ELECTRONIC PUBLISHING, VOL. 6(4), 397–411 (DECEMBER 1993)

Specification of temporal constraints in multimedia documents using HyTime

ROBERT ERFLE

IBM - European Networking Center
Vangerowstr.18
D-69115 Heidelberg, Germany

e-mail: erfle@dhdibm1.bitnet

SUMMARY

Multimedia documents also include time dependent media like audio and video. In contrast to traditional text documents temporal constraints have to be determined that tell a presentation application when and for how long certain parts of the document have to be presented. The paper shows how temporal constraints may be specified with HyTime. An analysis of different approaches covering the specification of temporal constraints resulted in a catalogue of relevant issues. They are explained in the context of an abstract document model. Then it is shown how each issue may be specified with HyTime introducing and explaining all necessary constructs and principles. Several HyTime encoded example scenarios illustrate the actual usage of HyTime building blocks.

KEY WORDS Multimedia Structured documents HyTime SGML MHEG

1 INTRODUCTION

An important aspect of multimedia documents is the specification of temporal constraints with respect to presentation. HyTime offers a rich set of mechanisms to represent such constraints. The goal of this paper is to provide an introduction into these mechanisms, demonstrate their usage within several examples, and show their expressive power. The last point is achieved by an analysis of several approaches which resulted in a catalogue of issues covering specification of temporal constraints. It is shown that HyTime supports all of them.

The term *multimedia document* used in this paper needs some definition. According to the HyTime standard [1] a document is *a collection of information that is identified as a unit and that is intended for human perception.* The attribute *multimedia* is understood as indicating that the document might also contain so called continuous or time dependent media.

The *collection of information* contained in a document is assumed to be logically subdivided into *media items.* A media item constitutes an atomic entity with regard to the specification of temporal constraints. A media item does not necessarily constitute a storage entity. Two voice annotations may be the content of separate media items but be stored in the same file. In such cases there must be means to address that part of the storage entity

CCC 0894–3982/93/040397–15
Received 15 August 1993
Revised 1 December 1993

that makes up the content of the media item. We do not discuss addressing schemes, but assume that they allow to specify media items in such a granularity necessary to determine temporal constraints according to the author's intention.

The temporal constraints of a document are specified by assigning media items temporal properties and by relating media items in time. Additionally, the specification of user interaction plays a role as user interaction may influence the document's presentation with respect to time. Apart from that, a document usually contains the specification of further constraints, like spatial layout. However, in this paper only temporal aspects are considered.

The analysis of temporal constraints includes the following approaches found in literature and in standardization activities.

- *OCPN* described in [2–5].
- *MODE* described in [6,7].
- *Firefly* described in [8,9].
- *Path Operators* described in [10].
- *Presentation Frames* described in [11].
- *MHEG* described in [12,13].

The paper is further structured as follows. First, the different issues of specifying temporal constraints are outlined in the context of the document model used in this paper. At the end of this, a table summarizes the result of the analysis by indicating which issue is supported by which approach. Then, an introduction is given into those parts of HyTime that are required to specify the issues of the catalog. Finally, a conclusion ends the paper.

2 SPECIFICATION OF TEMPORAL CONSTRAINTS

The actual presentation of every document has a duration. Traditional documents usually do not have any notion of time. That means, the author determines neither the overall duration of the presentation nor when and how long single media items are presented. Furthermore, the author does not specify user interaction.

In contrast, multimedia documents have a notion of time because certain media items have a notion of time. In addition to including an audio sequence and a video clip into some document, the author has to specify when these media items have to be presented and for how long. The requirement of specifying when certain media items are to be presented introduces the dimension time into the overall document and thus involves also media items that do not have a notion of time like pure text parts.

According to [9], one can distinguish between three kinds of media items regarding their temporal properties.

1. Media items that do not have a notion of time. The presentation of these media items is not bound to temporal constraints like a certain presentation rate. They will be called *discrete media items* in the following. Examples are text parts or images.
2. Media items that have a notion of time. The presentation of these media items is bound to a certain play back rate. They will be called *continuous media items*. Provided the encoding and the size of the encoded media item is known, the presentation duration can be calculated. Examples are video clips or audio annotations.
3. Media items which have an indeterministic presentation duration. The presentation of the same media item will lead to several different durations when presented several

times. Such media items will be called *indeterministic media items* in the following. Examples are media items that involve a program execution.

Imagine for a moment that the finite presentation duration of a multimedia document is associated with a timeline. Then, each media item can be associated with three figures.

1. The start time of the presentation of the media item.
2. The duration of the presentation of the media item.
3. The end time of the presentation of the media item.

Obviously, these figures contain redundant information. Two of them are sufficient to calculate the third one.

The duration will either be determined by the author (author level) or will be left undetermined until presentation (presentation level).

1. Author level (ISSUE 1). The author assigns a certain duration to a media item either explicitly or implicitly by relating the media item with other ones that already have a fixed duration (relating media items is discussed below). Examples of explicit associations might be: 'show image for 20 seconds' or 'play audio sequence for 30 seconds'.
2. Presentation level. Media items receive their eventual duration during presentation in the following cases.

    - User interaction stops the presentation of a media item (ISSUE 2).
    - Media items that are related to other media items which have a presentation level duration (relating media items is discussed below).
    - Continuous media items which have not been assigned a fixed duration (ISSUE 3).
    - Indeterministic media items which have not been assigned a fixed duration (ISSUE 3).

Having considered duration specification of media items, now the specification of start or end points will be discussed. Start or end points might either be specified absolutely within the duration of the document's presentation or by establishing relationships between media items. Such relationships may involve

- the start points of two media items (ISSUE 4),
- the start position of one media item and the end position of another media item (ISSUE 5),
- the end positions of media items (ISSUE 6),
- the durations of media items (ISSUE 7).

In order to allow delays within relationships like 'start audio 10 seconds after video ended', artificial media items might be introduced. They refer to no content, but represent only the amount of delay which in the example was 10 seconds.

Combinations of the four issues allow the specification of all 13 cases of temporally relating two media items which were introduced by Allen in [14].

Instead of specifying start and end points of media items by explicit relationships, one could also place them absolutely along a common timeline (ISSUE 8). The advantage

would be that extracting or inserting a media item into the document would not interfere with any other media item. This may become annoying when a media is involved in several relationships. Absolute placement of media items also relates them in an implicit way. However, the disadvantage is that one cannot relate objects of unknown duration, as for example, 'play audio when user ends presentation of image'. Additionally, it is not clear which relation, if any at all, the author had in mind. Such knowledge is likely to be important when the document has to be altered later on.

In [10] and [11] relationships may include the order of end times of several media items. An author may specify that the presentation of a media item starts when the first or last of a set of other media items ends its presentation. In the first case, additionally, the rest of the set of media items must also end at that particular point of time (ISSUE 9). An example would be: 'play audio A when first of media items B, C, D ends, and stop the other two media items'. Obviously, such constructs make sense only if not all positions and durations are known *a priori*, as otherwise there would be no indeterminacy regarding which event ends first.

Relationships as well as positions and durations may either be specified precisely or imprecisely (ISSUE 10).

1. Precise Specification. Positions and durations are expressed by exact figures, e.g., 'show image 3 seconds after audio ends'.
2. Imprecise Specification. Positions and durations are expressed in a way that allows interpretation like 'show image shortly after audio ends'. It is then the presentation application that must eventually decide which exact figure the term 'shortly' is mapped to. Owing to explicit relationships one imprecise statement may recursively lead to many other imprecise positions and durations.

There will be cases where the author wants the actual presentation to exactly follow the specified values and others where he or she tolerates some deviations (ISSUE 11). An application might use a given tolerance to compensate for performance deviations. An example may be: 'show image for 30 seconds within a tolerance of 10%'.

Provided the duration specified for a media item does not fit its inherent duration, the author may want to specify how the application should deal with such a situation (ISSUE 12). An application might employ *adjustment* or *replace* policies. There may be two reasons for such cases.

1. The author takes such situations into account during specification of temporal constraints. For example, an audio that accompanies an image where the end of the image is determined by user interaction. Then the author might want the audio to loop as often as necessary or leave the rest of the image presentation unaccompanied after the audio ended.
2. The situation arises during presentation because the application is not able to exactly fulfil the specified temporal constraints. This may, for example, lead to a longer duration for a video which then can't be filled by an accompanying audio.

User interaction may influence when and for how long certain media items are presented. Therefore the author may want to explicitly specify what kind of user interaction will be made available and which media items are concerned.

The following provides a list of different kinds of user interaction.

1. Pause and continue operations during presentation of a continuous media item (ISSUE 13).
2. Fast forward and fast reverse operations during presentation of a continuous media item (ISSUE 14).
3. End presentation of a media item (ISSUE 15).
4. Start presentation of a media item (ISSUE 16).
5. Skip operation to a nondeterministic position within the document (ISSUE 17).
6. Skip operation to a deterministic position within the document (ISSUE 18).

The first two list items describe well known features from analogue devices like a tape recorder. They are obviously meaningless in the case of discrete media items which have not been assigned a certain duration.

The following boolean table tells which issues mentioned above are supported by the approaches taken into account. For lack of space an introduction into the approaches has been omitted. Instead readers are asked to refer to the original papers. The fact of not supporting certain issues does not necessarily constitute a deficiency as it always depends on the set of applications which issues are actually needed.

Some entries in the table receive a reference to an additional remark following the table.

Issue	**Approach**					
	OCPN	**MODE**	**Firefly**	**[Höpner92]**	**[Höpner91]**	**MHEG**
ISSUE 1	YES	YES	YES	YES	YES	YES
ISSUE 2	NO	YES	YES	NO[4]	NO[5]	YES
ISSUE 3	NO	YES	YES	YES	YES	YES
ISSUE 4	YES	YES	YES	YES	YES	YES
ISSUE 5	YES	YES	YES	YES	YES	YES
ISSUE 6	NO[1]	YES	YES	YES	YES	YES
ISSUE 7	NO	NO	NO	NO	NO	NO
ISSUE 8	NO	YES[2]	YES[2]	NO	NO	YES[2]
ISSUE 9	NO	NO	NO	YES	YES	YES[3]
ISSUE 10	NO	NO	YES	NO	NO	NO
ISSUE 11	NO	NO	YES	NO	NO	NO
ISSUE 12	NO	NO[6]	NO[7]	NO	YES	NO
ISSUE 13	NO	YES	YES	NO	NO	YES
ISSUE 14	NO	YES	NO	NO	NO	YES
ISSUE 15	NO	YES	YES	NO	NO	YES
ISSUE 16	NO	YES	YES	NO	NO	YES
ISSUE 17	NO	YES	NO	NO	NO	YES
ISSUE 18	NO	YES	NO	NO	NO	YES

1. May be achieved by using ISSUE 4 and ISSUE 5 taking into account the duration of media items.
2. May be achieved by definition of an artificial ‘timer’ object which is related to other objects.
3. No single specification construct available, but kind of relationship may be expressed by multiple constructs of less complexity.
4. Taken into account within the applications in mind, but not explicitly specifiable.
5. See 4.
6. Some kind of replace policy (e.g., speech synthesis of text part) specifiable, but no adjustment policies.
7. Some adjustment policies specifiable, but no replace policy.

3 SPECIFICATION OF TEMPORAL CONSTRAINTS USING HYTIME

This chapter is intended to give an introduction into those features of HyTime that allow specification of temporal issues within multimedia documents. The catalog from the previous chapter will be used as a guideline along which the HyTime constructs and principles necessary for the specification will be introduced.

There are two articles which give an overall introduction into the basics of HyTime [15,16]. This paper covers only a certain area of HyTime, but presents that in more detail including excerpts from HyTime encoded specifications.

HyTime is an SGML [17] application. Thus, a HyTime document is syntactically an SGML document. This paper does not give an introduction into SGML. However, the description of the HyTime principles and constructs should be understandable without knowledge of SGML. Only the digestion of the encoded examples requires basic SGML skills. For those who are interested an introduction into SGML may be found in ANNEX A, B, C, D of the standard and [18].

HyTime is about *addressing*, *linking*, and *alignment*. Addressing deals with identifying a certain amount of information, which, for example, is necessary by specifying anchors in hypermedia systems. The target of a Hytime address might be a chapter within a text document, the sequence between frame #210 and #510 of some video clip, or a certain rectangle within a still picture. Thus, HyTime addressing schemes could be used to support the logical subdivision of the document content into media items which was assumed in the introduction of the paper. The linking features allow to create links between parts of information which are, for example, fundamental in hypermedia applications. Finally, alignment supports placement of pieces of information within finite coordinate systems. In the following, the focus is on those features that cover alignment in time.

Concerning alignment, HyTime only knows about bounding boxes, which are called *events*. An event usually contains a reference to actual content. This might be a still picture stored in a separate file or some subsequence of a video clip. Addressing schemes are applied to identify the content if necessary. Events are grouped in *event schedules* which provide further structuring of the document.

HyTime does not restrict the content of events to certain media types or coding schemes. A HyTime document contains information about the media types and coding schemes used. For this purpose, SGML provides the NOTATION feature. It is within the scope of applications to use this information in order to properly present the content of events.

Events are placed in *finite coordinate spaces*. A finite coordinate space may have one or more *axes*. Each axis is associated with a *measurement domain* and an *addressable range*. HyTime provides a set of predefined measurement domains like seconds, minutes, or days and allows application-defined domains like frames. The addressable range defines the portion of an axis within which events may be placed.

Each event has an *extent* which defines its *dimension* on each axis. Each dimension consists of two *markers*, the first giving the position of the *first quantum* of the event on a certain axis and the second giving the size of the event. In HyTime the size is called *quantum count*. Figure 1 shows an event placed within a finite coordinate space having only one axis which is measured in seconds and has an addressable range of 600 seconds. The event represents an audio sequence that takes 300 seconds and should start 30 seconds after the document's presentation started, which is assumed to coincide with the beginning of the addressable range.

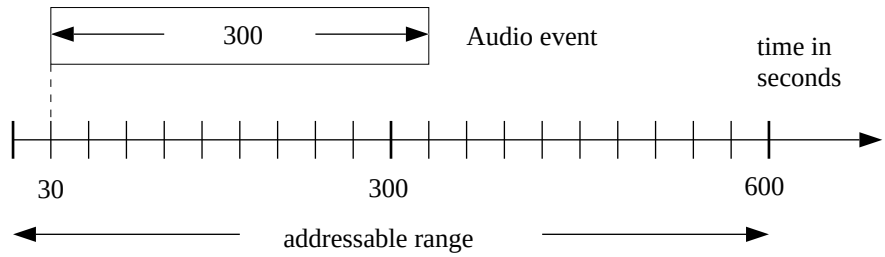

Figure 1. Absolute event placement

The following excerpts from a customized HyTime DTD and a conforming document instance show how the scenario might be encoded.

```
<!-- excerpt from a customized HyTime DTD -->
<!element (timefcs) -- Finite coordinate space --
                    - O      (evsched) >
<!attlist timefcs   HyTime   NAME     fcs
                    -- measurement domain and scaling unit --
                    fcsmdu   CDATA    #FIXED "SIsecond 1 1"
                    -- reference to axis --
                    axisdefs NAMES    #FIXED "timeline" >
<!element (timeline) -- Axis declaration --
                    - O      EMPTY >
<!attlist timeline HyTime   NAME     axis
                    -- measurement domain of axis --
                    axismeas CDATA    #FIXED "SIsecond"
                    -- addressable range --
                    axisdim  CDATA    #FIXED "600" >
<!-- excerpt from a document instance -->
<timefcs>                   <!-- finite coordinate space -->
 <evsched>                  <!-- event schedule -->
  <event exspec="ext-ev-1"><!-- event with ref. to its extent -->
    <containr content="audio-1" type="audio">
<extlist id="ext-ev-1">     <!-- extent -->
 <dimspec>                  <!-- dimension specification -->
    30 300                  <!-- first quantum, quantum count -->
```

Assigning an absolute value to the quantum count allows an author to explicitly determine the duration of a media item (ISSUE 1). Of course the content of the event could have also been a discrete media item.

The extent of an event may also be specified through reference to another event's extent. A *dimension reference* refers to either the *first quantum (selcomp=first)*, *quantum count (selcomp=qcnt)* or *last quantum (selcomp=last)* of another extent. Figure 2 shows two events A, B where event B is positioned directly after event A and has the same quantum count.

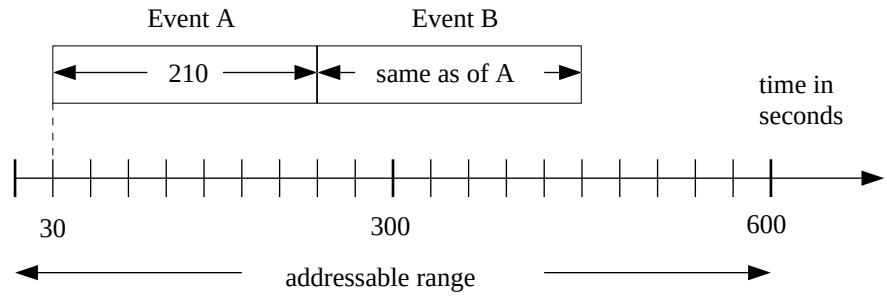

Figure 2. Absolute and relative event placement

A HyTime encoded version of the scenario may look like the following.

```
<!-- excerpt from a document instance -->
<timefcs>
  <evsched>
    <event exspec="ext-ev-1">
    <event exspec="ext-ev-2">
<extlist id="ext-ev-1"> <!-- Extent of event A: absolute -->
  <dimspec id="dim-ev-1">
    30 210
<extlist id="ext-ev-2"> <!-- Extent of event B: relative to A -->
  <dimspec id="dim-ev-2">
    <dimref elemref="dim-ev-1" selcomp="last" flip="flip" >
    <dimref elemref="dim-ev-1" selcomp="qcnt" >
```

Dimension references allow the specification of all the relationships mentioned in the previous chapter as follows:

1. The first quantum of an event refers to the first quantum of some other event (ISSUE 4).
2. The first quantum of an event refers to the last quantum of another event (ISSUE 5).
3. The last quantum of an event, say B, refers to the last quantum of another event, say A (ISSUE 6). As extents are specified by the first quantum and the quantum count, the quantum count of event B must be determined using the dimension reference and the first quantum of event B. This is achieved by applying *marker functions* which are introduced below.
4. The quantum count of an event refers to the quantum count of another event (ISSUE 7). For example, in a slide show the first event (= slide) will be assigned a fixed quantum and all the others will refer to this duration. Subsequent changes to the duration then only require the change of one figure.

The feature of dimension referencing allows the specification of all the 13 temporal relationships between two media items that Allen introduced in his often cited paper [14]. Examples of all cases may be found in ANNEX C of the Draft version of the HyTime standard [19]. As can be seen there, there are even several different possibilities to specify a single case. This holds for most scenarios that are to be specified using HyTime constructs. It is up to the author to choose the one that appears to be most appropriate.

Dimension references explicitly relate events. Apart from the fact that each event might also be placed absolutely along a timeline without using dimension references (ISSUE 8). Thus, HyTime leaves it to the author to decide which method is appropriate in a certain context. Both methods may be employed in the same document.

Explicit relationships may include delays as indicated in the previous chapter. HyTime marker functions in combination with dimension references allow to represent them.

HyTime provides two types of generic marker functions, *HyOp* and *HyFunk*. HyOp supports a fixed set of simple operations like sum, max, or mod. HyFunk is used to define more complex functions by combining simple operations, constants, and dimension references. Thus, highly customized functions may be specified. Function declaration must be done according to the *HyFunk notation* defined in Annex A of the HyTime standard.

Figure 3 shows two events, A and B. Event A is placed absolutely at a timeline. Event B should start some time (150 sec.) after A has started and should end when event A ends.

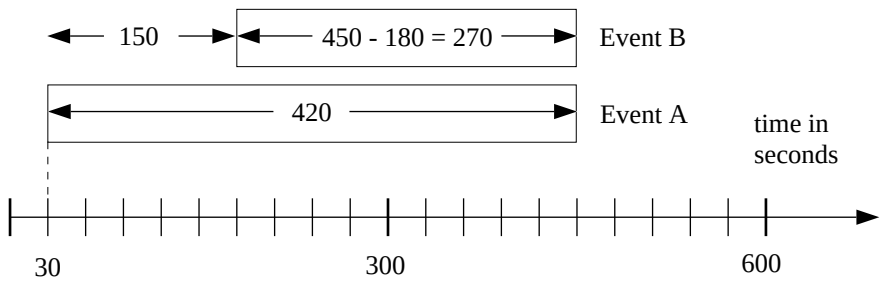

Figure 3. Related events with delay

The element HyFunk may be used either to declare a function or to actually apply a function. Both cases appear in the following HyTime encoded excerpt. The first HyFunk statement declares a function called cal-qcnt which performs a subtraction between the last quantum of some dimension and the first quantum of some dimension. The function expects two parameters which are indicated by the applies the function cal-qcnt. The function performs the calculation ‘last(A) - first(B)’.

```
<!-- excerpt from a document instance                    -->
<!-- Function declaration -->
<HyFunk fn="cal-qcnt">@sub(@fllast(%1) @first(%2))</HyFunk>
<timefcs>
  <evsched>
    <event exspec="ext-ev-1"><!-- Event A -->
    <event exspec="ext-ev-2"><!-- Event B -->
<extlist id="ext-ev-1">
  <dimspec id="dim-ev-1">
    30 420                  <!-- Absolute extent of event A -->
<extlist id="ext-ev-2">
  <dimspec id="dim-ev-2">
    <HyOp opname="sum">     <!-- First(b) = sum(first(A),150) -->
      <dimref elemref="dim-ev-1" selcomp="first" >  150
    <HyFunk usefn="cal-qcnt" args="dim-ev-1 dim-ev-2">
```

An important feature of HyTime is *event projection*. Projection maps the extent of an event from one finite coordinate space to another finite coordinate space. Projection may range from simple one to one mapping to complex transformation functions which, for example, serve to perform 3D rendition of an image. Each *projector* is applied within its *projector scope* that determines a certain range within the source coordinate space. Projector scopes are defined like event extents stating their first quantum and their quantum count.

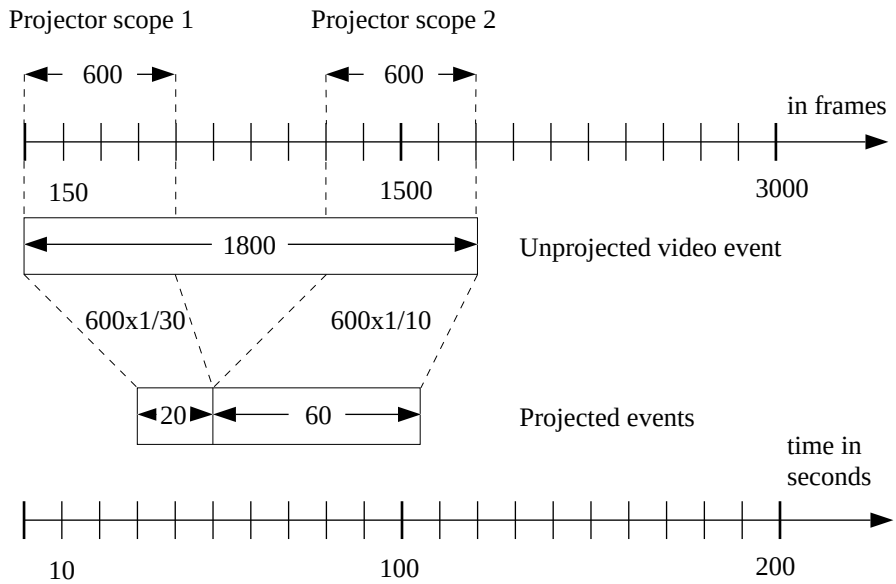

Figure 4. Event projection

Figure 4 shows a video clip event placed at an axis which measures in frames. The video consists of 1800 frames. The projectors map the unprojected video event to two projected video events that are placed along an axis measuring in seconds. The projectors define the projected extents. The extents are specified with absolute values, dimension references, and marker functions as in the previous examples. The functions in the example calculate the projected quantum counts by using different mapping factors (= play back rate). The first part (frame #1 to #600) is to be played back with normal speed (30 frames per second) whereas the second portion (frame #1200 to #1800) is to be played back in slow-motion mode (10 frames per second). The projector functions make use of the current projector scope indicated by #SCOPE as a factor.

The actual presentation includes the projected events only. As can be seen, a portion of the source video clip has been left out (frame #600 to frame #1200). So projection may also be used as an addressing facility defining media items. The projected events do not explicitly refer to any content. They contain content of the unprojected events according to the projector scopes.

The projection example in HyTime encoded form:

```
<!-- excerpt from a customized HyTime DTD -->
<!-- timefcs and timeline as in first example -->
<!element (framfcs) -- Finite coordinate space --
                    - O      (evsched) >
<!attlist framfcs  HyTime   NAME     fcs
                    fcsmdu   CDATA    #FIXED "frame 1 1"
                    axisdefs NAMES    #FIXED "framline" >
<!element (framline) -- Axis declaration --
                    - O      EMPTY >
<!attlist framline HyTime   NAME     axis
                    axismeas CDATA    #FIXED "frame"
                    axisdim  CDATA    #FIXED "3000" >
<!-- excerpt from a document instance -->
<!-- Function declaration: projectors -->
<HyFunk fn="fqc2rqcn">@mult(@div(1 30) @qcnt(#SCOPE))</HyFunk>
<HyFunk fn="fqc2rqcs">@mult(@div(1 10) @qcnt(#SCOPE))</HyFunk>
<extlist id="expsco-1">   <!-- Projector scope for 1st portion -->
 <dimspec id="prosco-1">
   1 600
<extlist id="expsco-2">   <!-- Projector scope for 2nd portion -->
 <dimspec id="prosco-2">
   1200 600
<extlist id="ext-unpr">   <!-- Extent of unprojected event -->
 <dimspec id="dim-unpr">
   1 1800
<framfcs>                 <!-- Unprojected coordinate space -->
 <evsched id="upevsch">
  <event exspec="ext-unpr">    <!-- Unprojected event -->
   <containr content="video-1" type="video">
 <baton id="frambat">
  <proscope exspec="expsco-1">
   <projectr id="pjfr2rt">
    <extlist id="pex-1">  <!-- Extent of projected event -->
     <dimspec id="dim-pr-1">
       30
      <HyFunk usefn="fqc2rqcn">
  <proscope exspec="expsco-2">
   <projectr id="pjfr2rt">
    <extlist id="pex-2">  <!-- Extent of projected event -->
     <dimspec id="dim-pr-2">
      <dimref elemref="dim-pr-1" selcomp="last" flip="flip" >
      <HyFunk usefn="fqc2rqcs">
<timefcs>                 <!-- Projected coordinate space -->
 <evsched id="pevsch">
  <event exspec="pex-1">  <!-- Projected event -->
  <event exspec="pex-2">  <!-- Projected event -->
<batrule evscheds="upevsch" baton="frambat" pevsched="pevsch">
```

Projection keeps the intention of the author in the document. Preserving the unprojected events and the projectors instead of just the projected extents tells why certain events lasted as long as they did. This information might be useful in subsequent editing of the document.

The kind of relationship mentioned within ISSUE 9 can be specified in HyTime using the functions min and max together with dimension references. The example given may be represented by defining the first quantum of event A to be a reference to the minimum of the last quanta of event B,C and D. This makes event A start presentation when the first of events B, C, and D ends which is obviously the minimum of last quanta. Additionally the last quantum of all the events B, C, and D must be the minimum of these last quanta. This can be achieved by specifying the quantum counts of events B,C, and D using a function that takes the minimum of the last quanta as a parameter.

So far, all figures were meant to be precise. Instead, the author might want to specify certain quanta or quantum counts imprecisely (ISSUE 10). This can be done in two ways which both imply projection.

1. Each projector carries an attribute *strict* which determines the strictness of the projection. By default, projection is meant to be strict. The author may instead provide application-defined values for this attribute like 'rubato' or 'within a certain percentage'.
2. The projector itself does not contain a precise formula, but again application-defined descriptive text like 'rubato' or 'fast'. It is then up to the application which exact projection should be applied to some descriptive text like 'rubato'.

Music notation also uses descriptive text for certain aspects like, for example, tempo. During rendition, which means during play time, the musician must decide a certain tempo for instructions like 'allegro'.

Similarly to the previous point the author may provide precise figures but tolerate a certain deviation during presentation. The allowed degree of deviation can be specified using the strict attribute of each projector (see above) (ISSUE 11).

Each event refers to an extent specification which determines the size of the event. There may be cases in which the content of the event does not fit the specified size (ISSUE 12). Therefore, the author can determine what should be done in such cases by using an *extent reconciliation* strategy. A reconciliation strategy constitutes a separate HyTime element which event elements refer to. It encompasses, among others, the following attributes:

- The optional attribute *replace* refers to another event. The content of this event is to be used instead of the not fitting one.
- The attribute *align* determines at which point the content media item and the extent are to be aligned before applying the strategy. This might either be the first, last or center quantum or some arbitrary point within each of them.
- The attribute *vamp* mainly specifies whether cropping or filling should be applied in the case the media item is too big or too small.

There may be a chain of such strategies to be tried serially. Some application-defined criteria in the attribute *altcrit* provide information about the circumstances under which the strategy should be applied.

HyTime does not provide explicit means to represent user interaction as MHEG does. Instead, certain HyTime constructs are to be used to specify a kind of interface which will become active during presentation.

Such mechanisms include:

- Declaration of application-defined elements which may have several attributes. HyTime explicitly defines which of its element types may contain application-defined ones. For example the content of an event will often be of this kind (see *containr* element in the first example). Thus, an event may also contain or represent means of user interaction like a push button or a mouse click.
- HyTime provides a generic *marker query* element type which may contain any application-defined function. It resolves to marker values. In the domain time such functions may deliver certain points of time or intervals depending on user interaction which may then be used to process extents of events. Such functions may also be applied to deliver the endpoints of the presentation of continuous media items and indeterministic media items (ISSUE 3).

The availability of pause and continue operations (ISSUE 13) may be expressed by an application-defined attribute declared for the content of an audio or video event. Alternatively, a new event may be introduced which signifies the availability. By relating this event to others it would be possible to dedicate the operations to particular time intervals.

Fast forward and fast reverse operations (ISSUE 14) may be specified as described in ISSUE 13. Apart from that, a projector function, e.g., one that calculates real-time extents from frame extents, may make use of an application-defined marker query function which requests the desired play back rate during presentation. So, a user may choose from a predefined set of play back rates which determines the rate parameter. The default value which is active unless the user performs the operation is the normal play back rate.

End operations (ISSUE 15) influence the quantum count of media items. The possibility of ending a media item may be expressed by defining the quantum count with the help of a marker query function that returns the point of time when the user ends the presentation of the media item. This feature becomes important when a scenario is to be specified that includes static media items which should be presented to a user until explicitly ended (ISSUE 2).

Start operations (ISSUE 16) can be handled similarly to end operations. Here, the marker query function would be used to define the first quantum of the media item which may be started by the user.

Skip operations to nondeterministic positions (ISSUE 17) within a multimedia document are a difficult matter. The new position might be in the middle of a video or audio event. Presentation applications must perform a context switch to the context of an arbitrary position which might be very difficult to obtain especially with regard to the domain time. The operation may be specified by an additional event that contains a *contextual link* element from the HyTime linking features. The link target is a marker query element which represents a kind of slider and resolves to the new position once the slider is used. Thus, a skip operation to a nondeterministic position is modelled as activating a hyperlink whose target is determined at presentation time.

Skip operations to deterministic positions (ISSUE 18) may also be modeled using an event that contains a contextual link. The target of the link is a set of predefined positions a user can skip to by activating the link and selecting the desired target position. Thus the link event might be represented by some menu offering the possible target positions.

4 CONCLUSION

As was shown in the previous chapter, all issues of specifying temporal constraints within multimedia documents can be represented using HyTime constructs. Some of them require application-defined element types or marker functions. This is true particularly for those issues that deal with user interaction. HyTime elements provide dedicated places for the integration of application-defined elements thus offering a kind of interface which eases processing.

HyTime was not designed to standardize the specification of every feature of every application. This would be impossible in any case as one cannot foresee future developments. Instead, HyTime standardizes a framework of means that allow the specification of basic properties that are supposed to be common to most hypermedia documents. It is the extensibility and flexibility of the framework that will allow usage of HyTime in many different applications still handling many issues in the same manner and thus reducing investment.

The exchangeability of multimedia documents suffers from the fact that many different models are used to specify the same set of constraints. The set of different approaches considered in the analysis of this paper proves this fact. Usage of HyTime would increase the exchangeability of multimedia documents or parts of them between applications.

In today's information bases nonlinear navigation supported by hyperlinks plays an important role. Nearly all on-line help facilities provide hyperlink functionality. Thus multimedia document presentation and hyperlink navigation will come together. The usage of a common model that covers specification of constraints in multimedia documents and specification of anchors and links will probably promote the integration of both applications. HyTime would be a candidate for such a model. In ISSUE 17 and 18 it was shown how linking and temporal alignment may be combined in HyTime.

ACKNOWLEDGEMENTS

Many thanks to Charles Goldfarb. The discussions with him helped a lot to understand the principles of HyTime and to develop the HyTime examples in this paper.

REFERENCES

1. *International Standard ISO/IEC 10744: Information Technology — Hypermedia/Time-based Structuring Language (HyTime)*, ed., International Organization for Standardization, Geneva/New York, first edition, 1992.
2. T.D.C. Little and A. Ghafoor, 'Synchronization and Storage Models for Multimedia Objects', *IEEE Journal on Selected Areas in Communications*, **8**(3), (April 1990).
3. T.D.C. Little and A. Ghafoor, 'Network Considerations for Distributed Multimedia Object Composition and Communication', *IEEE Network Magazine*, (November 1990).
4. T.D.C. Little and A. Ghafoor, 'Scheduling of Bandwidth-Constrained Multimedia Traffic', in *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg 1991*, ed., R.G. Herrtwich, Springer, Berlin, (1992). Lecture Notes in Computer Science, No. 614.
5. T.D.C. Little, A. Ghafoor, and C.Y.R. Chen, 'Conceptual Data Models for Time-Dependent Multimedia Data', in *Proceedings of the Workshop on Multimedia Information Systems* , (1992).
6. G. Blakowski, J. Hübel, U. Langrehr, and M. Mühlhäuser, 'Tool support for the synchronization and presentation of distributed multimedia', *Computer Communications*, **15**(10), (December 1992).

7. G. Blakowski, *Entwicklungs- und Laufzeitunterstützung für verteilte multimediale Anwendungen*, Verlag Shaker, Aachen, 1993.
8. M.C. Buchanan and P.T. Zellweger, 'Scheduling Multimedia Documents Using Temporal Constraints', in *Proceedings of the Third International Workshop on Network and Operating System Support for Digital Audio and Video, La Jolla, CA, 1992*, ed., P.V. Rangan, Springer, Berlin, (1993). Lecture Notes in Computer Science, No. 712.
9. M.C. Buchanan and P.T. Zellweger, 'Automatic Temporal Layout Mechanisms', in *Proceedings of the First ACM International Conference on Multimedia* , ACM Press, New York, (1993).
10. P. Höpner, 'Synchronizing the presentation of multimedia objects', *Computer Communications*, **15**(9), (November 1992).
11. P. Höpner, 'Presentation Scheduling of Multimedia Objects and Its Impact on Network and Operating System Support', in *Proceedings of the Second International Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg 1991*, ed., R.G. Herrtwich, Springer, Berlin, (1992). Lecture Notes in Computer Science, No. 614.
12. *ISO/IEC CD 13522-1: Information Technology — Coded Representation of Multimedia and Hypermedia Information Objects (MHEG)*, ed., International Organization for Standardization, first edition, 1992.
13. F. Kretz and F. Colaitis, 'Standardizing Hypermedia Information Objects', *IEEE Communications Magazine*, (May 1992).
14. J.F. Allen, 'Maintaining Knowledge about Temporal Intervals', *Communications of the ACM*, **26**(11), (November 1983).
15. C.F. Goldfarb, 'HyTime: A standard for structured hypermedia interchange', *Computer*, (August 1991).
16. S.R. Newcomb, N.A. Kipp, and V.T. Newcomb, 'The HyTime Hypermedia/Time-based Document Structuring Language', *Communications of the ACM*, **34**(11), 67–83, (November 1991).
17. *International Standard ISO/IEC 8879: Information Processing — Text and Office Systems — Standard Generalized Markup Language (SGML)*, ed., International Organization for Standardization, Geneva/New York, first edition, 1986.
18. C.F. Goldfarb, *The SGML Handbook*, Oxford University Press, Oxford, 1990.
19. *Draft International Standard ISO/IEC DIS 10744: Information Technology — Hypermedia/Time-based Structuring Language (HyTime)*, ed., International Organization for Standardization, Geneva/New York, 1991.