Do You Have the Time?
Composition and Linking in Time-based Hypermedia

Lynda Hardman, Jacco van Ossenbruggen, K. Sjoerd Mullender, Lloyd Rutledge, and Dick C.A. Bulterman

CWI
P.O. Box 94079
1090 GB Amsterdam, The Netherlands
Email: {Lynda.Hardman, Jacco.van.Ossenbruggen}@cwi.nl

ABSTRACT

Most hypermedia models and systems do not incorporate time explicitly. This prevents authors from having direct control over the temporal aspects of a presentation. In this paper we discuss the concept of presentation time — the timing of the individual parts of a presentation and the temporal relations among them. We argue why time is necessary from a presentation perspective, and discuss its relationship with other temporal views of a presentation. We discuss the requirements and present our solution for incorporating temporal and linking information in a model of time-based hypermedia.

KEYWORDS

Time-based hypermedia, composition, links, Amsterdam Hypermedia Model, SMIL.

INTRODUCTION

While time is a dominant characteristic of multimedia presentations, temporal aspects of hypermedia presentations have, until recently, been neglected. We claim that both temporal and linking aspects need to be supported by authoring and playback systems and the underlying hypermedia document models. We argue the need for this integration from three perspectives—those of an author, a hypertext system designer and a multimedia system designer.

From an author's perspective, time factors are important elements in the way hypermedia is conceived and received [18]. Fiction, for example, is typically based on chronological development of the narrative. When the narrative is expressed as multiple media elements then the temporal arrangement of these elements should be able to reflect its chronological development. An author thus requires some means of manipulating the timing within a presentation.

From a hypertext system designer's perspective, temporal aspects of a presentation are a consequence of extending the hypertext model with concurrent multiple media streams. The traditional node/link model underlying most hypertext systems is unable to express temporal relations explicitly. We draw a parallel with hierarchical and temporal structures—just as composite structure can be expressed using links, temporal relationships could also be encapsulated in a special user-defined link type. Halasz [7] points out that although composition is possible using links, users expect the system to understand the semantics of composition. Composition is thus included explicitly in the Dexter hypertext reference model [5]. Likewise, to support fundamental temporal aspects of a document, we advocate that these should not be modelled by links but be encoded explicitly within a hypermedia model [9].

From a multimedia system designer's perspective, temporal aspects are already supported but need to be integrated with hypertext linking. The links supported by current multimedia systems do not alter the inherent linearity of the multimedia presentation. We distinguish three cases of linking. The first is essentially traversing the timeline of a single linear multimedia presentation. The second allows “hypertext” navigation among a web of several linear multimedia presentations. The third allows navigation within and among non-linear multimedia presentations. This third type of hypermedia linking requires the full integration of both multimedia synchronization and hypertext linking.

The research reported here is a continuation of work reported in previous articles: the concept of link context is discussed from a system implementation point of view in [12]; an overview of the Amsterdam hypermedia model is given in [11]; and complete details of the model are presented in [9]. This paper reports new insights in the temporal and link activation aspects of a hypermedia presentation. In particular, we present four distinct temporal axes, three types of linking and introduce explicit activation information necessary for multiple ended links within time-based hypermedia.

In the following sections we discuss the temporal aspects of a hypermedia presentation and how they influence the underlying document model. We introduce the concept of presentation time—the timing of the individual parts of a presentation and the temporal relations among them. Presentation time can be further divided into different stages of authoring and playing back a presentation. We describe each of these and discuss their relationship with other temporal views of a presentation. We derive the requirements for including presentation time along with the consequences for linking in a model of hypermedia. We illustrate solutions for incorporating temporal and linking aspects using the Amsterdam hypermedia model [12] and SMIL [14].

PRESENTATION TIME

To clarify the discussion, we use the word *presentation* to refer to the artifact the reader experiences during the course of playing a document. The *document* is the underlying storage representation which is perceived by an author as a single unit, independent of whether it is a single file, multiple files, or stored in a database.

A document contains information needed by a presentation system to direct the playback of the elements making up the presentation. Part of this information relates to the temporal aspects of the presentation, that is, when each element is displayed on the screen, for how long, and when it is removed from the screen. A hypermedia presentation consists of multiple elements of various media, of which, at any time during the presentation, some number are playing on the screen.

While aspects of time can be stored in the document, they manifest themselves in different ways in different places, and at different stages of the authoring and presentation processes. We introduce the term *presentation time* to denote the timing from the perspective of the presentation.

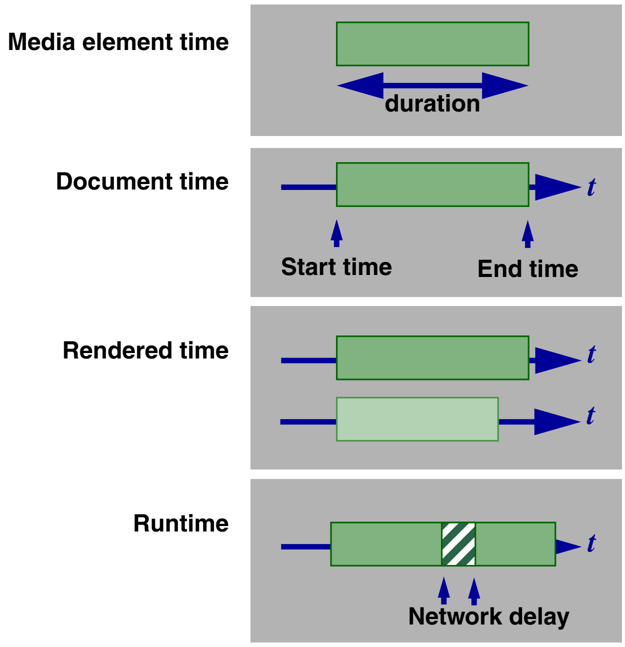

Figure 1. Presentation time.

Presentation time can be split into four sorts of time: media element time, document time, rendered time and runtime, shown in Fig. 1. Each of these corresponds to a stage in the authoring or presentation process.

Media element time is the duration of (part of) the media element included in a presentation. The duration of a video or sound clip can be calculated from the data item format using an interpreter for that format and is intrinsic to the media item. The duration of text, images or live feeds is indefinite (unless otherwise specifed by an author). If no further temporal transformations are carried out on the media element, then the media element time is the length of time it would take to play the media item on an ideal system.

Media element time is a property of the media element and cannot be manipulated by an author, other than by editing the media element itself. *Document time* is the time the author can assign and manipulate within the presentation and can be stored in the document. Since it is confined to the authoring stage of a presentation, it is independent of processing in the subsequent stages—rendering and playback—and can thus be calculated statically on the basis of the durations of the media items and the specified temporal relationships among them.

If alternate media elements are supplied in the document for different reader or system requirements, then choices have to be made after the authoring stage and before the presentation is displayed to the reader. Since alternatives may specify different durations, these choices will affect the overall timing of the document. The decisions are made at rendering time when a reader's preferences and system capabilites are known. This results in the *rendered time* of the presentation. If no choices were available then rendered time is the same as document time. The rendered time is the time the presentation would take to play on an ideal system.

The final stage of the process is displaying the presentation to the reader. This takes place at *runtime*, which is the time it takes, in real time, to play the presentation. This is when a reader can interact with the presentation and can manipulate the presentation's timeline by, for example, link traversal.

Each time type has its own time axis which can be used for synchronizing the control (e.g. start, stop) of other events. For example, a media element can be synchronized to start 5 seconds after the beginning of a video element. If the video is delayed at runtime then so will the start of the media element. Alternatively, the media element could be scheduled to start 5 seconds after the beginning of document time. In which case, if the video is delayed it has no influence on the start time of the media element. Synchronization within media element time can be compared with intramedia synchronization; synchronization within document and rendered time can be compared with intermedia synchronization [4].

Fig. 1 illustrates media element time, document time, rendered time and runtime for a single media element in a document. The only time axis a media element has is that based on its data format. The document time axis is calculated on the basis of all the media items in the presentation and their temporal relationships. In other words, the multiple time axes of the media items are united to form a single time axis. If there are choices as to which media element should be displayed in the final presentation, e.g. whether to display text or play audio, then there is insufficient information for calculating the document time and this has to be postponed until rendering time. The rendered time axis is calculated in the same way as the document time axis, where there are no remaining choices for alternate media items. The runtime time axis is the same as the rendered time axis for an ideal playback system and otherwise the runtime time axis (realtime in fact) will continue while the rendered time axis is paused, due, for example, to download delays or user interaction.

WHO ELSE HAS THE TIME?

Luesebrink [18] makes the distinctions of interface time and cognitive time. *Interface time* is the physical span of time that the reader interacts with the text and includes mechanical time, reading time and interactive time. *Cognitive time* is the span of chronological time that the reader constructs or reconstructs to encompass the content of the narrative. It includes real, narrative and mythic time.

In our terms, cognitive time is the time expressed and developed within the content of the presentation and cannot be directly found within presentation time, although can be expressed through it. For example, an event in the narrative occurs in the morning and a second event occurs that afternoon. An author may choose to place the morning event before the afternoon event in the presentation, but may also choose, on artistic grounds, to reverse the order. So while presentation time is necessary for conveying the temporal information, there is not necessarily a direct correspondance. Rutledge *et al.* [20] explore this topic further, in particular how temporal relations, such as those relevant to cognitive time, can be expressed in temporal, spatial or even navigational ways within a document.

Interface time corresponds closely to our definition of runtime. *Mechanical time* is the time occupied by the computer on non-content processing activities, such as downloading data and registering mouse movement. *Reading time* is the time the reader is engaged with the presentation. *Interactive time* is the time the reader is engaged with meaningful exchange with the text, for example when deciding which link to follow next. In our terms, mechanical time is part of runtime, that is, the time it takes to play the presentation independent of any reader experience or interaction. Reading time can occur only during runtime, since otherwise there is no presentation to experience. Interactive time can also only occur during runtime.

The precise details of media item time, document time and rendered time are irrelevant to a reader, and as such they are not covered by Luesebrink's classification. The author, however, needs to know the details of how the presentation is built up. Making the document time explicit and editable gives the author the necessary control.

In their Firefly system, Buchanan and Zellweger [3] discriminate compile time and runtime temporal formatting algorithms. Compile time formatting yields a single, contiguous temporal extent which corresponds to our document time. Runtime formatting takes interactive events into account, consistent with our definition of runtime. Although not explicitly discriminated, aspects of document time and runtime temporal constraints are also used in [1], [8], [16], [17]. Temporal constraints in HyTime typically relate to media element time and document time. Issues regarding rendering time and runtime can be addressed using application defined element types or marker functions [5].

WHERE IS THE TIME?

In this section we look at the different types of presentation time—media item, document, rendered and runtime—and describe the types of control an author may wish to have over these. We then describe how the required information can be made explicit and encapsulated in a model of hypermedia.

Media element time

The media element is the basic building block of a presentation. It can have its own intrinsic duration, for example if it is a video clip or an audio fragment. The media element need not refer to a complete video file, but may be a part of it. This is, for example, expressed in SMIL [14] as `clip-begin` and `clip-end` attributes of the media object element, illustrated in Fig. 2.

Document time

Single media element object

Whether or not a media element has an intrinsic duration, an author should be able to assign a duration to the element, where this can be interpreted as either looping/stretching or cutting/shrinking the element by the playback system, Fig. 2. Transformations such as these are termed event projections in HyTime [15]. SMIL has the `repeat` attribute for looping the element, along with `begin`, `end` and `duration` attributes[1]. Note that the duration may be indefinite, either because a text item has been given no other duration, because the

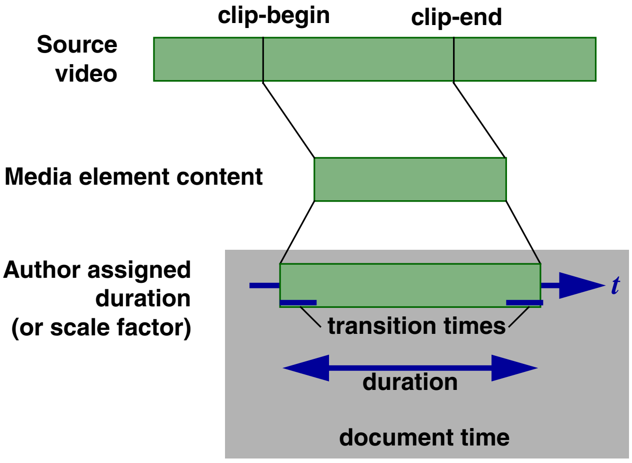

Figure 2. Duration of media element object.

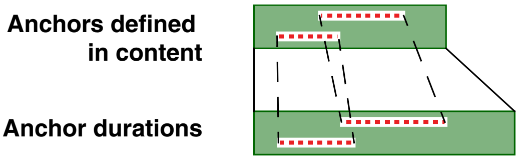

Figure 3. Duration, start and end times of anchors.

repeat factor is set to “indefinite” or because the video/audio is a live feed.

In the Dexter model [5], the media element is encapsulated in an atomic component along with other information, such as a unique identifier. The atomic’s presentation specification allows the addition of other information such as spatial and, in this case, temporal information. In the Amsterdam Hypermedia Model (AHM) [9], [10], based on the Dexter model, the duration is a requirement[2] for an atomic component. We will refer to the duration based on all the information about the media element as the *atomic component duration*.

In addition to the assigned duration, a transition may be associated with the beginning or end of a media element. Transition information includes the duration of the transition and a special effect, such as “fade in” or “fade out”. The transition duration is not part of the content of the media element, so is defined in document time, Fig. 2.

An anchor can also have associated time. For a continuous media type, the duration of the anchor is implicitly given by specifying a part of the content, Fig. 3. For a non-continuous media type, such as text or images, the anchor can be assigned a duration. An anchor is defined with respect to media element time, so that as well as duration it also has a start time and end time with respect to the content. If the duration of the content is scaled in the atomic component then the anchor times are similarly scaled. The anchor duration cannot be stored within a Dexter anchor, but it can be within an AHM anchor.

Multiple media element objects

An atomic component has of itself a duration, but no start time, since there is no external timeline against which to specify a start time. Bringing a number of media elements together in a single presentation requires not only that their durations are known, but also that their respective start times are known. This requires temporal relationships to be defined among the constituent atomic components which, in effect, align the individual time axes with one another. This results in what is commonly termed a multimedia document, where the timing relations among the elements of the scene are fixed. A further distinction can be made in that the synchronization relationship specified can be soft or hard.

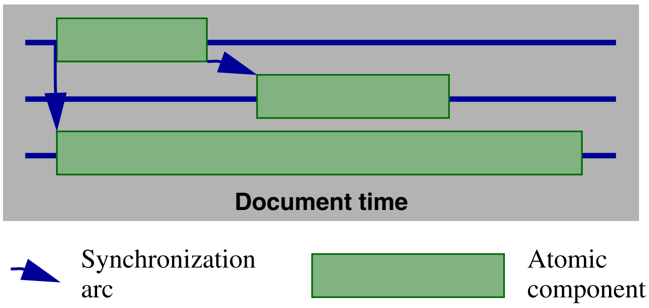

Figure 4. Temporal composition

Soft synchronization specifies that if, at runtime, content is delayed in reaching the reader then the rest of the presentation should continue to play. Hard synchronization specifies that the rest of the presentation should pause and wait for the content that was delayed.

The composition of atomic (and other) components is encapsulated in the composite component in the Dexter model. The corresponding presentation specification allows the addition of temporal information. In the AHM, however, the temporal composition of atomic (and composite components) is modelled as a specialised composite component, termed a *temporal composite*.

Fig. 4 shows a temporal composite, where three atomic components are shown along with their temporal constraints. The duration of each of the elements is the atomic component duration. The relations among components are specified in the AHM using synchronization arcs. These allow the start time of the destination component to be specified relative to the source component. A synchronization arc is directional and does not affect the timing of the source. Each end of the arc is given in terms of an anchor value, so that the synchronization point need not be restricted to being at the beginning or end of a component, but can be related to some content-based event within the object.

A single child of a temporal composite requires no synchronization arc and the duration of the composite is equal to the duration of the child. For each subsequent child a synchronization arc is needed. The arcs are required to form a fully-connected graph and thus define a single contiguous temporal extent for the composite. This is to ensure that, from a temporal perspective, once a temporal composite has been created it can be treated as if it were an atomic component. This allows nesting of temporal composite components in a temporal hierarchy, which is comparable with Allen’s “reference hierarchy” in [2]. The temporal extent of the composite, as calculated from all its children and synchronization arcs, is the equivalent of the duration of an atomic component and so can, for example, be scaled. SMIL provides two temporal composites—`par` and `seq`. Each allows scaling through the assignment of `duration` and `repeat` attributes.

[1] Since these can give conflicting information, when for example the end time minus the begin time does not equal the duration, the SMIL specification includes a number of rules for conflict resolution.
[2] One of the allowed values is “indefinite”.

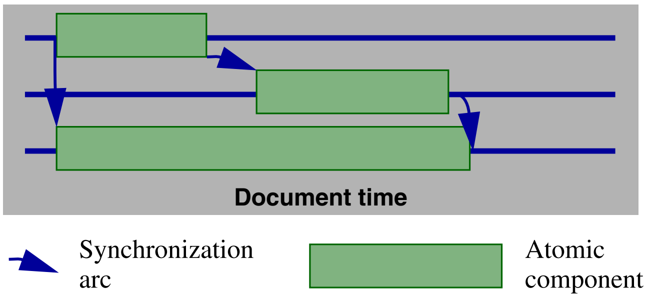

Figure 5. Two synchronization arcs for one component.

A single synchronization arc specifies the relationship of the time axes between two components, thus determining the relative start times. If the durations of both components are definite then the end times are also determined. If one of the durations is indefinite, then a second synchronization arc to that component can be used to instantiate the duration. A second synchronization arc may also be used to scale the duration of the destination component. These cases are illustrated graphically by the lowest component in Fig. 5.

No more than two synchronization arcs are valid, since if a third synchronization arc is specified then it either gives information consistent with that already specified, in which case it is redundant, or it gives conflicting information. A more complex model could be developed which would allow different temporal scaling within parts of a piece of content, so that the third, and subsequent, synchronization arcs could specify that different sections of the component play at different rates.

Note that with constructs such as `par` and `seq` in SMIL the synchronization arcs are defined implicitly, Fig. 6. Explicit synchronization information can be given using attributes such as `begin`.

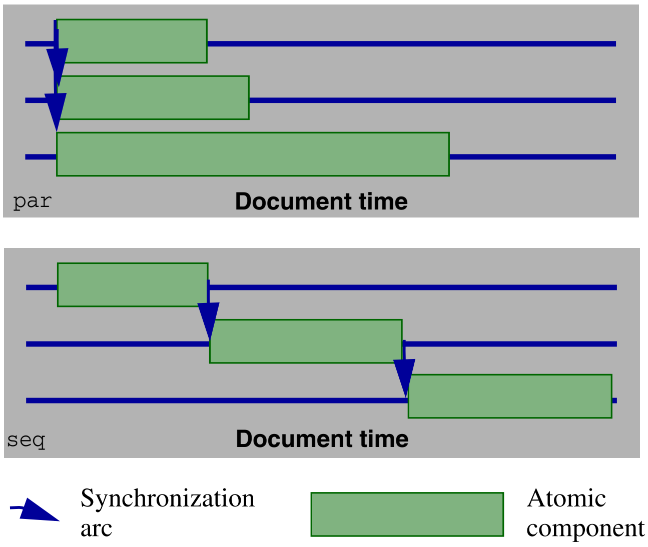

Figure 6. `Par` **and** `seq` **in SMIL.**

Information describing transitions among multiple media element objects, in particular the transition duration, can also be usefully associated with a temporal composite, and should be recordable in a document model.

Rendered time

Presentations can be adapted to reader preferences and system hardware by providing several alternatives for each media element. This allows the same information to be expressed in different user languages or different bandwidth media, such as video or still images. In documents that contain alternative media elements, the temporal information does not specify a single contiguous extent, since the duration of each of the alternates is not guaranteed to be the same. This means that for each occurrence of an alternative there is a break in the document timeline. At rendering time there is sufficient information to make a choice of media element, and the rendering software should be able to generate a consistent schedule for all possible choices. This results in the timing of all the alternatives being known and thus reduces the presentation to a single contiguous extent on the rendered timeline.

In order to accommodate alternatives within a presentation, a grouping is needed of components out of which at most one will be shown to the reader. No synchronization information is needed among the children of this grouping. In the AHM we thus provide, in addition to a temporal composite, an *atemporal composite*. A form of atemporal composition for alternatives is embodied by the SMIL `switch` statement, where the rendering software selects to use the first suitable child, on the basis of lexical ordering. This, however, leads to problems with linking to a destination within a switch. A SMIL link can have only one destination, and when linking to nodes within a switch it is not known until rendering time which node will be played. A solution to this is to specify a link destination in each of the children of the switch, but this would require support for multiple-destination links. These are included within the AHM, although are not currently implemented.

Runtime

There are a number of ways of interacting with time-based hypermedia at runtime. We discuss these in three sections: interacting with a single linear multimedia presentation, linking withing a collection of linear multimedia presentations and linking within non-linear presentations. We first describe the ways within which a reader may wish to interact and then discuss the implications for a model that captures these.

Interaction within a single multimedia presentation

The basic runtime behaviour of a multimedia presentation is that it starts playing and continues to play for its rendered time duration, interrupted only by network delays. A reader can interact with a single, linear multimedia presentation in two ways—by interacting with the rendered timeline or by following predefined links. Player controls, such as pause/play and/or fast forward/backward, allow the reader to traverse the timeline at different speeds in different directions. The difference with controlling a single video is that multiple media elements are bound to the same time axis, and are thus manipulated as a unit.

Links can be used to traverse the rendered timeline, where the destination anchor is in the same multimedia presentation. This is the equivalent to fast-forwarding (reversing or jumping) the presentation to the beginning of the destination anchor. In SMIL, the behaviour of following the link is defined to be that of playing the presentation as if it had been fast-forwarded to the beginning of the specified element, Fig. 7. This is sufficient for linking within a single presentation, since the point on the rendered timeline from where the runtime should play can be calculated.

While both reader interaction and links can be used to traverse the rendered timeline, making use of a link component allows other information, such as transition information, to be specified, [9]. The transition information includes temporal information—the duration of the transition—and style.

Linking among linear multimedia presentations

In addition to playing or linking within a multimedia presentation, a reader can also interact by following links to other presentations. This requires specification of what happens to the playing presentation, which we term the source presentation, and the destination. For links among linear multimedia presentations the activation rules are straightforward—the source presentation can continue playing, pause or be replaced [12]. In the pause case, the source presentation pauses while the destination presentation is started up. The reader can interact with the playing destination presentation or continue the source presentation. When the destination presentation replaces the source presentation, the source is removed so the only active timeline is the destination, which can be one or more linear multimedia presentations.

The continue case is more interesting, since both the source and destination presentations play simultaneously. A temporal composite is, as it were, created on the fly when the reader traverses the link. This temporal composite is determined not at document or rendering time, but at runtime, illustrated in Fig. 8. Here, the source presentation contains a list of music fragments from which can be selected. When a fragment is selected, the list remains visible and the reader can select another fragment. The reader can start up a first motif, then at a later time a second. The reader thus creates a synchronized musical composition from the available components. This is entirely dependent on the reader's interaction at runtime and is not encoded beforehand in the document or rendered presentation.

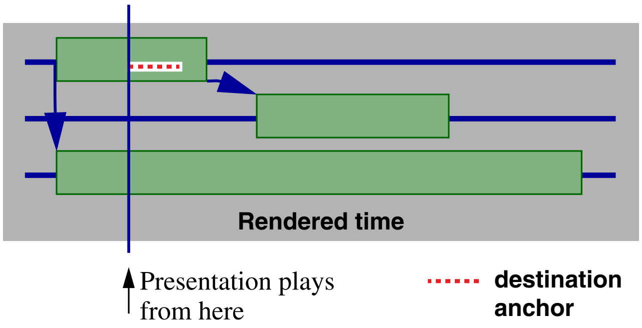

Figure 7. Link destination on rendered timeline.

There remains the question, however, of what happens when there are network delays in one of the presentations. The other presentations may wait or continue, depending on the implementation of the player software. This information too could be recorded in the link.

In SMIL, link behaviour is controlled by the `show` attribute of a link whose values are `replace`, `new` and `pause`. The behaviour of `replace` and `pause` is as described above, and that of `new` is as described in the case of continue. In the AHM, a link also contains activation states for the destination and transition information. The activation state specifies whether the destination presentation should be started up in play or paused mode.

Linking within and among non-linear multimedia presentations

The case of linking within a single, linear multimedia presentation and the case of linking among a web of linear multimedia presentations are two extremes of the potential range of interactions through link traversal. Each is an “all or nothing” approach to following the link. There are, however, other degrees of link interaction that may be useful to an author which we illustrate in the following paragraphs. We then discuss these in composition and activation terms.

An example which illustrates the limitations of the simple link model is the following. A video is played, along with some accompanying music, in the background. Displayed on top of the video is a list of text choices which the reader can browse. As the reader navigates through the text choices,

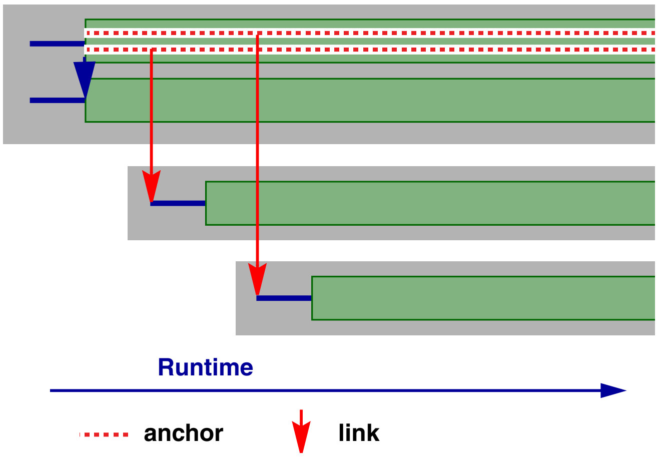

Figure 8. Following links among linked linear multimedia presentations—source continues.

reading the text, the video and music continue uninterrupted, Fig. 9.

In this example, the presentation is made up of more than one temporal composite. One of the rendered timelines continues while a reader navigates through others. There is a semantic relationship between the parts, but not a predetermined temporal one. This is similar in concept to frames in HTML, where a contents page remains on the screen while the reader can navigate among different central pages. In the HTML case, however, time is not an integral part of the document.

Uninterrupted playing of *part* of a presentation during link traversal is useful since, as the example illustrates, it allows the author to create an atmosphere which is maintained while the reader navigates detailed content. There is also a system related advantage, since information duplicated on several screens does not have to be downloaded multiple times.

The problem with defining this type of interaction is that some way is needed of isolating a part of the document so that all its temporal constraints are independent of the rest of the document. In the example, this means that navigating through the text items has no influence on the runtime behaviour of the background music and video because the behaviour of both the source and destination of the links is specified independently of the rest of the document. The parts of the document which are affected by traversal of a link are termed *source* and *destination contexts* in [12].

While this behaviour cannot be expressed in SMIL, it can be expressed using atemporal composition in the AHM. This requires, however, an extension to the link component. In addition to the information required for a link among linear multimedia presentations, links also require source and destination contexts for link anchors, controlling which parts of the presentation should remain or become active. In the example the scene is a temporal composite with two children—a temporal composite of the video and music and an atemporal composite with a number of text items as children. On following a link from a text item, a single text item or the complete scene can be replaced [12]. When a link is traversed, the temporal integrity of the presentation has to be preserved, so that the source and destination contexts have to be defined as direct children of the atemporal composite component.

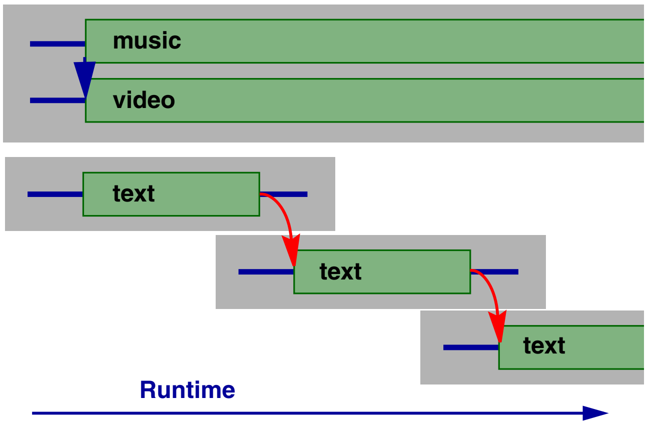

Figure 9. Actioning links among linked, non-linear multimedia presentations.

While the children of an atemporal composite can be activated through link traversal as described above, they can also be activated as part of the presentation without additional user interaction. In this case, the runtime system needs to know which of the children should be played. Thus, when grouping components into an atemporal composite, additional activation information is needed. In AHM, for every child its initial activation state (active or inactive) is specified, along with a play/pause state for continuous components. When the atemporal composite is scheduled, the subset of its children marked as active are played. The other children are played only as the result of a link traversal.

CONCLUSION

Just as the link is a first class citizen of hypertext, so time needs to become a first class citizen in time-based hypermedia. We have argued that, from multiple perspectives, time should be incorporated in a document model of hypermedia. We identified four types of time, based on the stages of creating and playing back a presentation, and on the basis of these identified a number of aspects of time that should be incorporated in a document model.

Media element time is the intrinsic time of the media elements constituting the presentation. Document time is the most interesting from the author's perspective, since it is here that the author can manipulate and store temporal information within a presentation. Rendered time is relevant from a system perspective, where the stored document is processed to give a particular version relevant to the reader. Runtime is of most interest from the reader's point of view, since it is at runtime that a reader can interact with the presentation. We discussed the interactions that the reader can carry out with a hypermedia presentation, including traversing the time axis of the presentation and following links. For the three cases of linking within and among linear and non-linear multimedia presentations we described a selection of desired interactions and their consequences for a document model.

SMIL [14] currently supports linking within and among linear multimedia presentations. One of the results of this research is to use it as input to the development of the next version of SMIL to support linking within non-linear multimedia presentations, in particular by adding constructs such as full atemporal composition, multiple destination links and transitions between elements and presentations.

ACKNOWLEDGMENTS

The work reported in this article was funded in part by the European Union ESPRIT Chameleon project. The authors would like to thank members of the SYMM working group, in particular Warner ten Kate and Philipp Hoschka for the constructive discussions throughout the development of SMIL.

REFERENCES

1. P. Ackermann (1994). Direct Manipulation of Temporal Structures in a Multimedia Application Framework. In Proceedings: *Multimedia '94*, San Francisco, CA, Oct, 51 - 58.

2. J.F. Allen (1983). Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26 (11), Nov, 832-843

3. M.C. Buchanan and P.T. Zellweger (1993). Automatic temporal layout mechanisms. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 341-350.

4. J.F. Koegel Buford (ed.) (1994). Multimedia Systems. Addison-Wesley, New York, New York. ISBN 0-201-53258-1.

5.. R. Erfle (1993). Specification of Temporal Constraints in Multimedia Documents using HyTime. *Electronic Publishing*. 6(4), 397-411.

6. Frank Halasz and Mayer Schwartz (1994). The Dexter Hypertext Reference Model. *Communications of the ACM*, 37 (2), Feb, 30-39.

7. F.G. Halasz (1988). Reflections on Notecards: Seven issues for the next generation of hypermedia systems. *Communications of the ACM*, 31(7), Jul, 836-852.

8. R. Hamakawa and J. Rekimoto (1994). Object composition and playback models for handling multimedia data. *Multimedia Systems* 2: 26-35.

9. L. Hardman (1998). Modelling and Authoring Hypermedia Documents. PhD dissertation from the University of Amsterdam. ISBN: 90-74795-93-5. `http://www.cwi.nl/~lynda/thesis`

10. L. Hardman and D.C.A. Bulterman (1997). Document Model Issues for Hypermedia. In Handbook of Multimedia Information Management, edited by W.I. Grosky, R. Jain, and R. Mehrotra. Prentice Hall.

11. L. Hardman, D.C.A. Bulterman and G. van Rossum (1994). The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37 (2), Feb, 50 - 62.

12. L. Hardman, D.C.A. Bulterman, and G. van Rossum (1993). Links in Hypermedia: The Requirement for Context. In Proceedings: *ACM Hypertext '93*, Seattle WA, Nov, 183-191.

13. L. Hardman, G. van Rossum, and D.C.A. Bulterman (1993). Structured Multimedia Authoring. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 283-289.

14. P. Hoschka (ed.) (1998). Synchronized Multimedia Integration Language. W3C Recommendation 15-June-1998. `http://w3c.org/TR/REC-smil`

15. ISO (1997). HyTime. Hypermedia/Time-based structuring language. ISO/IEC 10744:1997.

16. M. Jourdan, N. Layaida, C. Roisin, L. Sabry-Ismail, L. Tardif (1998). Madeus, an Authoring Environment for Interactive Multimedia Documents. In Proceedings: *ACM Multimedia '98*, Bristol UK.

17. M.Y. Kim, J. Song (1995). Multimedia Documents with Elastic Time. In Proceedings: *ACM Multimedia '95*, San Francisco CA, 143-154.

18. M.C. Luesebrink (1998). The Moment in Hypertext: A Brief Lexicon of Time. In Proceedings: *ACM Hypertext '98*, Pittsburgh, PA, Jun 20-24, 106-112.

19. G. van Rossum, J. Jansen, K S. Mullender and D.C.A. Bulterman (1993). CMIFed: a presentation environment for portable hypermedia documents. In Proceedings: *ACM Multimedia '93*, Anaheim CA, Aug, 183-188.

20. L. Rutledge, L. Hardman, J. van Ossenbruggen, and D.C.A. Bulterman (1998). Structural Distinctions Between Hypermedia Storage and Presentation. In Proceedings: *ACM Multimedia '98*, Bristol, UK, Sep 12-16, 145-150.