EBU – TECH 33??

EBU-TT Segmentation

Status: Draft for Comments

Version: 0.6

09/09/2013

Source: SP/MIM – XML Subtitles

Geneva [Publicationdate]

This page intentionally left blank

Conformance Notation

This document contains both normative text and informative text.

All text is normative except for that in the Introduction, Examples, any section explicitly labelled as 'Informative' or individual paragraphs which start with 'Note:'.

Normative text describes indispensable or mandatory elements. It contains the conformance keywords 'shall', 'should' or 'may', defined as follows:

'Shall' and 'shall not':	Indicate requirements to be followed strictly and from which no deviation is permitted in order to conform to the document.
'Should' and 'should not':	Indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others. OR indicate that a certain course of action is preferred but not necessarily required. OR indicate that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
'May' and 'need not':	Indicate a course of action permissible within the limits of the document.

Default identifies mandatory (in phrases containing "shall") or recommended (in phrases containing "should") presets that can, optionally, be overwritten by user action or supplemented with other options in advanced applications. Mandatory defaults must be supported. The support of recommended defaults is preferred, but not necessarily required.

Informative text is potentially helpful to the user, but it is not indispensable and it does not affect the normative text. Informative text does not contain any conformance keywords.

A conformant implementation is one which includes all mandatory provisions ('shall') and, if implemented, all recommended provisions ('should') as described. A conformant implementation need not implement optional provisions ('may') and need not implement them as described.

This page intentionally left blank

Contents

1 Introduction

1.1 Background

EBU-TT is the EBU's broadcast application domain variant of W3C TTML. Part 1 (EBU Tech 3350 Ref. [5]) specifies the XML based archiving and interchange format for subtitles as a follow-up to the EBU STL format. The implied assumption is that the duration described by a single EBU-TT part 1 document is equivalent to the entire duration of the programme to which its contents apply.

There are at least two use cases in which this assumption is not necessarily valid, broadly corresponding to live authoring and online distribution. In both of these cases a mechanism is required to describe small segments of subtitles applicable to a different, usually shorter, duration. These use cases may differ in other ways too, for example having different reference timescales or metadata.

Similarly, where subtitles are created live, there is a use case for archiving small segments of subtitles and combining them into a longer document that can more easily be stored and processed later.

1.2 Relation to other specifications

The contents of this document are intended to inform and be used within the forthcoming EBU-TT part 3 document specifying Live subtitling, and the forthcoming EBU-TT-D document specifying a distribution profile for subtitles.

EBU-TT-D will be based primarily on W3C TTML 1.0 (Second Edition) Ref. [1], therefore this document also relates its specifications directly to TTML. Note that, although it may be possible to apply the semantics described here to TTML, we have not attempted to verify that that this will work in the general case, only for those features expected to be used by EBU-TT-D and EBU-TT part 3. For example, the approach taken within this document has not been shown to work for TTML documents containing animations: to do so would require further work.

In developing this specification, reference was made to the relevant parts of the MPEG specifications published as the ISO/IEC 14496 ISO base media file format (ISO BMFF) Ref. [2] and ISO/IEC 23009 Dynamic adaptive streaming over HTTP (DASH) Ref. [3] to understand the core concepts applicable to the encoding and division of media objects.

1.3 Purpose of this document

This document is intended to define the rules for creating and manipulating EBU-TT samples. However it does not in itself define any EBU standard or specification: it is expected that EBU-TT Part 3 and EBU-TT-D will reference or include sections of this document, at which point they will become part of a specification.

1.4 Definition of terms

1.4.1 Media timeline

This is the timeline of the media to which the subtitles apply. It excludes any packaging offsets that may have been added for distribution purposes. It shall be a linearly increasing scale with a defined datum point, common across all media streams that require alignment. Examples of this include a SMPTE timecode on a pre-recorded programme, or a

'wall' clock assigned at a defined point in a broadcast workflow such as video frames leaving a video switching matrix.

1.4.2 Sample

The smallest unit of segmentation shall be known as a *sample*. This is intended to be equivalent to the meaning of sample within the ISO BMFF. The focus of this document is how to define an EBU-TT sample.

A sample is a valid EBU-TT document. Carriage mechanisms that are not based on ISO BMFF but that depend on segmentation may define other terminology, in which case the sample should correspond to the smallest temporal unit required.

1.4.3 Segments and Subsegments (Dash specific)

Each sample may occupy a temporal extent equivalent to a *Segment* or *Subsegment* or even a *Period* in Dash terminology, however this document does not require use of those terms.

1.4.4 Temporal extent

The period of time between the active begin and end of each sample in media time, i.e. the time period or window that defines which subtitles must be within the sample. This term has been chosen because it aligns with the term 'root temporal extent' in the TTML 1.0 specification.

Note that it is possible in some schemes for the earliest begin time and latest end time within a document to extend before and/or after the externally defined temporal extent, however processors are not expected to generate any intermediate synchronic documents corresponding to times outside this period.

2 Use cases

This section describes the use cases that have been considered and that are supported by this segmentation scheme.

Three specific use cases are presented initially, which are then generalised to more abstract segmentation requirements that must be solvable.

2.1 Use cases

2.1.1 Live contribution

A subtitler has access to programme content and creates a live stream of subtitles in real time during transmission of the programme. These subtitles are distributed to an encoder and packaged for distribution.

Note that the programme itself may be live or pre-recorded, or some mixture of the two, but the subtitles are effectively being created as live.

A key requirement[1] is that latency is minimised, so there shall be as little delay as possible between the subtitler creating a subtitle and it being made available to the audience.

In many broadcast environments the stream of live subtitles is inserted into the video signal so that both can be routed together e.g. to distribution encoders. This is an example of a uni-directional flow of subtitles that must be supportable. Thus the second requirement is that no 'back channel' must be required for the contribution of live subtitles to be effective.

A third requirement is therefore that the subtitle data is stateless, i.e. no knowledge of prior subtitle data 'samples' or 'instructions' is required to process or present current or future subtitles.

The requirements from this use case may be met by creating many samples of subtitles each applicable over a short duration. Additionally, it may be that no individual sample has a defined end time when it is authored: if the subtitler is unaware of what is about to happen next in the programme, it is difficult to be precise about when a subtitle should disappear.

2.1.2 Segmented Distribution

A broadcaster wishes to distribute programmes including subtitles using a scheme that delivers short sections (or segments) of media that are decoded and concatenated by the display device to appear continuously without interruption.

An example of this is an MPEG Dash stream.

The key requirement here is that there is defined alignment between the different media types being distributed together so that the desired synchronisation can be achieved, e.g. to avoid lip-sync errors between video and audio etc.

Considerations of latency and availability are not part of the requirements for the contents of the EBU-TT samples.

[1] This is a real world requirement of the architecture, rather than a segmentation requirement per se.

2.1.3 Archival

An automated system takes copies of all the EBU-TT samples that are used to create distributed subtitles over a given period, from a defined point in the broadcast architecture, and combines them into an EBU-TT part 1 document for archiving. The period and service may be defined for example by a playout automation system.

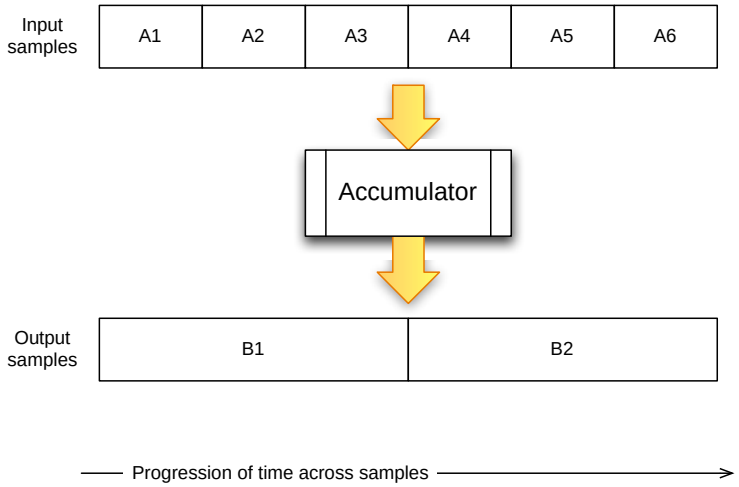
This is likely to be of particular interest for programmes whose subtitles are created live. Prepared subtitles are likely to be delivered in an archive and exchange format such as is defined in EBU-TT part 1 so completing a round trip may have little benefit for archive purposes – though there may be a requirement for creating an 'as broadcast' recording of subtitles for compliance audit purposes.

2.2 Generic requirements

As well as the requirements highlighted above, the following more abstract generic requirements need to be solved to allow the use cases from the previous section to be achieved.

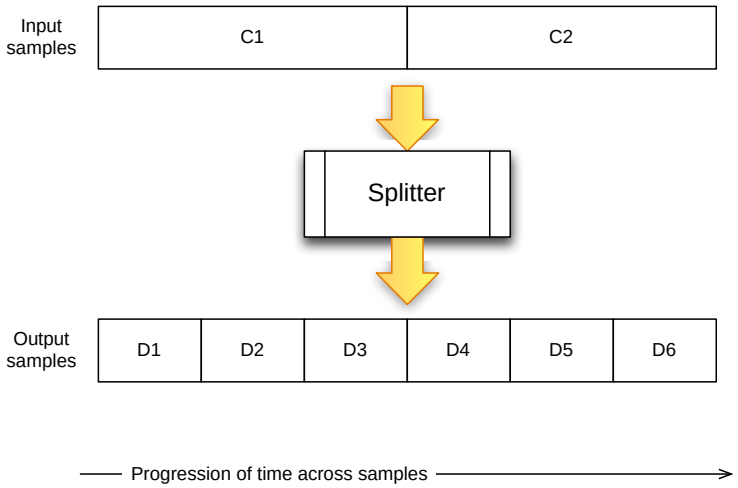
2.2.1 Accumulating short samples into longer ones

A processor combines a series or set of short samples into a smaller set of longer samples.



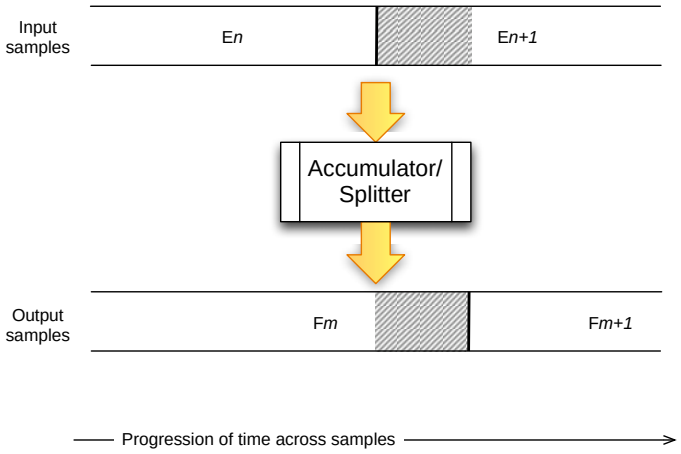
2.2.2 Splitting long samples into shorter ones

A processor divides a set of samples into a larger set of shorter samples.



2.2.3 Managing non-aligned boundary points

While accumulating or splitting samples the processor must be able to manage boundary points that do not align.



2.2.4 Sample ordering

A set of samples is captured and stored for later processing. The original order and timing of each sample must be preserved.

2.2.5 Clock overflow

The clock used to define the media timeline has a maximum value and periodically overflows back to zero, resulting in a sample whose end time is apparently less (or *earlier*) than its begin time. The architecture must either arrange for such a sample never to occur, or the processor that encounters this sample must be able to interpret the contents of the document correctly, e.g. by using heuristics such as ‘maximum normal duration of a sample’ etc.

2.3 Excluded use cases

The following use cases are out of scope of this document.

2.3.1 Reprocessing word by word subtitles into block subtitles

Where live subtitles are created word by word the archiving processor re-assembles the content into line by line (also known as 'block') subtitles. This is an automated editing process not directly related to segmentation and is out of scope of this document.

2.3.2 Live subtitles are re-aligned with the source audio and video

The problem of time alignment, i.e. attempting to present live subtitles at the time in the media that is closest to when the subtitled words were spoken, is orthogonal to the segmentation problem and is out of scope of this document.

2.3.3 Sample re-timing

There are, or are likely to be, some real world use cases that require processors to receive samples of EBU-TT with one reference timeline and create samples of EBU-TT with a different reference timeline, for instance as part of a distribution encoding architecture. Such re-timing is out of scope of this document, which assumes that a common timeline is available to hypothetical processors for both their inputs and their outputs.

2.3.4 Restricting sample complexity and data rates

A carriage mechanism or processing or rendering device has limited capabilities and this imposes constraints e.g. on data rates, document complexity etc. To balance the variables that impact these constraints it may be necessary to modify aspects of the segmentation e.g. the duration of each sample, but this is orthogonal to the general problem of segmentation in itself and is out of scope of this document.

3 Assumptions, simplifications and constraints

This section describes the assumptions and simplifications made, and constraints applied, and within the context of which the segmentation semantics shall be understood.

3.1 Sample re-ordering is not needed

There's no compression, coding or decoding advantage in deliberately sending out of order samples for subtitles. If the carriage mechanism for each sample may deliver samples out of order, it is expected that the carriage mechanism will have sufficient information to re-order samples itself.

However implementations may add metadata within each sample to assist with re-ordering as an off-line process, e.g. to allow live samples to be stored and converted to archive EBU-TT part 1 documents later.

3.2 The root temporal extent of each sample is externally defined

It is a requirement that any presentation or processing system shall be aware of the temporal period during which each sample is active, as defined externally to the data within each sample. These define, in TTML terms, the *Root Temporal Extent* of the `body`.

To be more specific, W3C TTML includes the following definition:

> **Root Temporal Extent**
> The temporal extent (interval) defined by the temporal beginning and ending of a *Document Instance* in relationship with some external application or presentation context.

That relationship is defined within this document to be as follows:

- The external presentation or processing system fully defines the period during which the sample is active.
- The time coordinates used within `begin` and `end` attributes should be with reference to this external period.
- Only contents (or intermediate synchronic documents) that are active within the externally defined period should be displayed; the rest should be disregarded by any presentation processor.

The options for `ttp:timeBase` and `ttp:markerMode` are not explored here but do not modify this rule.

The rationale for this approach is that the architecture and environment to support video publication with subtitles is expected to exert a higher level of control, i.e. to be able to decide exactly which EBU-TT sample is 'active' at any given moment. Even if an EBU-TT sample contains subtitles timed to be on-screen at a particular time, if the whole sample is inactive at that time, the subtitles will not appear.

For example, if the sample is being carried within the VANC in an HD-SDI signal, then the timing semantics must be known to the processor. Alternatively, if the sample is being distributed over HTTP and is referenced by a Dash MPD, then the packaging defines the media timeline for each sample. This is true even if the same programme content is made available with different adjacent content in different Periods[2] perhaps because different advertisements are being provided based on audience characteristics, time of day etc.

[2] *Period* is a Dash term used for example purposes only. The concepts defined in this document do not reference or use Period directly.

3.3 There is an externally defined media timeline

EBU-TT documents, or TTML documents carrying subtitles, are intended usually for display such that individual subtitles are synchronised to some media timeline. We assume that this media timeline is defined externally to the EBU-TT documents that are segmented and that it is possible for the authoring system or other processing to create timestamps (begin and end attributes) that are within this timeline.

The mechanism for this external definition of the timeline is strictly out of scope of this document. It may be imagined that various schemes could exist, such as knowledge of the frame that each sample is attached to, an explicit definition within a Dash MPD, etc.

It is possible, and indeed probable, that in real world architectures a variety of different timelines may co-exist simultaneously, dependent on the media distribution mechanism in use, and the set of supported applications. Our assumption is that this external context is able to identify the single relevant media timeline for subtitle timing.

3.4 The media timeline is a linearly increasing scale

The media timeline is assumed to be a linearly increasing scale rather than event based, not necessarily so that calculations can be performed on durations and end times but rather so that, if a presentation or other system needs to 'seek' to a specific point in the media and find the subtitle that is appropriate to a single frame, then this can be done without hunting backwards for the most recent event.

This also is a requirement to allow a deterministic algorithm to map between different time expression types, e.g. from ticks at a given rate to milliseconds.

Note that in some architectures linearly increasing timelines are permitted to contain discontinuities[3]. This makes it impossible to calculate durations by subtracting begin times from end times but still permits random access seeking, as long as the seek algorithm simply compares target time to begin and end times, i.e. if begin<=target<end for any given subtitle or sample then it shall be active.

[3] See DVB Specification for the carriage of synchronized auxiliary data in DVB transport streams (Ref. [6]), for an example of this.

4 Segmentation semantics

This section defines the semantics for segmented EBU-TT, and the rules governing how EBU-TT shall be split up and accumulated as necessary to form EBU-TT documents of different durations.

4.1 Axioms

4.1.1 Every sample shall be a complete valid document

Each sample shall be a complete, valid EBU-TT document, containing everything needed for displaying the subtitles that occur within its temporal extent, such as layout in regions, styling, metadata etc. Note that the definition of the temporal extent for the sample is provided externally[4].

4.1.2 In the processing and presentation of segmented EBU-TT a maximum of 1 document can be 'active' for any given moment

It is possible to create a set of EBU-TT samples that have overlapping temporal extents as defined by the range [earliest `begin` to latest `end`] *internally* within each sample. This apparent conflict is partially resolved by our statement that the actual temporal extents are defined *externally*. However a further constraint is helpful, namely that at any single moment in media time the processing system must hold exactly zero or one EBU-TT documents as 'live', i.e. in operation for the purpose of defining the subtitles.

There are (at least) two mechanisms by which this can be achieved. The first is to define the period of each sample such that no two samples overlap in time. This works well in a prepared subtitle environment when the duration of each sample is predetermined.

The second is to define a rule when receiving a stream of samples when the arrival time of the next sample is unknown, as may be the case in a live subtitling environment. This rule states that on receipt of a sample all previously received samples become inactive.

This second rule supports the live use case for example permitting the equivalent of a 'clear' of all subtitles by issuing a document with no subtitles in the body.

4.1.3 The mapping from externally defined times to begin and end times within each sample shall be defined and deterministic

It shall be possible to translate an externally defined media time to a `begin` or `end` time within each sample. It must also be possible to translate a `begin` or `end` time to the corresponding media time.

The precise rules for this translation cannot be defined in the general scope of this document and are in fact defined informatively within W3C TTML. However typically there are two aspects that must be defined.

Firstly the relative rate at which each of the time scales changes must be known and constant. For example if the media time is expressed in frames and the subtitle sample time is expressed in hh:mm:ss.ms then there shall be exactly one 'frame rate' that defines how many frames pass per second in the media.

[4] In the general case, it is to be expected that other information might also need to be provided externally but this is out of scope of this document.

Secondly there must be some kind of datum point to act as a common reference time where it is known that a value in one timeline is equal to a value in the other timeline. This might conveniently be expressed as a 'zero' time in one timeline or the other, but this is not a requirement.

The mapping shall, in general, be defined externally to each sample, however in most schemes the rate of change is related to real world clock units, i.e. seconds, so this may be implied rather than explicit. For instance the Dash `@timescale` attribute describes the number of units per second.

4.1.4 Untimed EBU-TT content applies to the entire duration of the sample as externally defined

This is equivalent to the default behaviour of TTML for documents with unspecified `begin`, `end` and `timeContainer`[5] attributes. It results in all content within the document being displayed for the entire active duration of the sample.

It may be useful to create untimed EBU-TT samples in architectures that issue a new sample for every change in the subtitles that appear on screen, or that create a separate EBU-TT sample for every frame of video. These architectures may be present in broadcast environments designed to support both pre-recorded and live programmes.

4.1.5 The temporal extent for the display of each sample shall be clipped

The clipping behaviour described within TTML 1 7.1.3 applies:

> "If the *Root Temporal Extent* is shorter than the computed duration of the `body` element, then the active time interval of a `body` element is truncated to the active end point of the *Root Temporal Extent*."

This means that any content within the sample's temporal extent shall be displayed as normal.

Any content outside the sample's temporal extent is meaningless and shall not be displayed.

There are potential applications for deliberately extending a sample's contents beyond its temporal extent. For example when splitting a prepared 'whole programme' EBU-TT document for routing with the video in a 'one sample per frame' architecture then a simple approach may be to search the entire prepared document for all elements that are active during the frame in question and reproduce them with minimal processing, so that for several frames at a time when there is no subtitle transition all of the samples are identical[6].

One may alternatively consider that modifying the `begin` and `end` times within each sample such that they do not fall outside of the sample's temporal extent might create samples that are in some way better, e.g. more interoperable, more elegant, more robust etc. Neither approach is preferred in this document.

[5] i.e. the `timeContainer` attribute has the initial value `par` and the implicit duration is indefinite as defined in SMIL 2.1 (Ref. [4])

[6] This approach does require that the external media time is correctly aligned to the times within each sample.

4.1.6 Any individual subtitle must be present within all the samples during which it is required to appear

If a subtitle needs to appear within timings that overlap one or more sample boundaries then the representation of that subtitle must be contained within every sample in which it occurs. Note that the term 'subtitle' is loosely defined and in this instance effectively means a piece of text with a computed style and location.

Furthermore, implementations shall provide continuity of appearance of such a subtitle across sample boundaries, as defined in the intermediate synchronic document. In other words, even if the way that the subtitle text is defined within two consecutive documents is different, e.g. in the IDs of the `region` or `style`, or in the `begin` and `end` times, presentation systems shall not introduce any flickering in the rendered output.

It is beyond the scope of this document to define optimisation techniques for use in implementations, however it should be noted that in some architectures implementations may be assisted in meeting this requirement by the way that subtitles are defined and identified across sample boundaries. For example it is not a requirement to re-use the same `<layout>` and `<styling>` elements within the `<head>`, and the same structure and identifiers within the `<body>`, but it is permitted to do so. Note that XML identifiers are only required to be unique within a single document instance. If, in a constrained architecture, an additional set of rules is imposed to manage this information for rendering optimisation[7], then this will add requirements to any device that splits or accumulates documents.

4.2 Splitting and accumulating documents

This section is an early draft to describe the common expected behaviours of a theoretical Splitter or Accumulator: it does not reflect EBU-TT conversations however some may consider it to obvious. Others may consider it, including the diagrams, to be useful background to assist with understanding the rest of the document.

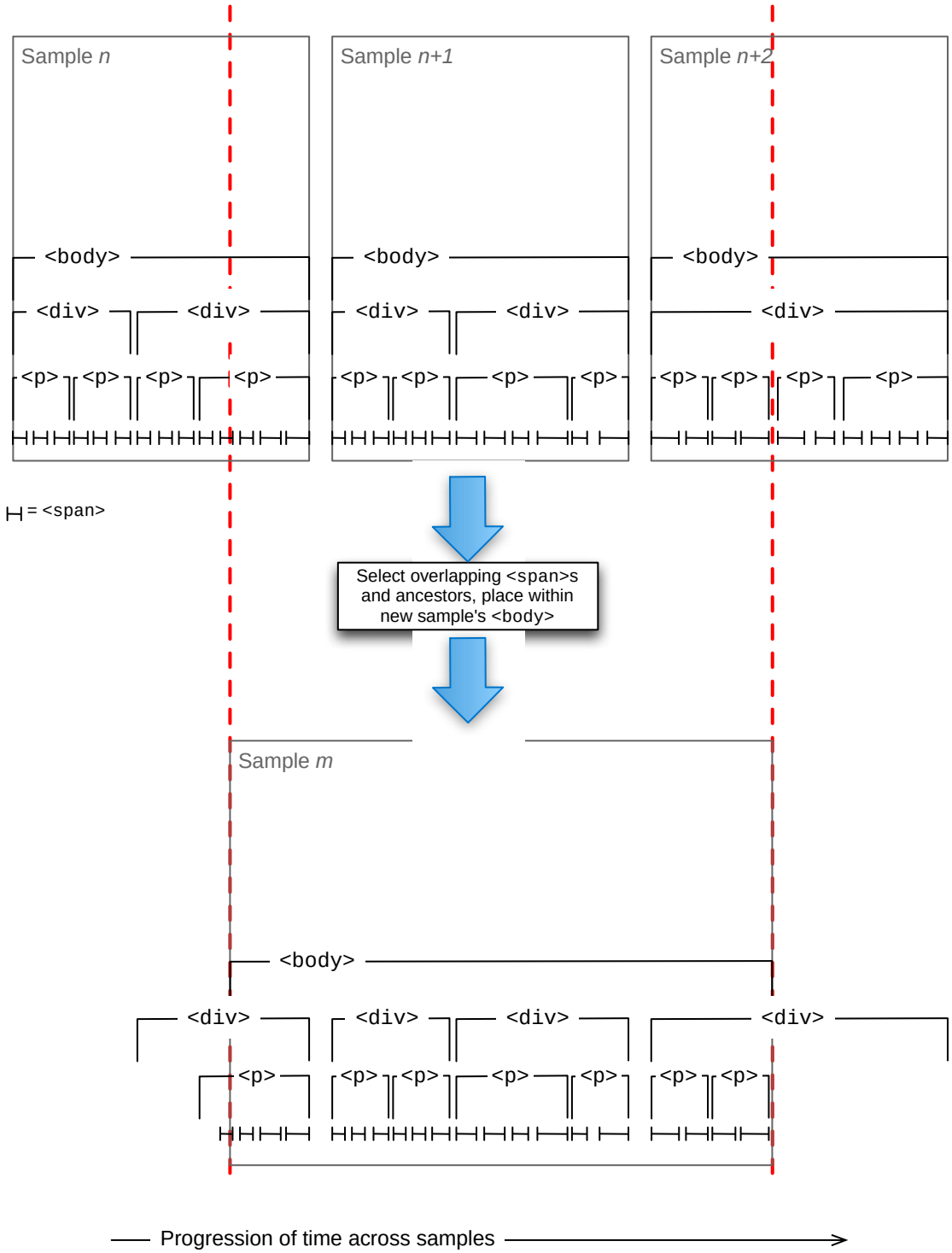
Following discussion of the topic this section should be updated. To reflect this status the text has been coloured in a paler shade.

[

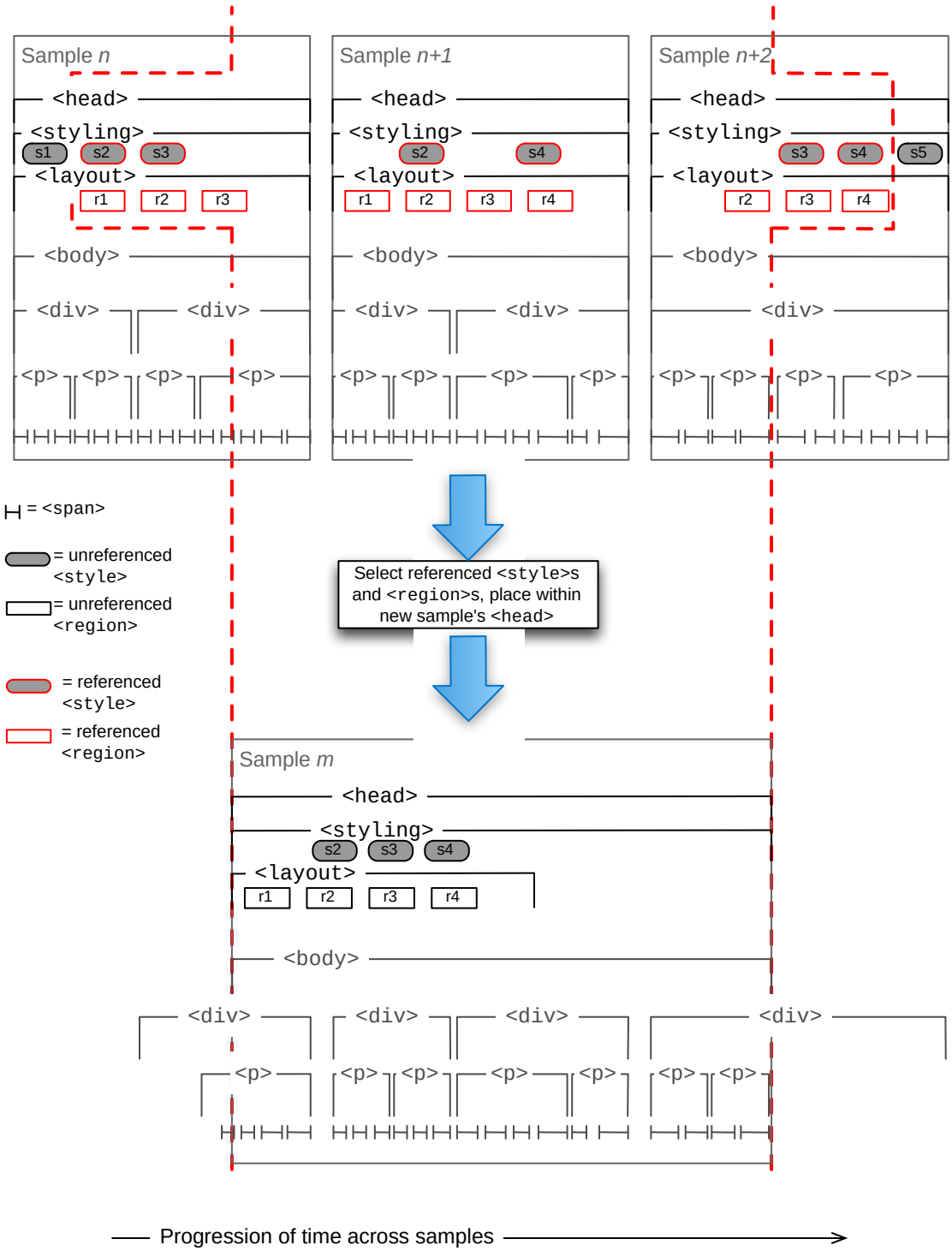
Notes to complete in detail later:

- Outcome at the processor/renderer must be identical when comparing the intermediate synchronic documents
- However it is not necessary actually to create those documents, select the required ones for the new temporal extent and rebuild each sample (though that would work)
- Can use knowledge of the EBU-TT documents to short-cut this process
    - Select s that overlap with the desired output temporal extent, from all the input samples that overlap with the output sample's temporal extent
    - Prune out <p>s and s that do not contain the selected s, or select the ones that do, to achieve the same thing, i.e. the set of s, <p>s and s that need to go into the output sample

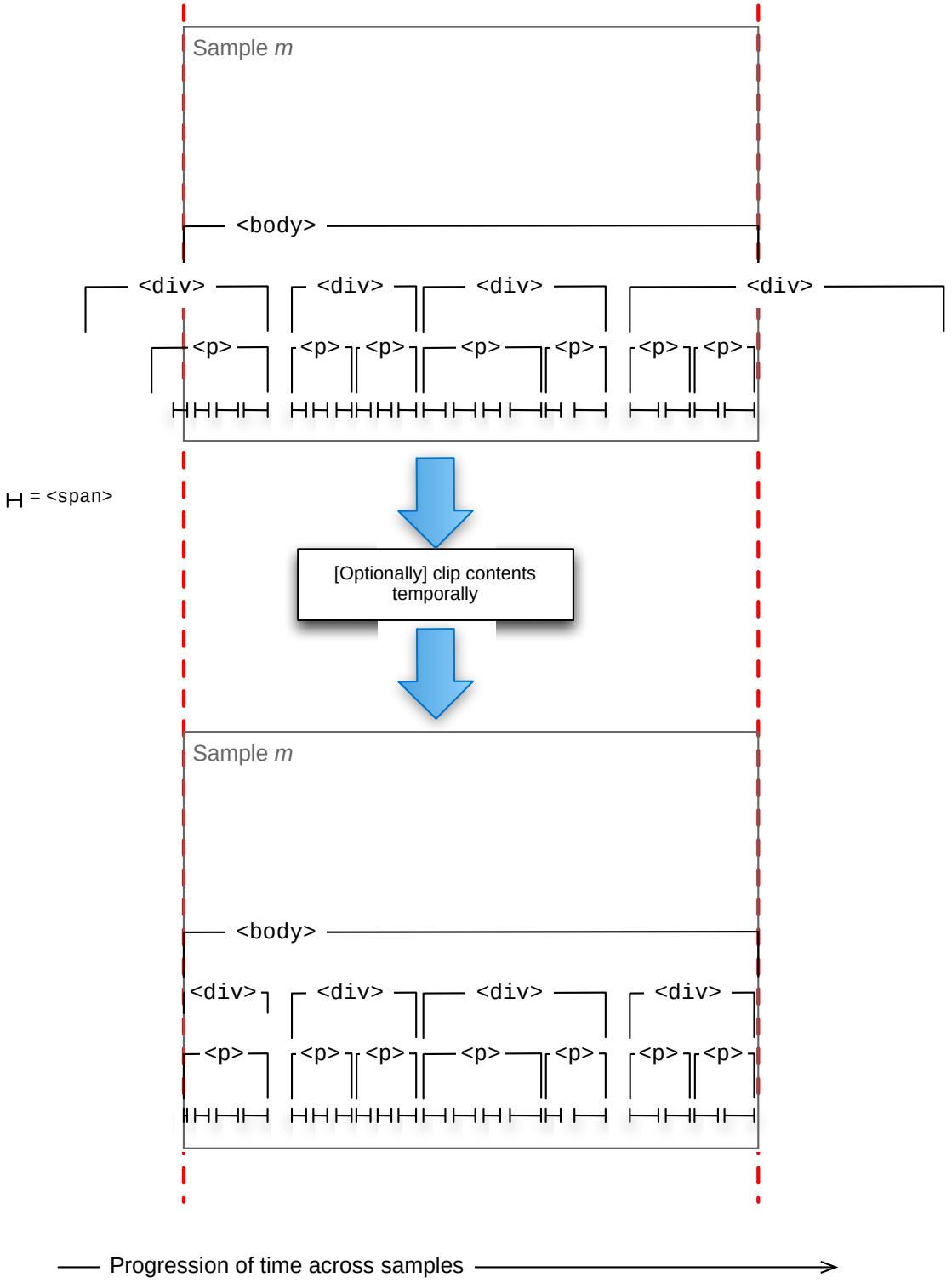
[7] Note that using structure and identifiers as a heuristic to estimate the similarity between subtitles could introduce a fault scenario if the actual subtitle is not also examined. Also it is expected that within rendering implementations the biggest computational cost is incurred by rendering glyphs and backgrounds rather than in deriving styles and positions.



- Prune out layout and styling that is not referenced by the remaining s, <p>s and s, or select the layout and styling that is referenced, to achieve the same thing, i.e. the set of layout <region>s and styling <style>s referenced within the output sample. NB need to ensure no erroneous duplication of xml:id of these so that if the same label was used for different styles or regions in different samples then unique ones are generated in the output



- Optionally clip the begin and end attributes of the <body> and all its descendants to be within the new sample temporal extent



- Add in any sample identification required in the output sample set
- Assemble the new TTML document, and output

]

4.3 Relating time within the sample to external media time

To clarify the impact of the above rules for segmentation on the contents of an EBU-TT sample, this section considers the problem from the perspective of 'what actual values shall go in the EBU-TT `begin` and `end` attributes'.

Note that the assumption that an external context is able to supply the media timeline applies here: this document is generic with respect to specific carriage schemes and architectures.

4.3.1 `begin` time within each sample

The time values within the EBU-TT sample must map to the externally defined media timeline of the sample.

In an extreme case, if and only if the carriage mechanism or other external context has been somehow configured to define that the media time for every sample begins at zero, then the timeline within the EBU-TT shall also begin at zero.

More generally however, and we expect more commonly, the carriage mechanism will specify a series of samples for a given programme, each of which begins at some point during the media timeline of the programme, for example the samples might all have duration 30s and have begin times on the media timeline of 0, 30s, 1m00s, 1m30s etc. In this case the times within each sample shall correspond to the media times, i.e. the first sample has begin and end attributes within the range 0-30s, the second sample 30s-1m00s, the third 1m00s-1m30s etc.

No mechanism is proposed within this document for applying a single offset to each sample, e.g. in the `head` and starting each sample's times within the `body` at zero. Such an approach is not recommended because it adds complexity to the calculations needed when accumulating and splitting documents.

4.3.2 Non-aligned `begin` times

If the time of a `begin` attribute, after mapping into media time, falls between two video frames, the subtitles shall appear on the first video frame immediately after that moment.

4.3.3 `end` time considerations

Note that TTML 1.0 section 10.2.2, and by inheritance EBU-TT, defines the `end` attribute such that it corresponds to a moment *after* the subtitle shall have disappeared. Therefore the value of the last `end` attribute may be exactly equal (after suitable mapping calculations have been applied) to the start of the following sample's media time, and this would not be considered an overlap. This is however not a requirement.

If the time of an `end` attribute, after mapping into media time, falls between two video frames, the relevant subtitle shall be visible on the frame immediately prior to that moment but shall not be visible on the frame immediately after it.

5 Bibliography

[1] **TTML 1.0** Glenn Adams. Timed Text Markup Language (TTML) 1.0. (Second Edition) W3C Proposed Edited Recommendation 09 July 2013 (subject to further edits). http://www.w3.org/TR/ttaf1-dfxp/

[2] **ISO/IEC 14496-12** Coding of audio-visual objects Part 12: ISO base media file format – with:-
Corrigendum 1:2008-12-01
Corrigendum 2:2009-05-01
Amendment 1:2009-11-15
Amendment 3:2011-01-28/DAM (Note 1)

[3] **ISO-IEC 23009-1:2011** Dynamic adaptive streaming over HTTP (DASH).

[4] **SMIL 2.1** http://www.w3.org/TR/2005/REC-SMIL2-20051213/

[5] **EBU Tech 3350 EBU-TT Part 1** Subtitling format definition.
http://tech.ebu.ch/docs/tech/tech3350.pdf

[6] **ETSI TS 102 823 V1.1.1 (2005-11)** DVB Specification for the carriage of synchronized auxiliary data in DVB transport streams:
http://www.etsi.org/deliver/etsi_ts/102800_102899/102823/01.01.01_60/ts_102823v010101p.pdf