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/M26901
October 2012, Shanghai, China

Source **Telecom ParisTech**
Status **For consideration at the 102nd MPEG meeting**
Title **Comments on Carriage of Timed Text**
Author Cyril Concolato, Jean Le Feuvre

1 Introduction

This contribution presents some comments on the amendment of 14496-12 regarding the carriage of Timed Text and on the WD of the new part of MPEG-4 regarding the carriage of TTML and WebVTT.

The general concern we have is that MP4 tools (packager, players) will have to deal with many subtitle formats (3GPP Timed Text, TTML, WebVTT and more). We want to harmonize behaviors as much as possible and avoid deep format-specific processing. In particular, we would like that text stream fragmentation does not require processing other than regular video and audio fragmentation.

2 Comments on WD of 14496-12

2.1 Align the use of "text" vs. "timed text"

Sometimes the WD says "text" and sometimes "timed text" such as in:
"The text media type presents only text."
and
" 'text' Timed text track " .
Only one term should be used.

2.2 Media Header for text tracks

What should be the media header for text tracks? nmhd?

2.3 Clarify the use of media types

2.3.1 Subtitles as images

The WD should not exclude subtitles encoded as bitmaps, even if they are not accessible, even if they are not trendy anymore. It is an important source of subtitling data (DVD, DVB subtitles …), easy to carry in MP4. For instance carriage of DVB subtitles in MP4 is already possible with "Nero Digital Subpicture", unfortunately this is not standard nor registered. This WD could make this carriage standard.

2.3.2 Usage of handler vs. coding formats

In general, the use of text and subtitles media types is not clear in the WD. The WD says:

"The text media type presents only text. The sub-title media type presents text and possibly other media, including particularly images."

This can be read (in particular because of the use of 'presents') as:

- for DVB subtitles (presenting text but stored as images), one can use the "text" media type.
- for WebVTT, one can use "text" or "subtitle". WebVTT can be used to present text only but also to carry anything from JavaScript to HTML with graphics.
- for SMPTE-TT with images, one must use "subtitle" but for SMPTE-TT carrying only text, one could use "text".
- for SVG or HTML content presenting only text, one can use the "text" media type.
- for SVG or HTML content presenting text with some additional graphics, one would have to use the "sub-title" media type (e.g. audio description such as in http://joeclark.org/access/crtc/CRTC-2008/reply/images/CCfoto-CBC-CC-ST-ManWithoutaFace.jpg with the musical notation).

Additionally, it is not clear if the stream carries only graphics (HTML or SVG or else) what type should be used. We believe we should have an additional handler type 'grap', with the same sample entry, for general graphics overlay whether it contains text or not.

In general, the use of media type should not be driven by the format of the media but by the author's intent for the media. If an image is meant to be displayed as subtitles, it should be tagged as subtitle.

What matters to the application is whether or not this track will produce/present subtitles or not, so as to offer the choice to the user. Players can tell if they support a format of subtitle and the associated rendering by looking at the mime type/namespace. Users don't care what format is used to present the subtitles as long as accessibility properties can be indicated, and that can be inferred from the mime type.

We suggest defining the handler types as follows:

- 'subt': this media type shall be used when the content once rendered on screen is human readable with no graphics

NOTE: this does not mean that the content is encoded as text. This can be encoded as images. This does not mean either that it cannot be XML. One can use SVG, HTML, TTML, TTML with images to display human readable text.

- 'grap': this media type shall be used when rendered content contains only non-human-readable content such as graphics annotations

NOTE: This can be used to store timed graphical annotations of a video (such as here: http://liris.cnrs.fr/advene/screencasts/advene-svg-overlay.ogv).

As a consequence, the use of the metadata track type is left for non-presentable/non-displayable data. However, for coding purposes (to reuse existing code and standards), we propose to reuse the same tools (metadatasample entry) for all media types[1].

Proposed changes to WD of Carriage of Timed Text

Change title
Carriage of Timed Text and other overlayable data other than video

In section 8.4.3.1, change the following paragraph as indicated
There is a general handler for ~~metadata~~ streams of any type other than audio and video; the specific format is identified by the sample entry, as for video or audio, for example. If metadata, sub-title, graphics or text streams samples are plain text, then a MIME format is supplied to document their format, and each sample is a valid document of that type; if the samples are in XML, then each sample is a complete XML document, and the namespace of the XML is also supplied.

Change the following text
Which type of sample entry form is used is determined by the media handler:

- video track VisualSampleEntry
- audio track AudioSampleEntry
- metadata track MetaDataSampleEntry
- ~~text track TextSampleEntry~~
- subtitle track MetaDataSampleEntry~~SubtitleSampleEntry~~
- graphics track MetaDataSampleEntry

- Hint tracks an entry format specific to their protocol, with an appropriate name.

Remove the definitions of:

- PlainTextSampleEntry
- SubtitleSampleEntry
- XMLSubtitleSampleEntry
- TextSubtitleSampleEntry

And reuse MetadataSampleEntry

```
      case 'subt': // for tracks displaying readable text
      case 'grap': // for tracks displaying graphics (possibly text)
      case 'meta': // Metadata track (for other content using the
MetadataSampleEntry)
            MetadataSampleEntry();
            break;
```

Remove the definitions of:
SubtitleSampleGroupEntry
TextSampleGroupEntry
And introduce a unique definition for:
MetadataSampleGroupEntry

[1] This does not mean that 'presentable data' = 'metadata'. It is just a coding shortcut.

2.4 Entire documents or not

The current WD proposes the following change in section 8.4.3.1:

"If metadata, sub-title, or text streams samples are plain text, then a MIME format is supplied to document their format, and each sample is a valid document of that type ".

The added part is a backward incompatible change. When using a text track to store a time-fragmented text file, one may not want each individual sample to conform to the format, but only the concatenation of all samples between the previous RAP (including it) and that sample (progressive loading).

We suggest including some flag to indicate when samples are full-documents, but the default value for the flag should keep the existing behavior, compatible with progressive loading.

3 Comments on WD of 14496-XX on TTML and WebVTT

3.1 Common behavior

The current WD defines two independent specifications and there is no common behavior. We suggest that some common behavior be explicitly specified in a general section. In particular, this section could contain the following points.

3.1.1 Layout

We propose reusing the track layout à la 3GPP Timed Text tracks for both WebVTT and TTML tracks and in general for other overlay rendering (e.g. 3GPP Timed Graphics).

3.1.2 Timing

We propose to define that the common behavior of all Timed Text tracks be that:

- samples should not overlap in time
- empty samples should use empty documents (empty TTML documents or empty WebVTT cue).
- And that a unique timing model is used for all timed text tracks.

This last point implies that the use of relative or absolute timestamps in WebVTT and TTML should be harmonized. They are currently different. The pros/cons of both approaches are as follows:

	Relative to the track start (TTML-MP4 Approach)	**Relative to the sample start (WebVTT-MP4 Approach)**
• Playback • Import • Export • Editing of embedded timing values	No adjustment needed	**Needs adjustment.** Each time a sample is read or exported, the sample needs to be rewritten to add the sample CTS to the time values. Upon import, deep parsing is needed to ajdust embedded timing values.

• Seeking • Track timescale editing	No specific action required	
• Track timestamp editing • Adding / removing samples	For both cases, such editing **can potentially create overlapping samples or gaps** and may need further operations. It can be argued that timeline editing will not be done at the MP4 level but content will be exported, edited and imported back	
	Needs adjustment Changing the MP4 sample timestamps requires editing the start and duration values in the sample content.	No timestamp adjustment needed.
• Fragmentation (splitting 1 sample into 2, generating a RAP in-between existing samples)	**Needs adjustment** Creating a new redundant sample means copying the previous one and requires adjusting the timestamps (not the internal timing values as they are relative to track)[2]	**Needs adjustment** Creating a new redundant sample means copying the previous one and requires adjusting the internal timing values which are relative to the sample time
• Splicing (E.g. effect of ad-insertion within a DASH-period)	**Needs adjustment** All additional samples need to be rewritten to adjust timestamps and timing values	No timestamp adjustment

Editing does not seem to be a discriminating point as both approaches require some adjustment. Fragmentation is problematic with both approaches. Only splicing within a single DASH Period seems simpler with the WebVTT-MP4 Timing approach.

3.2 Changes to TTML carriage

We propose that a "meta" box is used to carry resources associated to a timed text sample.
This has the following advantages:

- Reuse of existing standard tools (easier specification work, conformance, reference software)
- Easy importing of TTML files as no XML rewriting is required (no need to know the SubSegment index when writing the XML)
- Ability to define a MIME type per resource (images, fonts, …)
- Ability to adjust resource position on the file if reused between samples
- Ability to indicate possible encryption per resource

[2] See additional contribution m26900 "On fragmentation of long-lasting samples"

3.3 Changes to WebVTT carriage

We propose to carry WebVTT content as plain text without using the proposed boxes (possibly using relative times depending on the result of discussion about 3.1.2), to clearly separate the coding layer from the transport layer. The current carriage of WebVTT as specified in the WD has the following problems:

- General problems
  - The WebVTT syntax is not frozen. Definition of headers, styling, key/value pairs are being discussed. Defining a Box-based syntax is risky!
  - The set of valid WebVTT files is not the same as the set of Parseable WebVTT files. Parseable files can be invalid but treated meaningfully by the browser. The HTML5 specification defines a parsing algorithm that discards invalid data while preserving playback. This is a future-proof parsing algorithm. The parsing algorithm for importing WebVTT in MP4 is not specified. In other words, we should allow storage of parseable but non-valid WebVTT files, to be future proof, and let the browser defines the appropriate behavior.

- Specific problems
  - The current box-based syntax does not define where content not in cues but in-between cues is stored (multiple sample description entries?).
  - The current WD does not define how to preserve layout/display order of cues when using continuation cue boxes (see http://lists.w3.org/Archives/Public/public-texttracks/2012Sep/0067.html for more explanation).
  - The current WD relies on continuation boxes to solve RAP for instance. This is not strictly needed as overlapping cues can be dealt with prior to packaging (see http://concolato.wp.mines-telecom.fr/2012/09/12/webvtt-streaming/)
  - The current WD does not provide the flexibility to represent more than 1 cues in one sample.

The proposed approach is simply to store the cue text (including any non-cue text between this cue and the previous one) as a sample.

The advantages of this approach are:

- Reuse of existing standard tools (easier specification work, conformance, reference software)
- All functionalities of boxes are covered (RAP properties…) without new risky box syntax.