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

ISO/IEC JTC1/SC29/WG11 MPEG2012/M25978
July 2012, Stockholm, Sweden

Source Telecom ParisTech
Status **For consideration at the 101st MPEG meeting**
Title **Comments on carriage of timed text and graphics**
Author Cyril Concolato, Jean Le Feuvre

1 Introduction

This contribution reviews the proposed Working Draft for the Carriage of Timed Text in the ISO Base Media File Format, proposes additional requirements and use cases and discusses modifications to existing ISOBMFF constructs to fulfill the requirements.

2 General comments about Timed Text

Multiple technologies are already available to represent timed text (subtitles, closed captioning …). Some of them rely on raster images and some other on text. Some of them are recognized international standards and some others are de facto standards on the Internet. A good overview of the subtitle jungle is provided on the VideoLan Wiki[1].

As the SMPTE, EBU and W3C efforts show, it is likely that the future will give birth to yet another format (with better features, for sure). Unfortunately, each format so far requires MPEG to standardize a new mechanism for its carriage. For instance, MPEG has already defined how to carry MPEG-4 BIFS streams, MPEG-4 LASeR streams or 3GPP Timed Text in MP4 files. These three technologies can already and efficiently represent subtitles including fonts, images[2].

3 Additional use case

We propose the following additional use cases, for which we believe, the technical solution to the requirements and use cases in N12644 can also be used.

Frame-based synchronized graphics overlay on top of a video
Devices such as the Microsoft Kinect can be used to produce video content as well as synthetic information, which can be represented by graphics content, in 3D or in 2D. Such content can represent, for instance, the skeleton of the user moving in front of the camera. There is currently no standard format for storing this kind of visual timed data. Similarly, in augmented reality recording or cloud processing applications, it is useful to stream/store synchronized graphics on top of a video. For such applications, it might be interesting to store the timed graphics together with the video content and with properties similar to those of subtitles proposed in N12644

[1] http://wiki.videolan.org/Subtitles
[2] *Analysis of the streaming text requirements*, MPEG, Shanghai, China, October 2002, n° M8931

(selecting a graphics track, playing while keeping synchronization, accessing randomly in the graphics stream, enabling progressive download and streaming or adaptive streaming, registering the track on top of the video …). An example of frame-based synchronization of SVG graphics with HTML 5 video content is available here[3].

4 Requirements

Based on the previous comment and additional use case, we believe MPEG should design future-proof technology for the carriage in ISOBMF of subtitles, or more generally for the carriage of timed data, potentially to be displayed synchronously on top of a video. Such a future-proof solution should have the following requirements:

- The ISOBMFF should be able to carry timed data, in a generic manner, for which the exact type or format can be identified.

Note 1: Such identification can be made for instance by MIME type, or by use of XML namespaces.

Example: It should be possible to carry SMPTE-TT content, WebVTT content, frame-based SVG content, frame-based HTML content …and to overlay the content on top of the video.

Note 2: We should not have to invent new boxes when a new format arrives, at least at the track level.

- The ISOBMFF should be able to carry samples of timed data composed of a main sample data referencing several individual pieces of data (sample resource), each of them carried efficiently, without requiring modifications to the main sample data.

Note: this requirement indicates that, for instance, if a JPEG is used by an XML description, the JPEG should not have to be base-64 encoded in the XML to be transported in the MP4 file and the XML should not have to be modified (URL).

- The ISOBMFF should be able store sample resources together with or separately of the main sample data, possibly using movie fragments.

Example: if a JPEG image is referenced by a unique subtitle sample, it should be possible to package them physically in contiguous bytes for efficient reading or to keep them in the same fragment. But if a JPEG is used by several samples, it might be useful to share that resource across samples and to store it separately in the file/fragment, for instance in an initialization segment in DASH or at the beginning of a fragment.

- The ISOBMFF should enable the storage of timed data in a fragmented manner across samples, for progressive loading by the application consuming sample data.

Note: for instance, in some cases, it is possible and more efficient[4] to fragment an XML document according to the time and to deliver the consecutive chunks of the XML document to

[3] http://concolato.wp.mines-telecom.fr/2012/06/18/html-5-video-and-svg-graphics-synchronization/

[4] C. Concolato, J. Le Feuvre and J. C. Moissinac, *Timed-fragmentation of SVG documents to control the playback memory usage*, ACM Symposium on Document Engineering, Winnipeg, Canada, August 2007, pp. 121-124, available at http://biblio.telecom-paristech.fr/cgi-bin/download.cgi?id=7129.

the parser at the sample time. This requirement ensures that it should be possible to store non-well formed XML chunks in the sample data.

5 Technical elements towards a solution

As indicated previously, the ISOBMFF already supports scene description and timed metadata tracks. The timed metadata tracks are generic enough and good candidates to fulfill our requirements but lack the indexing of resources. We propose two options to solve this problem, both rely on the use of the 'meta' box in movie fragments.

5.1 Usage of 'meta' box in movie fragments

The ISOBMFF defines the `MetaBox` which provides a useful mapping between a URL and a location in the file, using the `ItemInfoBox` and the `ItemLocationBox`. Additionally, it gives a way to protect the resources using the `ItemProtectionBox`.
However, such mechanism is not yet allowed in movie fragments. We propose to allow the use of 'meta' boxes in movie fragments, in particular in the `TrackFragmentBox` (at most one, and possibly one meco box to be consistent with the rest of the specification).

5.2 Option 1: Usage of timed metadata samples in movie fragments

In this option, we propose to carry synchronized subtitles or vector graphics using timed metadata tracks, where:

- The track handler is 'meta'
- The sample entry is a `MetaDataSampleEntry` box, more precisely:
    - an `XMLMetaDataSampleEntry` box when the content of the sample data needs to be identified as XML data such as TTML, SVG, …
    - a `TextMetaDataSampleEntry` when the sample data is textual data (such as WebVTT, HTML, …)
    - a `URIMetaSampleEntry` is some other cases.

The exact choice of the `MetaDataSampleEntry` box provides either a mime_type or a namespace to identify the exact type of data being carried. It also provides a content encoding.

We propose to specify the joint usage of timed metadata tracks and of the 'meta' box in fragments as follows:

Upon processing of data from a sample of a timed metadata track, if this data references a resource by URL, the URL shall be processed according to the 'iloc' box of a 'meta' (as defined in 8.11.3 and 8.11.9 of ISO/IEC 14496-12:2012, taking the meta box in the following order: first the meta box in the TrackFragmentBox, then in the TrackBox, then in the MovieBox and then at the file level.

This solution has the advantage of reusing existing standard structures, requiring very few modifications to the standard (enabling the use of existing boxes at new places).

The drawback is visible when flattening the fragmented file, i.e. when producing a non-fragmented file from the fragmented file. Since there can be only one 'meta' box at the track level, flattening requires some attention. Merging the 'meta' box (possibly resolving id conflicts) is not sufficient as different resources with the same name may be carried in different meta boxes (and referenced by bifferent samples); we suggest using a 'meco' box to store the additional

‘meta’ boxes at the track level, and defining a new sampleGroup type associating samples to ‘meta’ box indexes in the ‘meco’ box.

```
abstract class MetaSampleGroupEntry (unsigned int(32) grouping_type) extends
SampleGroupDescriptionEntry (grouping_type) { }


class MetaIndexMecoEntry() extends MetaSampleGroupEntry (’miim’) {
   unsigned int(32) index;
}
```

5.3 Option 2: Usage of ‘meta’ box as samples

To remove the possibly problematic flattening, another option is to define a new generic sample entry with the ‘meta’ handler type and the ‘metb’ coding name as follows:

```
class MetaBoxSampleEntry() extends MetadataSampleEntry(‘metb’) {
   string namespace;
   string content_encoding; // optional
   string schema_location; // optional
   string mime_type;       // optional
   BitRateBox();           // optional
}
```

When this sample entry is used, the content of a sample is a ‘meta’ box, where the primary item is the document (XML or textual) and upon processing of a resource URL, the content of the ‘meta’ boxes are analyzed in the following order: the ‘meta’ in the sample, the ‘meta’ box in the track fragment, the ‘meta’ box in the track; in the movie and finally in the file.

This approach has the advantage of requiring no specific operations in the flattening process. It has the drawback to require another sample entry and another sample format.

5.4 Miscellaneous points

There are additional technical points from N12644 that we would like to discuss here.

5.4.1 Time mapping

The WD mentions a requirement about “time mapping”. It is not clear when this requirement is needed. Is it envisaged, for instance, in TTML when the timeBase attribute is set to GPS or UTC, or only in local?

Additionally, it speaks about “adjacent time ranges”. It is not clear what this means. The term is not defined in the SMPTE specification.)

5.4.2 Spatial registration

Defining an area, with respect to the video area, where the text/graphics are to be displayed is one thing. This is already provided by the track header attribute, even if there is some ambiguity in case of multiple videos.

Exposing/duplicating the values of width and height of the text/graphics file is another, and might not be needed. The SVG standard solves that by using the notion of viewport/viewbox

negotiation, where the width/height is given by the application displaying an SVG, and if not provided the SVG file provides fallback values.

5.4.3 Document fragments

The WD mentions the following requirement:
"• Support fragmented subtitle tracks that store multiple documents sequenced on the presentation timeline to enable limiting the document, movie fragment, and Segment size to spread the bandwidth demand over the presentation duration."

This is not clear to us. In particular, we would like to make sure that incomplete document (document fragments) can be streamed to progressive parsers. With the solution aboves this can be done using the non-XML version of the delivery (TextSample with an XML MIME type).

5.4.4 Timescale and restrictions

The current WD imposes unnecessary restrictions on the timed track. Such restrictions are:

- On the timescale: to be the same as the video
- On the track flags (`track_enabled`, `track_in_movie`, and `track_in_preview`)
- On the track run flags

We think it is up to the authoring system to set them depending on the application. There should not be a deviation from standard ISOBMFF behavior here.

6 Conclusion

As described in this contribution, storing timed text within the ISOBMFF to enable efficient playback, positioning over and synchronization with video, streaming (in particular over HTTP) is restriction of the more general problem of storing scene descriptions with explicit positioning of the scene on top of the video (as opposed to the scene controlling the position of the video, the default behavior of BIFS and LASeR).

We support most of the use cases and requirements in the current WD but would like to include support for storage of video-synchronized graphics overlay.

In terms of technical solutions, we would like MPEG to reuse existing solutions (scene description tracks or timed metadata tracks), possibly amended or corrected, as presented, rather than define duplicate ones.