AUTHORING AND NAVIGATING HYPERMEDIA DOCUMENTS ON THE WWW

Augusto Celentano, Ombretta Gaggi

Dipartimento di Informatica, Università Ca' Foscari di Venezia
Via Torino 155, 30172 Mestre (VE), Italia
auce@dsi.unive.it,ogaggi@dsi.unive.it

ABSTRACT

This paper discusses some issues related to authoring and navigating World Wide Web documents composed of continuous and non-continuous media, based on a video or audio narration to which static or dynamic documents are attached. The discussion stems from a model able to describe synchronization among media elements and media behavior depending on user interaction. A prototype implementation addressing SMIL as a target language is described.

Keywords: Hypermedia authoring, World Wide Web, synchronization, interactive presentation.

1. INTRODUCTION

The use of multimedia components in web pages is getting more and more widespread, demanding additional work to authors since they must design not only the layout of such complex documents, but also the synchronization between different media objects. The task is more difficult when documents are interactive, since uncontrolled user interaction can alter the correct timing relationships between media.

In this paper we discuss issues related to authoring and navigating World Wide Web documents composed of continuous and non-continuous media, based on a synchronization model defined in [7, 8]. We shall briefly present the synchronization model which has two facets, related to static (structure dependent) and to dynamic (time dependent) relationships among component media objects. We shall then introduce a prototype implementation based on the SMIL language, discussing its weakness to describe non-trivial user actions that affect the relationships among the media.

This work is targeted to hypermedia presentations made of one or more continuous media file (video or audio streams) which are presented to the user. As streams play, static documents (images or text) are sent to the browser and displayed in synchrony with them. Documents can contain links to other documents. The user can interact with the hypermedia presentation in several way: by pausing and resuming it, by moving forward or backward along its timeline, or by following a link. Since a link can drive the user to a completely different context, the presentation should in some cases be paused or stopped, while in other cases it can continue playing. Media delivery and playback must be coordinated in such a way that the user always sees a coherent presentation.

2. RELATED WORK

Other research papers have presented different ways of specifying temporal scenarios, focusing on temporal synchronization of different multimedia elements.

In [1, 2] CMIFed (CWI Multimedia Interchange Format Editor), a presentation editing tool for hypermedia documents, is presented. CMIFed gives the author more flexibility allowing the use of multiple simultaneous channels in which to dispose hypermedia objects for presentation. Unlike other systems, that uses a timeline for the temporal representation, the author has to deal with a collection of events and timing constrains.

CMIFed implements the Amsterdam Hypermedia Model[2] document structure, providing three different "views" of a presentation: the hierarchy view, the channel view and the player view to preview the presentation's behavior.

In [3] an authoring and presentation tool for interactive multimedia documents named Madeus is presented. Multimedia objects are synchronized through the use of timing constrains, as spatial disposition is specify by spatial relations, such as *align* or *center*.

Document's structure, both temporal and spatial, is represented by a graph, where the horizontal axis represent time values. Graphs are not only suitable for authoring but also for scheduling and time-based navigation.

HyperProp [4] is an authoring and formatting system for hypermedia documents. In [4] the authors present a model for describing this kind of documents and a tool for generating them, once modelled. Soares et al, stress the importance of the documents' logical structure and use composition to represent any kind of relation, both temporal (synchronization relationships) or not. HyperProp offers three different graphical views to browse and edit the logical structure of the document: the structural view, the temporal view and the spatial view.

In [5] Schnepf et al present a framework designed and implemented to support authoring of hypermedia presentation according to FLIPS model. FLIPS, FLexible Interactive Presentation Synchronization, define two temporal relationships, the *barriers* to prevent an event to occur, and the *enablers* to allow it.

The authoring tool provides an interface to create and edit presentation specification using a graph of media objects connected by barriers and enablers.

3. A MODEL FOR SYNCHRONIZED HYPERMEDIA DOCUMENTS

3.1. Hypermedia document structure

The model we present here is described in greater detail in [7, 8]. It is not designed to cover efficiently all hypermedia document types, but its reference context is wide in the fame of the WWW world. Examples are Web advertising and selling, news-on-demand and professional training. During this paper we shall refer to examples

drawn from self-learning and virtual exhibitions examples. While sometimes overused, their requirements are clear and easily understood, and user interaction is of primary importance.

Hypermedia documents are modelled along two directions, the hierarchical structure and the temporal synchronization. In a self-learning application we can assume that a lesson is delivered to the user as a set of modules hierarchically organized. Each module contains a continuous stream–the audio or the video file with the teacher lesson. As it plays, other module's components, like text documents, slides and images, are delivered to the user.

A virtual museum can also be arranged as a hierarchy of environments –sections, floors, rooms–which the users navigate freely or according to a guided tour. As they walk in a virtual space, museums items are displayed, possibly at several levels of detail, and images, text, voice, sound, music, etc., provide further information (see [9] for a sample of such virtual exhibitions built for a cultural institution in Venice, Italy, during last years).

In many cases, a second continuous media stream plays in parallel with the first one, e.g. a sound track or a voice comment. In such cases, we approach only coarse-grain synchronization between the two media, and assume that fine-grain synchronization (like lip-synchronization) is coded as a multi-track media stream.

Using a video lesson as an example, we call the video stream a *story*, which is divided into a sequence of *clips* each of which corresponds to a different argument of the lesson. We impose a correspondence between the logical and the physical structure, i.e., we assume a clip is also a file which has to be played continuously unless the user interacts to modify its behavior. The correspondence is not a serious drawback, since delivered material is normally edited before being put on-line. It also improves efficiency since establishes a strict correspondence between what is delivered by the server and what is played at the user site.

Clips are divided into *scenes*, each of which is associated to one or more static documents, e.g., a text page or the image of a slide. As the scene that build up a clip play, the other documents are displayed in sequence.

We denote the static documents associated to the scenes with the generic name of *pages*. A page is a time-independent file which is displayed as a whole by the browser. A clip, with its scenes and pages associated, makes up a *section*; more sections therefore build a module.

3.2. Channels

Media objects require a portion of the screen to be laid out or played. We define a *channel* as a virtual device allocated to a document media component, like a window on the user screen, or an audio device needed to play a sound track. In general, a channel is the set of resources a media object needs for its playback; every media type requires a different channel type.

A channel cannot be shared at the same time between two or more media objects: a channel is *busy* if an active object is using it, otherwise it is *free* and can be used by another object of the same type of medium. A free channel may however hold some content, for example the last frame of a movie.

While continuous media have a duration defined by their playing time, for static media the concept of duration is fuzzy, since they do not evolve in time. We assume that a static medium has an unlimited duration, i.e., it holds the channel unless it is forced to free it. The reasons for such an assumption cannot be explained here, the reader is therefore referred to [8].

3.3. Temporal synchronization

The dynamic behavior of a presentation cannot be defined by the static structure, even if some basic play relationships are a consequence of the hierarchical organization of the presentation elements. We introduce five synchronization primitives which define the reaction of media objects to events. The events can be *internal*, like the beginning or termination of an object playback, or *external*, like a user action. In the following discussion we omit some details which can be found in [8].

To define the parallel play of two media objects A and B we introduce the relationship *"A plays with B"*, drawn $A \Leftrightarrow B$. If A or B are activated, then the other object becomes active. As soon as object A reaches its end point, B is forced to terminate too. Referring to the self-learning example, to describe that each page has to begin and to end together with the associated scene, we use the relation *scene* $\Leftrightarrow$ *page* for each pair of such objects.

This relationship defines also the inheritance of behavior between hierarchical components of a document. If a story is activated, its first clip begins, and this is described by the relationship *story* $\Leftrightarrow$ *firstclip*. The same happens between a clip and its first scene. We note that such dynamic relationships are automatically inferred from the knowledge about the static structure, therefore can be supplied by the authoring system without intervention of the designer.

The relationship *"A activates B"*, drawn $A \Rightarrow B$, defines the play of two objects in sequence. When the object A naturally ends, object B begins its playback. An object *naturally ends* when it goes through its ending point without external events. If the object stops playing as a consequence of a user interaction or another external event, we say that the object *is forced to terminate*. As we have seen above, since a static object doesn't have an ending point, it never naturally ends, but can only be forced to terminate.

In a clip, the termination of each scene causes the beginning of the next one, therefore $scene_0 \Rightarrow scene_1$, $scene_1 \Rightarrow scene_2$ and so on. The same relationships holds between the clips that build a story, e.g., a whole lesson, or a guided tour in a museum.

The relationships above do not consider the user interaction, but only the natural synchronization between media objects evolving during time. A user can stop a presentation by stopping the continuous stream which gives the overall timing. When a continuous object ends or is forced to terminate it releases its channel. This effect does not propagate naturally to other objects and channels. If the user stops a video lesson, its channel becomes free, but the channels associated to texts, images and slides remain active, leaving the whole presentation in an inconsistent state. We need a new relationship to manage the propagation.

The relationship *"A is terminated with B"*, written $A \Downarrow B$, defines that the forced termination of object A makes object B to terminate too. For example, given the relationship *video* $\Downarrow$ *slide*, if the user stops a lesson's video, the window occupied by the slide is released, and can be used by another document.

A user can move backward or forward along the time span of a presentation. This situation can be represented as a stop and a subsequent start in another point of the presentation, therefore can be modelled by the relationships introduced so far. This is not true if a user follows a hyperlink.

Let us suppose that a static document, which is displayed because it is associated with the current scene, contains a reference to another static document. If the user follows the link, the target document can be opened in a new window (i.e., a different channel) or

in the same window currently used by the referencing document. The first case does not require any further specification, since the use of a new channel does not require any synchronization with the current state of the presentation. The use of the same window requires the channel to be released by the current document.

The relation *"A is replaced by B"*, written $A \rightleftharpoons B$, allows object B to use A's channel when this object is forced to terminate. So $A \rightleftharpoons B$ causes the termination of object A if object B is activated: in our example, when the user follows the link and activates the document, the relationship *olddocument* $\rightleftharpoons$ *newdocument* causes the termination of the active static document and the release of the channel.

If the link doesn't refer to a simple static document, but to a continuous one, like a video file, or to another presentation, the relationship $\rightleftharpoons$ is no longer adequate. Let us suppose that the author defines a reference to a video clip from a document. Independently from the purpose and length of the new video clip, having two continuous independent media streams at the same time has no purpose, since the user can pay attention to only one at a time. The author therefore has to decide what the user should do with the stream that was playing at the moment when the link is followed: keep it in a paused state while the user is watching at the video clip, and resume it at a user command, or stop it, so the user can only play it again from the beginning. The choice depends on the meaning and purpose of the documents and media delivered, so we do not elaborate on the deep semantics of such definitions, but only on the consequent behavior.

In order to describe this situation we introduce the relationship *"A has priority over B with behavior* α*"*, written $A \overset{\alpha}{>} B$, where α can be P or S. If $\alpha = P$ then object B is *paused* when object A is activated, and can be resumed when object A ends; if $\alpha = S$ then object B *stops* when A is activated, and the user can only restart it from its beginning on A's termination.

3.4. An example

As an example we briefly introduce the synchronization scheme for an excerpt of the virtual exhibition "Maya" shown on the Web site of Palazzo Grassi in Venice, Italy [9]. The presentation is a guided tour into a virtual 3D reconstruction of the Maya city of Chichen Itza.

When the presentation starts a short animation takes the user to the first building. When the animation ends, a sound track begins playing, and narrative information about the building is displayed in a text frame. When the user advances to the next step a second animation takes him or her to the second building, and so on. The text frame changes accordingly, displaying information relevant to each building. The sound track continues playing or is replaced by a new one, according to author design for the different modules. Sometimes, additional sound effects are played to attract the user attention.

Figure 1 shows the synchronization relationships among the media elements that the author should define to model this presentation. The author imposes the relationship $\Rightarrow$ between animations and sound tracks to play them sequentially. In the same way, text pages are shown and additional sound effects are played at the end of each animation (*animation* $\Rightarrow$ *page* and *animation* $\Rightarrow$ *soundeffect*). When a sound track reaches its end point it continues playing, so a relationship *soundtrack* $\Rightarrow$ *soundtrack* is imposed. Some minor details are omitted from this example.

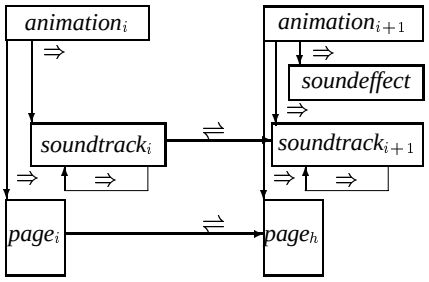

Figure 1: Media synchronization in the virtual exhibition "Maya"

4. AUTHORING SYNCHRONIZED HYPERMEDIA PRESENTATIONS

The synchronization model we have presented is well suited for a wide range of hypermedia presentations in several application domains, but it must rely on standard and efficient protocols and tools for delivery on the World Wide Web. WWW browsers are so widely used as document managers that any deviation from their use should be carefully evaluated. For this reason, the primary concern of an authoring system should be the target language for WWW delivery.

We experimented a prototype authoring tool built using the Java language and the Java Media framework API. The authoring tool helps to create hypermedia presentations designed according to the model we have discussed. The target language is SMIL [6], Synchronized Multimedia Integration Language, that is now the only realistic proposal for synchronizing different media for a coordinated presentation to be accessed in a WWW environment.

Besides the many reasons that suggest SMIL as a good target language, mainly its non-proprietary status, this choice brings also some limitations, because it doesn't support all the behaviors described by our model.

4.1. Model translation using SMIL

SMIL is a very simple markup language for presenting multimedia objects in a coordinated way. Synchronization is achieved primarily through two tags: `seq`, to play two or more objects sequentially, and `par`, to play them in parallel. These two tags allow a natural translation of the relationships $A \Leftrightarrow B$ and $A \Rightarrow B$.

The relationship $A \Leftrightarrow B$ can be translated into SMIL with the tag `par`, using the attribute `endsynch="id(A)"`. This attribute makes object B to end if object A ends by defining a synchronization point at A's end.

The relationship $A \Rightarrow B$ can be translated into the tag `seq` since it simply models the sequential composition of different media.

The relationship $A \rightleftharpoons B$ has no correspondence in SMIL, since SMIL do not defines channels as independent entities. Channels must be translated into SMIL regions, which are screen areas hosting media to be played.

It is therefore necessary to assign to object B the same region of object A to make it replace object A, thus obtaining the same result of the $\rightleftharpoons$ relationship.

The relation $\Downarrow$ cannot be implemented using SMIL native features, since SMIL only permits the user interaction with the whole

presentation, and not with the different objects within it. It is not possible to stop an object, e.g., a soundtrack, making the rest of the presentation evolve. Only the whole presentation can be stopped, paused or resumed.

Problems also arise for the relationship $A \overset{\alpha}{>} B$. SMIL models the creation of a hyperlink through the tag `a href`. This tag allows the use of the attribute `show`. If this attribute has value `pause`, it translates the relationship $A \overset{P}{>} B$, but there is no way to stop the original presentation and to display another hypermedia document in a new window. The only possibility given by SMIL is to replace the whole presentation (thus stopping it) with a new one, giving value `replace` to the attribute `show`.

4.2. System Design

The authoring tool we have built around the translation into SMIL language of the model relationships allows the design of presentation with a predefined set of components. It has mainly the purpose of demonstrating the translation process, therefore this constraint is not important. The user interface provides four channels: a video channel (which can host combined video and audio tracks), an audio channel and two window frames for displaying static documents such as text and images. In a self-learning application the video channel is dedicated to teacher's video playback, which includes the audio component, while a text channel is shared by the slides and the second text channel is used to contain links to additional educational material. This application does not use the independent audio channel.

A second application, a virtual museum tour, uses the video channel to host a 3D rendering of the museum rooms, the audio channel to play soundtracks related to the various museum environments, and the text channel to display information about the exhibition.

Authoring is divided into three phases, which correspond to three different subsystems of the authoring tool. In the first one the user can edit a video file (a clip, according to the model) to identify the scenes. The user can play the clip, mark the scene starting and ending points, revise the scenes, and possibly correct the interval boundaries.

In the second phase the user associates static documents (pages, according to the model) to each scene. A table displays how the clip is segmented into scenes. The user chooses pages and optionally a sound track to be delivered in synchrony with each scene. A list of links pointing to additional information can be associated to the second text channel.

In the third phase the SMIL file is created, providing facilities to preview the resulting hypermedia presentation and to go back to previous phases for further editing.

5. CONCLUSION

In this paper we have discussed the problem of defining synchronization relationships in a hypermedia document composed of several continuous and non continous media objects, introducing a formal model and suggesting a SMIL-based implementation.

Synchronization is achieved toward a set of primitives which are translated into SMIL tags and attributes by the tool we provide.

Three other works, presented in section 2, discuss issues very close to the one approached by our model.

Amsterdam Hypermedia Model describes hypermedia documents' temporal behavior inside objects' structure. Differently from our model, synchronization is achieved through the use of objects' composition and synchronization arches, permitting the insertion of offsets into timing relationships. Like our model, AHM defines channels to play media items.

The main difference between SMIL and our model concerns user interaction, since SMIL does not consider it as a part of its model. SMIL's native features do not allow interactions with a single object, but only with the whole document. Moreover, SMIL does not define a reference model for the data structure.

FLIPS's main differences from our model concern the system environment and the hypermedia dynamics modelling. No structure is in fact provided, other than the ones coming from the objects mutual interrelationships. Due to the absence of a hierarchical structure, the re-use of an object, or of a time span inside its timeline, is not possible.

Synchronization is defined between object's states and not between the objects themselves. Using barriers and enablers, the start or end of an object cannot directly cause the start or end of another object, but can only change the state of the object at hand.

6. ACKNOWLEDGEMENTS

The authors wish to thank Mauro Scarpa for his contribution to the work.

7. REFERENCES

[1] L. Hardman, J. van Ossenbruggen, L. Rutledge, K. Sjoerd Mullemder, D.C.A. Bulterman. CMIFed: A Presentation Environment for Portable Hypermedia Documents. *Elettronic Proceedings of the ACM Multimedia Conference*, California, USA, 1993.

[2] L. Hardman. Modelling and Authoring Hypermedia Documents. *PhD. Thesis*, University of Amsterdam, 1998. `http://www.cwi.nl/~lynda/thesis`

[3] M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismaïl, L. Tardif. Madeus, an Authoring Environment for Interactive Multimedia Documents. *Elettronic Proceedings of the ACM Multimedia Conference*. Bristol, UK, 1998.

[4] L. F. G. Soares, R. F. Rodrigues, D. C. Muchaluat Saade Modeling, authoring and formatting hypermedia documents in the HyperProp system. *Multimedia Systems*, 8(2), 2000.

[5] J. Schnepf, Y. Lee, L. Lai, L. Kang, D.H.C. Du. Building a Framework for FLexible Interactive Presentations. *Proceedings of Pacific Workshop on Distriubted Multimedia Systems*, Hong Kong, June 1996.

[6] Synchronized Multimedia Working Group of W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification. W3C Recommendation, 15 June 1998. `http://www.w3.org/TR/REC-smil`

[7] A. Celentano, O.Gaggi. Synchronization Model for Hypermedia Document Navigation. *Proceedings of the 2000 ACM Symposium on Applied Computing*, Como, 2000.

[8] A. Celentano, O. Gaggi. Modeling Synchronized Hypermedia Documents. Technical Report n. 1/2001, Department of Computer Science, Università Ca' Foscari di Venezia, Italy, January 2001, submitted for publication.

[9] `http://www.palazzograssi.it`