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/M29240
April 2013, Incheon (KR)

Source **Telecom ParisTech**
Status **For consideration at the 104nd MPEG meeting**
Title **Editorial and technical comments on carriage of timed text and visual overlays in MP4**
Author Cyril Concolato, Jean Le Feuvre

1 Introduction

This contribution proposes a review of the current DIS of MPEG-4 Part 30 with both editorial and technical comments. Some of the comments result from implementation of the support for WebVTT in MP4Box as reported in contribution m29230.

Editorial comments are proposed as a separate document with revision tracking. Technical comments are proposed herein.

2 Technical comments on WebVTT

2.1 Timeline

The WebVTT specification is defined by a Community Group within W3C, not by a Working Group. The ISO standard won't be able to progress to IS until the WebVTT specification has been transferred from the CG to the WG and until it has reached Proposed Recommendation status. Even though there is pressure in some countries, such as the USA, to provide a solution for captioning on the Web, and the WebVTT may be a solution for that, MPEG should not hurry to reach IS stage since modifications on the WebVTT recommendations may affect its carriage on ISOBMFF and require a corrigendum.
Recommandation: do not progress to IS at the next meeting or split the WebVTT part as an amendment to a part 30 containing only the core and TTML parts.

2.2 Parser behavior

As indicated in contribution m26901 (Shanghai, October 2012), the normative algorithm for a WebVTT parser[1] is resilient to files which differ from a WebVTT file conformant to the syntax[2]. For instance, the following WebVTT files contains more elements at odd places that the syntax does not permit, but is perfectly parseable.

[1] http://dev.w3.org/html5/webvtt/#parsing
[2] http://dev.w3.org/html5/webvtt/#syntax

```
WEBVTT – this is text after the signature (not syntax compliant)
This is a header
on three lines
headers are not specifically mentioned in the syntax

This line is not in the header, but also parseable but not
mentioned in the syntax

00:11.000 --> 00:13.000
We are in New York City

This line is in between cue and is not mentioned in the syntax

00:13.000 --> 00:16.000
<v Roger Bingham>We're actually at the Lucern Hotel, just down
the street

NOTE – This is a comment, mentioned in the updated WebVTT
syntax, but not carried in MP4 yet

00:16.000 --> 00:18.000
<v Roger Bingham>from the American Museum of Natural History

00:18.000 --> 00:20.000
<v Roger Bingham>And with me is Neil deGrasse Tyson
```

We think that the intention of the carriage of WebVTT in ISOBMF is as follows (depicted in Figure 1): the result of the parsing of a given (not-necessarily conformant) WebVTT file by a conformant WebVTT parser should be the same as the result of the parsing by a conformant WebVTT parser of the same file after a round-trip in an ISOBM file.

Note that this comparison is different from doing a string comparison on WebVTT files 1 and 2 as they might differ on many aspects such as empty lines, discarded text....

However, since the output of a WebVTT parser in not defined in terms of conformance points and is likely to change (e.g. text on the signature line, header lines, comments or in-between cues lines are currently discarded), the ISOBMF round-trip should be as conservative as possible. We recommend clarifying that in the specification. This drives the following comments.

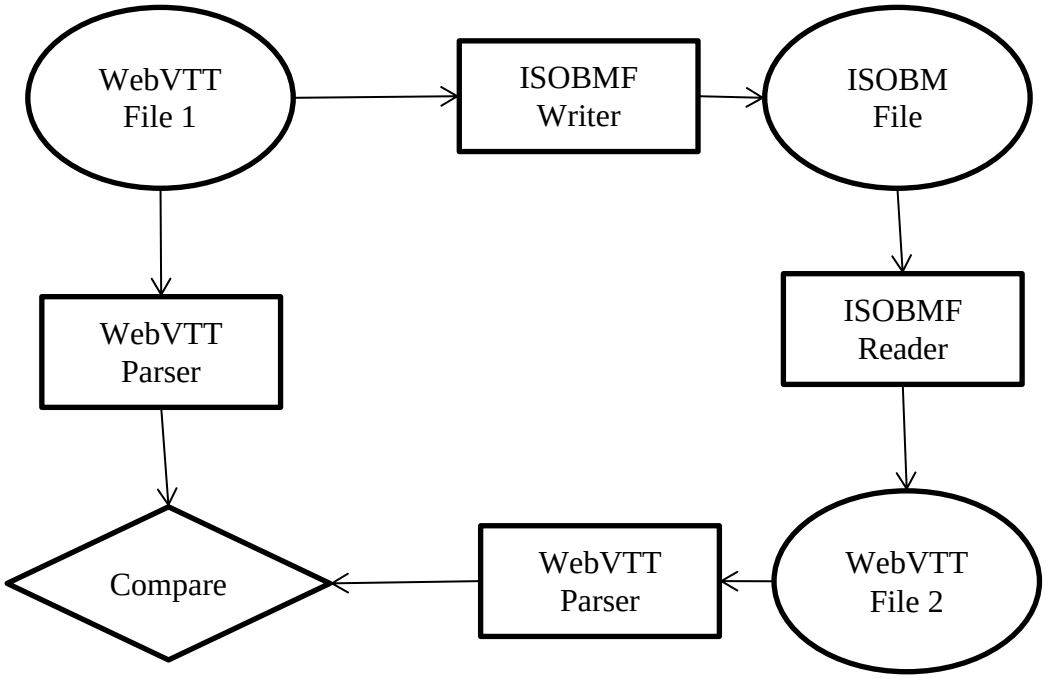

Figure 1 - Round-Trip equivalence of WebVTT files in ISOBM files

2.2.1 Comments and in between cue-text

We recommend introducing a new box called VTTAdditonalTextBox as follows:

```
class VTTAdditionalTextBox extends Box('vtta') {
   boxstring   cue_additional_text;
}
```

And modify the text as follows:
"Each sample is either:
(a) exactly one VTTEmptyCueBox box (representing a period of non-zero duration in which there is no cue data) or
(b) one or more VTTCueBox boxes that share the same start time and end time, each containing the following boxes. Only the CuePayloadBox is mandatory, all others are optional, or
(c) zero or more VTTAdditionalTextBox boxes, interleaved between VTTCueBox boxes and carrying text in between cues, in the same order as in the input file."

2.2.2 Signature, Header and SampleEntry

The handling of signature and handler is not clear in the specification.

We recommend clarifying that the config string in the sample entry carries both the signature line and all header lines.

2.2.3 Handling of carriage return, line feed and null chars

The parsing algorithm of WebVTT files as specified in the W3C specification has special handling for U+000D CARRIAGE RETURN, U+000A LINE FEED and U+0000 NULL characters as follows:

- Replace all U+0000 NULL characters by U+FFFD REPLACEMENT CHARACTERs.
- Replace each U+000D CARRIAGE RETURN U+000A LINE FEED (CRLF) character pair by a single U+000A LINE FEED (LF) character.
- Replace all remaining U+000D CARRIAGE RETURN characters by U+000A LINE FEED (LF) characters.

It is not clear from the ISO draft standard if these behaviors should be applied prior to the storing the text in the ISO file.

As in the model of Figure 1, we suggest saying "ISOBMFF packagers may apply character replacements as specified in step 1 of the WebVTT parsing algorithm, prior to storing the strings in the ISOBMFF constructs. ISOBMFF readers should be prepared to apply these replacements if integrated directly with a WebVTT renderer."

It is not clear if the CueIDBox, the CueSettingsBox and the CuePayloadBox should contain the trailing LF characters.

We suggest:

- Trailing CR, LF, or CRLF characters in CueIDBox and CueSettingsBox boxes shall not be stored in the corresponding boxstring.
- Trailing CR, LF or CRLF at end of the signature line, at the end of each header lines, and at the end of the cue payload lines shall be stored in the respective box strings.

It is not clear if the CueIDBox, the CueSettingsBox and the CuePayloadBox should contain the trailing LF characters.

We suggest:

- The leading space separating the timings from the settings shall not be stored in the CueSettings box.

In the current implementation, MP4Box:

  - Does not replace U+0000 NULL characters by U+FFFD REPLACEMENT CHARACTERs, NULL characters are not handled;
  - Does not replace CRLF or CR by LF; and

Boxes such as CueSourceIDBox, CueLocalIDBox, CueEndTimeBox, CueStartTimeBox are not supported, as they are not needed.

The CueTimeBox is supported in reading/writing but not generated during parsing nor exploited during export, yet.

2.3 Decoder Model

With the use of ISOBMF, the decoding model in a WebVTT player can be one of the 3 vertical options depicted in Figure 2: playing VTT files (including WebVTT segments), playing ISOBM files (or segments), going through a serialization to WebVTT or not.

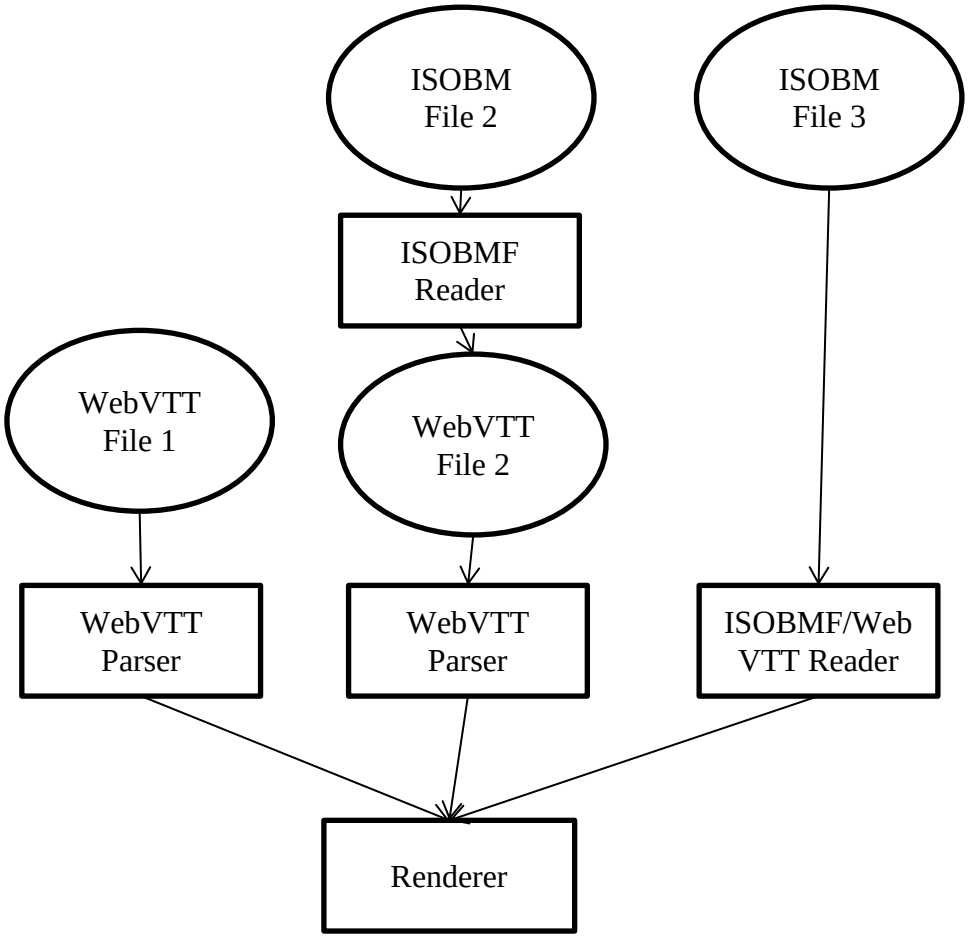

Figure 2 - Decoder Model for WebVTT players with ISOBMFF support

2.4 Cue overlap and empty samples

The algorithm for splitting cues is complex and the processing of samples is not consistent with TTML. In TTML, durations may be included in the TTML document, while sample durations are only used to reconstruct a decode time. We recommend to follow the same approach and to store the cue duration in each cue (CueDurationBox) if it differs from the duration of the sample the cue is in. The CueDurationBox (if present) is then used to produce the cue end time, ignoring the duration as indicated in the ISO file.

The key points of this approach would be:

- Harmonized behavior with TTML carriage where documents have a duration and samples have a duration;
- Simplified handling overlapping cues. Only cues with the same start time but different end times need to be split. All cues with the same start and end time would be stored in the same sample.

- The sample duration would reflect the difference between cue start times and since start times are increasing in WebVTT files, sample duration would be positive.
- If the cue duration is smaller than the sample duration, this just means that there is a period with no cue being displayed. This is equivalent to having empty cues.
- Empty cues could be removed: an in-between empty cue is replaced by extending the previous sample duration; a first empty cue is replaced by an edit list.
- If the cue duration is greater than the sample duration: this just means the two cues overlap.
- Not all samples would be RAP. If RAPs are needed, cues can be split in the textual domain (similar to RAP being produced in the compressed domain for video) and then imported and if no overlap is detected (no backward frame reference in video coding), then the sample is a RAP.

3 SampleEntry

3.1 Text streams for HTML and SVG

The current specifications allow transporting different types of text streams, possibly HTML and SVG 'streams', but it is not clear if they should be transported as Metadata streams, Timed Text streams or Subtitle streams. This should be clarified in the spec.

Additionally, for those streams and in general a configuration stream may need to be carried when text formats are used, i.e. when samples do not represent whole documents, i.e. using the PlainTextSampleEntry or TextSubtitleSampleEntry.

Additionally, the current specs define the PlainTextSampleEntry class and the WVTTSampleEntry, as a sub-class. The configuration box is only present in the sub-class while it should be in the base class.

We propose to replace:

```
class TextSubtitleSampleEntry() extends SubtitleSampleEntry ('sbtt') {
   string   content_encoding; // optional
   string   mime_format;
   BitRateBox ();             // optional
}
```

By:

```
class TextSubtitleSampleEntry() extends SubtitleSampleEntry ('sbtt') {
   string   content_encoding; // optional
   string   mime_format;
   TextConfigurationBox config; // optional
   BitRateBox ();             // optional
}
```

And Replace:

```
class PlainTextSampleEntry(codingname) extends SampleEntry (codingname) {
}
class WebVTTConfigurationBox extends Box('vttC') {
  boxstring config;
}
class WVTTSampleEntry() extends PlainTextSampleEntry ('wvtt'){
  WebVTTConfigurationBox config;
```

```
  MPEG4BitRateBox (); // optional
}
```

By:

```
class TextConfigurationBox extends Box('texC') {
  boxstring config;
}
class PlainTextSampleEntry(codingname) extends SampleEntry (codingname) {
  string content_encoding; // optional
  string mime_format;
  TextConfigurationBox config;
  BitRateBox ();                    // optional
}
class WVTTSampleEntry() extends PlainTextSampleEntry ('wvtt'){
}
```

The following describes an example of carriage of SVG content as a text stream:

PlainTextConfiguration	<svg …> <defs> … some global elements …</defs>
Sample 1	<g id="frame1" visibility="hidden"> <rect id="aRect" fill="red"/> <set xlink:href="#aFrame" attributeName="visibility" to="visible" begin=1s"/> </g>
Sample 2	<g id="frame2" visibility="hidden"> <rect id="aRect" fill="red"/> <set xlink:href="#aFrame" attributeName="visibility" to="visible" begin=2s"/> </g>

3.2 Bucket media types

It has been raised in previous meetings that "The 'Codecs' and 'Profiles' Parameters for "Bucket" Media Types" Internet Draft needs to be updated to take into account new SampleEntry classes. This is in particular true for metadata, subtitle and timed text tracks.

Sample Entry	4CC	Proposed Bucket media type elements	Observation
MetaDataSampleEntry			
URIMetadataSampleEntry	urim	urim.<theURI>	
XMLMetaDataSampleEntry	metx	metx.<namespace>	
TextMetaDataSampleEntry	mett	mett.<mime_format>	What if the sub format has parameters?
SubtitleSampleEntry			
XMLSubtitleSampleEntry	stpp	stpp.<namespace>	
TextSubtitleSampleEntry	sbtt	sbtt.<mime_format>	
PlainTextSampleEntry			
WVTTSampleEntry	wvtt	wvtt	
TextSampleEntry	tx3g	tx3g	

???	text	text	Where is this SampleEntry defined?
DIMSSampleEntry	dims		
???	tigr	tigr	

4 Conclusion

We recommend producing a study of DIS to take these comments into account.