently of any other Representation. This attribute shall not be present where there are no dependencies.
---------------	---	---

With

@dependencyId	O	specifies all complementary Representations the Representation depends on in the decoding and/or presentation process as a whitespace-separated list of values of @id attributes. If not present, the Representation can be decoded and presented independently of any other Representation. This attribute shall not be present where there are no dependencies.
@dependencyType	O	specifies the kind of dependency for each complementary Representation the Representation depends on that has been signaled with the @dependencyId attribute. Values taken by this attribute are the reference types registered for the trak reference types at http://www.mp4ra.org/trakref.html *. If not present, it is assumed that the Representation depends on the complementary Representations for decoding and/or presentation process without more precise information. This attribute shall not be present when @dependencyId is not present.

* ideally codec-independent code points

Accordingly, the MPD schema would be modified:

```

<!-- Representation -->
<xs:complexType name="RepresentationType">
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SubRepresentation" type="SubRepresentationType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="SegmentBase" type="SegmentBaseType" minOccurs="0"/>
        <xs:element name="SegmentList" type="SegmentListType" minOccurs="0"/>
        <xs:element name="SegmentTemplate" type="SegmentTemplateType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="id" type="StringNoWhitespaceType" use="required"/>
      <xs:attribute name="bandwidth" type="xs:unsignedInt" use="required"/>
      <xs:attribute name="qualityRanking" type="xs:unsignedInt"/>
      <xs:attribute name="dependencyId" type="StringVectorType"/>
      <xs:attribute name="mediaStreamStructureId" type="StringVectorType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Into

```

<!-- Representation -->
<xs:complexType name="RepresentationType">
  <xs:complexContent>
    <xs:extension base="RepresentationBaseType">
      <xs:sequence>
        <xs:element name="BaseURL" type="BaseURLType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SubRepresentation" type="SubRepresentationType" minOccurs="0"
maxOccurs="unbounded"/>
        <xs:element name="SegmentBase" type="SegmentBaseType" minOccurs="0"/>
        <xs:element name="SegmentList" type="SegmentListType" minOccurs="0"/>
        <xs:element name="SegmentTemplate" type="SegmentTemplateType" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="id" type="StringNoWhitespaceType" use="required"/>
      <xs:attribute name="bandwidth" type="xs:unsignedInt" use="required"/>
      <xs:attribute name="qualityRanking" type="xs:unsignedInt"/>
      <xs:attribute name="dependencyId" type="StringVectorType"/>
      <xs:attribute name="dependencyType" type="StringVectorType"/>
      <xs:attribute name="mediaStreamStructureId" type="StringVectorType"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

4 Conclusion

We recommend WG11 to extend MPD syntax with the proposed attribute to allow finer grain descriptions of representation dependencies, thereby allowing a client to decide what is needed or not to download for the processing of a representation.

We suggest using ISOBMF track references as a starting point for identifiers, but codec-independent code point could also be envisioned.

5 References

- [1] [m32202](#) by M. Hirabayashi and S. Hattori: “Additional parameters proposal on CE SRD”, MPEG meeting 107, San José, January 2014.
- [2] [m32190](#) by X. Wang and S. Zhang: “Analysis of Dependency and Association Relationships of Representations in DASH”, MPEG meeting 107, San José, January 2014.
- [3] [m32291](#) by P.Gendron, X. Ducloux, J. Le Feuvre: “*Metadata Representation Carrying Green Information Signalling for DASH*”, MPEG meeting 107, San José, January 2014.
- [4] [m32655](#) by I. Sodagar: “DASH Subgroup Report”, MPEG meeting 107, San José, January 2014.

- [5] MPEG-4 Registration Authority: registered types for track references at: <http://www.mp4ra.org/trackref.html>
- [6] [w14123](#): “WD of ISO/IEC 14496-15:2013 AMD 1 Enhanced carriage of HEVC and support of MVC with depth information”, MPEG meeting 107, San José, January 2014.