Multimedia Tools and Applications, 27, 53–78, 2005

Modelling Synchronized Hypermedia Presentations

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

Abstract. This paper presents a synchronization model for hypermedia presentations. Several media, *continuous*, like video and audio files, and *non-continuous*, like text pages and images, are delivered separately in a distributed environment like the World Wide Web, and presented to the user in a coordinated way.

The model is based on a set of synchronization relationships which define media behavior during presentation playback, channels in which to play them, and the effects of user interaction. The model is suited for a wide range of applications, among which self and distance education, professional training, Web advertising, cultural heritage promotion and news-on-demand are good representatives. The model is formally described in terms of changes in the presentation state due to media-related events. A virtual exhibition is analyzed as a test bed to validate the model.

Keywords: hypermedia design, interactive multimedia presentation, media synchronization, World Wide Web

1. Introduction

In this paper we present a model for designing hypermedia presentations integrating and synchronizing several media items, and supporting user interaction. Media items are files in a distributed environment, which have to be coordinated and delivered to a client, like in the World Wide Web. Documents contain continuous and non-continuous media items. Non-continuous (or static) media are text pages and images which, once displayed on the user screen, do not evolve along time. Continuous media are video and audio files, which have their own behavior.

The model is focused on a class of applications that we call "video-centered hypermedia": one or more continuous media files are presented to the user and, as streams play, other documents are sent to the client browser and displayed in synchrony with them. The user can interact with a presentation by pausing and resuming it, by moving forward or backward, and by following hyperlinks that lead him/her to a different location or time in the same document, or to a different document. At each user action the presentation must be kept coherent, resynchronizing the documents delivered.

The model addresses both temporal and spatial aspects of a web-based multimedia presentation, and arranges the media involved according to a hierarchical structure. Such a structure allows a designer to neatly organize the presentation components, and to focus on a subset of the synchronization constraints, since many of them can be automatically derived from the presentation structure. The temporal behavior is based on reactions of media items to events.

The model presented in this paper is not oriented to providing an execution language, but to designing and prototyping multimedia presentations [9]. It is a good trade-off between two main requirements: expressiveness and simplicity. The temporal behavior of a presentation is described with five synchronization relationships which define temporal constraints among the media components. The presentation is thus represented by a graph whose nodes are the media items involved and edges are the synchronization relationships.

Compared to other approaches, discussed in the paper, this model is simpler; e.g., compared to timeline-based models it is more flexible since a change in an object scheduling time does not need to be propagated to other media by explicitly redefining their behavior.

When a model is too complex it could be difficult to apply it to any particular presentation, and an authoring system may become cumbersome to use. But a simple model, supported by user-friendly tools, can become too restrictive for specifying all aspects of any multimedia presentation. We do not claim that our model allows an author to design efficiently any hypermedia document, due to the great variety of hypermedia types and structures, mainly on the World Wide Web. However, its reference context is wide, and includes applications like self and distance education, professional training, Web advertising, cultural heritage promotion and news-on-demand.

In this paper we shall refer to some sample scenarios to illustrate our model: news-on-demand, cultural heritage promotion and e-learning. We have also applied the model to the automatic generation of presentations from templates [6], and to the browsing of compound multimedia documents returned from querying multimedia information systems [7].

The paper is organized as follows: Section 2 comments the related literature. In Section 3 a model of video-centered hypermedia presentations is illustrated. Section 4 describes the primitives for synchronized delivery of different media. Section 5 discusses synchronization issues due to user interaction. Section 6 evaluates the model against other proposals discussed in the literature. Section 7 applies the model to a non-trivial Web presentation of a virtual exhibition, used as a test bed. Section 8 introduces an XML language for presentation description. Section 9 describes an authoring environment to design and easily test multimedia presentations using the proposed model, and Section 10 presents some final remarks.

2. Related work

Media synchronization in hypermedia documents has been largely investigated. In [2], Bertino and Ferrari review a number of models for synchronization of multimedia data concluding that "... much research is still needed to develop multimedia scenarios models with a complete set of features able to address all the requirements of multimedia data synchronization". Since then, several proposals have been made which approach the problem in different ways and with different perspectives.

A first class of models describes synchronization and behavior of media components of a presentation by a set of temporal relations.

Allen [1] defines a list of thirteen relationships between temporal intervals which can be used in domains where interval durations are known, but information about their occurrence is imprecise and relative. As an example, the relation *a before b* means that the media object *a* is played before *b*, but it does not define how much time is elapsed between the two objects.

In [15], King et al. present a taxonomy of seventy-two possible synchronization relationships types which provide precise timing specifications. The authors classify media objects into *synchronization events*, which are available for synchronization, and *synchronization items*, which must be synchronized.

In [22], Vazirgiannis et al. define a model based on temporal and spatial relationships between different multimedia objects that build a presentation. Given temporal and spatial starting points, the objects are set using topological and time relationships between them.

A similar approach is used in Hytime [4,17], a modular standard for expressing document architectures in SGML, preserving information about scheduling and interconnections of media components. Hytime arranges media items in a multidimensional *Finite Coordinate Space* where events are n-dimensional areas.

In FLIPS (FLexible Interactive Presentation Synchronization) [19] media objects have no predefined time length, that instead can change according to user interactions. The synchronization scheme is based on *barriers* and *enablers*. If event a is a barrier to event b then b is delayed until event a occurs. If event a is an enabler for event b, when event a occurs, event b also occurs as soon as it is barrier-free. In [20], the authors present a framework designed and implemented to support authoring of multimedia presentations according to FLIPS temporal relationships.

A second class of models uses composition to describe synchronization relationships inside a multimedia presentation.

The Amsterdam Hypermedia Model [10–12] divides multimedia document components into *atomic* and *composite*. Media items are played into channels, and synchronization in composite components is described by synchronization arches and offsets.

SMIL [21], *Synchronized Multimedia Integration Language*, is a W3C standard markup language that defines tags for presenting multimedia objects in a coordinated way. Synchronization is achieved through the composition of two tags: *seq* to render two or more objects sequentially and *par* to reproduce them in parallel. The tag *excl* is used to model some user interactions. It provides a list of child elements and only one of them may play at any given time. The screen is divided into regions in which multimedia objects are placed. SMIL 2.0[1] increases interaction and timing semantics and extends content control and layout facilities adding new animation and transition primitive.

In [13], Hardman et al. discuss their modelling solutions for incorporating temporal and linking aspects using Amsterdam Hypermedia Model and SMIL.

ZYX [3] is a multimedia document model which focuses on reuse and adaptation of multimedia content to a given user context. ZYX describes a multimedia document by means of a tree: a node is a *presentation element* (i.e., a media object or a temporal, spatial or layout relationship), and can be bound to other presentation elements in a hierarchical way. The *projector* variables specify the element's layout. With this design, an element can be easily reused, and can change its layout simply by changing the projector variable.

Many other works present different ways of specifying temporal scenarios during the authoring of multimedia documents. Madeus [14] and HPAS [23] use the metaphor of a graph, where vertices represent media objects, and directed edges represent the flow of time. In Madeus, multimedia objects are synchronized through the use of timing constraints and layout is specified by spatial relations. Graphs are used not only for authoring but also for scheduling and time-based navigation.

Synchronization between time-varying components of a multimedia document is defined also in the MPEG-4 standard [16]. In the MPEG-4 context synchronization is bound to multiplexing and demultiplexing of continuous media streams through time stamping of access units within elementary streams. Differently from the models reviewed above, MPEG-4 synchronization is more oriented to fine-grain synchronization rather than to event-based coordination of media.

3. A model of video-centered hypermedia presentations

Hypermedia presentations are modelled along two directions, one describing the hierarchical structure of the presentation components, the other describing the presentation dynamics through synchronization primitives.

A hypermedia presentation is generally composed of a set of *modules*, which the user can access in a predefined order or through some table of contents. For example, a guided tour in a virtual museum steps through a predefined selection of the museum rooms; a multimedia course is divided into lessons, which can be indexed through a syllabus or presented according to a time schedule; a news-on-demand application delivers independent articles selected from an index; an estate selling application lets the user choose the relevant property from a catalogue, and so on.

Definition 1. A module is a collection of media items, both continuous and not continuous, relating to a common topic.

We assume that from the presentation dynamics point of view each module is completely autonomous: all the media presented to the user at the same time are enclosed in the same module; user interaction can move the presentation's playback to another module. We do not elaborate further on this level of access since it is not relevant for delivery and synchronization among the different components, and assume a module as the topmost level of aggregation of media components which is relevant for our discussion.

A module is divided into *sections*: a section is a multimedia document which is normally played without requiring user intervention. Moving from a section to the next one can be automatic or not, depending on the document purpose and application context.

Definition 2. A section is a continuous media stream acting as a "master" medium, and the set of other continuous media, text pages and images which are played in parallel.

Sections could be addressable from an index as modules are, but still we shall not enter into such details. Section playback is ruled by the continuous media, with static pages delivered and displayed at specific narration points. A section is thus divided into smaller components that correspond to the time intervals between two narration points in which one or more static documents must be delivered to the user.

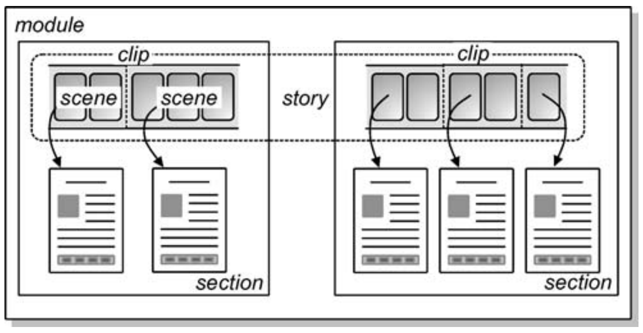


Figure 1. Structure of video-centered hypermedia document.

If we analyze the structure from a dynamic point of view, we can better refer to a terminology borrowed from a movie context.

Definition 3. A story is the continuous media stream which constitutes the main content of a module.

A *story* is made of a sequence of *clips*, each of which corresponds to a section of the module.

Definition 4. A clip is the continuous media stream which constitutes the main content of a section.

Clips are divided into *scenes*, each of which is associated with a different static document, e.g., a text page or an image, which is displayed during the scene playback.

Definition 5. A scene is a time interval in the clip's playback.

The term *narration* introduced above denotes the playback of the continuous components of a hypermedia document. Figure 1 pictorially shows this structure.

3.1. Logical vs. physical structure

The structure illustrated in figure 1 is a logical structure, which is in principle unrelated to the physical organization of the media objects. A video file could correspond to a whole clip, to a scene within it, or to a sequence of clips, that can be played continuously or as independent segments. Indeed, the correspondence between logical and physical organization should be hidden in an implementation layer, without influencing the model organization. In practice this cannot be done without paying a price in terms of performance and manageability even in a local environment, e.g., in a CD-ROM based multimedia presentation. In a distributed environment like the World Wide Web the relationships between the logical and the physical organization of the media objects are of primary concern, and must be taken into account at the application design level.

As introduced in Section 1, we distinguish between two classes of media objects: continuous media files and static document files like texts and images. Continuous media files are usually referred to as *videoclips* or *audioclips*. Static documents are usually referred to as *pages* in the WWW parlance.

Since our goal is the description of the behavior of a multimedia presentation, we model as independent entities only the media items which interact with other components of the presentation, i.e., media objects which are related by some synchronization relationship. Therefore, we do not consider the internal dynamics of components which embed animations, or short video and audio sequences (like, e.g., animated banners, logos and visual effects) unless the behavior of the embedded media items is subject to synchronization constraints.

A correspondence between the logical and the physical structure exists, based on two cases:

- a (logical) clip, i.e., the narrative content of a section, is a file which is assumed to be played continuously from beginning to end, unless user interaction modifies this behavior;
- a static document is a file which is supported and displayed as a whole by the browser, both directly or indirectly through a plug-in or helper application. We do not enter into details about the internal format (HTML, XML, PDF, etc.)

From the above assumption comes that one logical clip is implemented by one media file. Scenes are defined by time intervals in the clip: they must be contiguous and are normally played one after the other. Also pages are bound to specific contents associated with a scene. We thus can say that a scene with its associated page is the smallest amount of information that is delivered continuously as a whole, and a clip and its associated set of pages (i.e., a *section*) are the minimum self-consistent and autonomous information with respect to the application domain. According to the synchronization primitives that we shall discuss in Section 4, playing a scene causes the corresponding page to be displayed; conversely, accessing a page via a hyperlink causes the corresponding scene to be played.

3.2. Channels

Media objects require proper *channels* to be displayed or played. A channel is a (virtual) display or playback device like a window, a frame, an audio device or an application program able to play a media, that can be used by one media at a time. Several compatible media objects may share at different times the same channel. The model does not enter into details about channel properties, the only relevant property being an identifier that uniquely identifies it and a type that defines the media types that can use it.

Channels are defined and associated with media objects at design phase, and defaults are established consistently with the WWW environment: e.g., the main browser window is the default channel for all the media with visible properties. Channels can be defined as areas in the main window (e.g., frames) or as separate windows, if the presentation requires the contemporary activation of several media of the same type.

A channel is *busy* if an active media object is using it, otherwise it is *free*. Free channels may however hold some visible content, e.g., at the end of a movie playback the movie window could still display the first or the last frame.

4. Synchronization primitives

A presentation is made of different components which evolve in reaction to some events, such as user interaction, or due to an intrinsic property of components, like time length. We consider only the events that bring some change into the set of active objects during presentation playback. For example, window resizing is not a significant event, while stopping a video clip is.

In a hypermedia presentation each component plays a specific role during playback, requires channel assignment and must synchronize properly with other components. Thus we need some relationships to model object behavior and channel usage, which we call *synchronization primitives*. There are five basic relations:

- a plays with b, denoted by $a \Leftrightarrow b$
- a activates b, denoted by $a \Rightarrow b$
- a terminates b, denoted by $a \Downarrow b$
- a is replaced by b, denoted by $a \rightleftharpoons b$
- a has priority over b with behavior α, denoted by $a \overset{\alpha}{>} b$.

Some relation instances need to be explicitly defined during a presentation's authoring; others can be automatically inferred from the hierarchical structure of the presentation and from other relationships.

4.1. Basic concepts

If we observe the presentation along time, it can be divided into a number of time intervals in which some conditions hold, e.g., some media are active while other are paused. If $\mathcal{MI}$ is the set of media components which build a presentation and $\mathcal{CH}$ is the set of channels of the presentation, we can describe the presentation evolution in terms of active media and channel occupation at any time instant. We assume that time is discrete, and marked by a variable $i \in \mathbb{N}$ which is updated by a clock. The actual time resolution is not important as long as it allows the capture of all the events related to media execution, and to observe the effect of time distinct events as distinct effects. Therefore we assume that variable i is incremented in such a way that if at time i an event e occurs, at time $i + 1$ we are able to see the presentation changes due to the event, and no other event occurs before time $i + 2$. Two or more events are *contemporary* if they occur at times denoted by the same value of the variable i.

For any media item $m \in \mathcal{MI}$ the possible events are *start*(m), when the user activates the object, *pause*(m) when m is paused, *end*(m) when the object ends, and *stop*(m) when m is forced to terminate.[2]

Two functions describe channel occupation:

- *channel*: $\mathcal{MI} \rightarrow \mathcal{CH}$ which, given a media object, returns the associated channel, and
- *usedBy*: $\mathcal{CH} \times \mathbb{N} \rightarrow \mathcal{MI} \cup \{\_\}$ that returns, for every channel, the media item that occupies it at the given instant i. The underscore symbol _ denotes the absence of media item; it is the value used to identify free channels.

At any time instant, a hypermedia presentation is completely described by the set of media that are active at that time, and the corresponding channel occupation. This information is captured by the following notion of *state of a presentation*.

Definition 6. The state of a hypermedia presentation is a triple $\langle \mathcal{AM}, \mathcal{FM}, \mathcal{UC} \rangle$, where $\mathcal{AM}$ is the set of active media, $\mathcal{FM}$ is the set of paused (*frozen*) media and $\mathcal{UC}$ is the set of pairs $\langle c, m \rangle$ where c is a channel and m is the media item that occupies it as defined by the function *usedBy*.

Definition 7. A media object is *active* when it occupies a channel, otherwise it is *inactive*. Then a media item m is active in a state s if $m \in \mathcal{AM}_s$.

Definition 8. A continuous media object *naturally ends* when it reaches its end point. A generic media object is *forced to terminate* when another entity (the user or another multimedia object) stops its playback or closes its channel, before its natural end.

We distinguish between the *length* and the *duration* of a generic media object, continuous or non continuous. Every object, once activated, occupies a channel for a predefined time span that is necessary to access the object content, unless the user interacts with it. We call object *length* such time span. The length is a static property, defined at object creation.

Static objects like text pages once activated hold the channel until some event removes them. The user can access the content during a time which is not defined by the objects content. Their *length* is therefore potentially infinite.

At run-time, a user can access a media object for a time span that can be different from its length, due to user interaction or to events occurred during the presentation play. We call object *duration* this time span; it is a dynamic property defined at run-time.

4.2. Basic synchronization

The first two synchronization primitives deal with presentation behavior without user interaction.

Definition 9. Let a and b be two generic media objects. We define "a plays with b", written $a \Leftrightarrow b$, the relation such that the activation of one of the two objects (e.g., a) causes the activation of the other (e.g., b), and the natural termination of object a causes the forced termination of object b. Each object uses a different channel.

The relation $a \Leftrightarrow b$ models this behavior in the following ways:

1. If the presentation at time i is in state s such that $a, b \notin \mathcal{AM}_s$, and $usedBy(channel(a), i) = \_$, $usedBy(channel(b), i) = \_$, and event $start(a)$ or event $start(b)$ occurs, then the presentation at time $i+1$ will be in state s' such that $a, b \in \mathcal{AM}_{s'}$, and $usedBy(channel(a), i+1) = a$, $usedBy(channel\ (b), i+1) = b$.
2. If the presentation at time i is in state s such that $a, b \in \mathcal{AM}_s$, and $usedBy(channel(a), i) = a$, $usedBy(channel(b), i) = b$, and event $end(a)$ occurs, then the presentation at time $i+1$ will be in state s' such that $a, b \notin \mathcal{AM}_{s'}$, and $usedBy(channel(a), i+1) = usedBy(channel(b), i+1) = \_$ (unless used by other media as a consequence of other relations; this is a general remark and for simplicity we shall not repeat it in the following).

This relationship describes the synchronization between two media active at the same time. If a video clip v goes with a static page p or a continuous soundtrack st, relations $v \Leftrightarrow p$ and $v \Leftrightarrow st$ mean that the video and the page or the soundtrack start together, and when v naturally ends the page and the soundtrack are also ended.

The relation "plays with" is asymmetric: object a plays the role of master with respect to slave object b. The master object is usually the one the user mostly interacts with. For example a video clip played with an accompanying text is the master because the user can pause, stop, or move inside it causing the accompanying text to be modified accordingly.

It is important to note that the "plays with" relation can't provide fine-grain synchronization (e.g., lip-synchronization) because it defines the media mutual behavior only at starting and ending points. This apparent limitation is consistent with the application domain of interest of the model.

Definition 10. Let a and b be two generic media objects. We define "a activates b", denoted by $a \Rightarrow b$, the relation such that the natural termination of object a causes the beginning of playback or display of object b. The objects can share the same channel or not.

If the presentation at time i is in state s such that $a \in \mathcal{AM}_s, b \notin \mathcal{AM}_s$, and $usedBy(channel(a), i) = a$, $usedBy(channel(b), i) = \_$, and event $end(a)$ occurs, then at time $i+1$ the presentation will be in state s' such that $a \notin \mathcal{AM}_{s'}, b \in \mathcal{AM}_{s'}$, and $usedBy(channel(b), i+1) = b, usedBy(channel(a), i+1) = \_$ (if $channel(a) \neq channel(b)$).

The relationship "activates" describes two objects that play in sequence. We limit the scope of this relationship to the natural termination of an object; the reason for doing so will be more evident after discussing other synchronization relationships, but we can anticipate that we interpret the forced termination as an indication to stop the presentation or a part of it. Therefore, such an action must be defined explicitly and should denote a deviation from the "normal" behavior of the presentation.

While more meaningful if applied to continuous objects, the $\Rightarrow$ relationship can define the sequential play of continuous and non continuous objects, e.g., a credit page after the end of a video sequence. It should be obvious that object a in the relation $a \Rightarrow b$ must be a continuous media object, because static media objects have an infinite length.

Simple timeline-based presentations can be described by using only "plays with" and "activates" relations (an example is illustrated in [5]) but more complex dynamics, involving also user interaction, require additional relationships which are described in the following Section.

5. User interaction

Users have several possibilities of interacting with a hypermedia presentation: they can stop playing a part of it and skip further, or can branch forward or backward along the time span of a continuous medium, and the presentation components must resume synchronized playing after the branch. Users can leave the current presentation to follow a hyperlink, either temporarily, resuming later the original document, or definitely, abandoning the original document and continuing with another one.

We define three synchronization primitives to model the presentation behavior according to user interaction.

Definition 11. Let a and b be two generic media objects. We define "a terminates b", written $a \Downarrow b$, the relation such that the forced termination of object a causes the forced termination of object b. The channels occupied by the two media objects are released.

The relation $a \Downarrow b$ models the reaction to the event *stop*(a). If the presentation at time i is in state s such that $a, b \in \mathcal{AM}_s$, and *usedBy*(*channel*(a), i) $= a$, *usedBy*(*channel*(b), i) $= b$, and event *stop*(a) occurs, then at time $i+1$ it will be in state s' such that $a, b \notin \mathcal{AM}_{s'}$, and *usedBy*(*channel*(a), $i + 1$) $=$ *usedBy*(*channel*(b), $i + 1$) $= \_$.

The relationship "terminates" models situations in which the user stops a part of a presentation, and the media must be re-synchronized accordingly. As an example, let us consider a multimedia presentation for travel agencies: a video clip a illustrates a town tour and a text page b describes the relevant landmarks, with $a \Leftrightarrow b$. If the user stops the video, object a terminates and releases its channel. Object b remains active, because relationship $a \Leftrightarrow b$ is meaningful only when a comes naturally to its ending point. Therefore the channel used by object b remains busy leading to an inconsistent situation. The relation $a \Downarrow b$ removes the inconsistency: propagating the forced termination of object a to object b, b releases its channel which can be used by another document. Similarly to the relationship $a \Leftrightarrow b$ the relation $a \Downarrow b$ is asymmetric.

It is worth to discuss why we have introduced in the model the relationship "terminates" instead of extending the relationship "plays with" to deal with *any* termination of an object. Should we have done so, the model would not be complete, i.e., some synchronization requirements could not be described. Several other approaches in fact, e.g. SMIL, do not distinguish the natural termination from the forced stop of an object, which are considered the same event. In our model they are two different events, which can bring to different paths of the presentation evolution. An example will illustrate this statement.

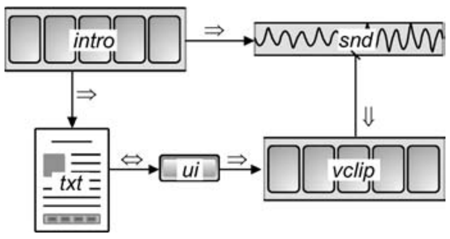


Figure 2. The relation $\Downarrow$.

Let us consider a presentation which begins with a short introductory animation. As the animation ends, a sound track starts playing, and the presentation displays a page which asks the user to chose among different video clips to continue.[3] The sound track doesn't stop when the user selects a video clip, but continues in the background. Figure 2 pictorially shows this situation: the objects involved are the first animation *intro*, the sound track *snd*, the text page *txt* and a video clip *vclip*. We introduce also an additional media object *ui*, standing for *user interaction*, which is an object whose temporal behavior is controlled by the user. We consider it a continuous object with a variable length, which *naturally ends* when the user activates it, e.g., with a mouse click.

The introduction starts both the page and the sound track by virtue of the relationships *intro* $\Rightarrow$ *snd* and *intro* $\Rightarrow$ *txt*. Object *ui* is activated together with the text page (*txt* $\Leftrightarrow$ *ui*) and ends when the user selects the video clip. If the user wants later to leave the presentation, he or she stops the video clip *vclip*, which is the foreground media object. It must stop the sound track, and this action cannot be described without a specific relationship between the two objects, that are otherwise unrelated. We have to set the relation *vclip* $\Downarrow$ *snd*.[4]

The relationship "terminates" also allows us to implicitly define the behavior of a presentation when a user branches forward and backward along the presentation time span. We consider a "branch" a movement inside the same multimedia presentation. If a user moves to another hypermedia document, we do not consider it a branch but the activation of a hypermedia link, which will be handled in a different way.

The presentation is stopped at the point in which the user executes the branch command, and starts again at the point the user has selected as the branch target. The synchronization primitives introduced so far can handle this case. The target of a branch can be the beginning of an object (a continuous object), but can also be any point contained in its length. This is possible if the user interface provides some mechanisms to navigate along the time span of an object, i.e. a cursor, or a VCR-style console. From the synchronization point of view, the relations are associated to the activation of a media item as a whole, so each point inside the length of the object is equivalent to its starting point. Then, the relationship "plays with" ($\Leftrightarrow$) is associated to the activation of the master object, regardless of *where* it starts, so it is able to activate the associated slave object even if the master starts from a point in the middle of its duration.

When a user follows a hyperlink to another presentation, modelling requires additional care: we must define the hyperlink source and destination, and the behavior of the objects involved in the link. The source of a hyperlink is the object which contains it, but the author can define some synchronization primitives which describe the behavior of the presentation when the user follows that link, possibly affecting other media objects. Therefore, all the objects active at the time of the user interaction are considered as the link source. In the same way, the destination of the link is the linked object, but the presence of some synchronization relations can involve other media items. Therefore, we consider the set of active objects after the user action to be the destination of the hyperlink.[5]

We can easily distinguish three cases, which need to be modelled in different ways:

– following the hyperlink does not imply any change in the continuous objects of the presentation,
– following the hyperlink takes a continuous component of the hypermedia document to a paused state, and
– following the hyperlink stops a continuous object of the presentation.

Let us consider the first case, taking as an example a self-learning application in which a video and some slides are displayed together. One of the slides contains a link to another text document with supplemental information. If the document is short, the user doesn't need to stop the video in order to read the text without loosing information. This example is illustrated in figure 3. Two video clips c_1 and c_2 are associated each with a text page (respectively p_1 and p_2). Page p_1 contains a hyperlink to page p_3. The author of the application can model the display of the linked page p_3 in two ways: opening another channel, or using the same channel of p_1, that must be released purposely. In the first case the link causes a new channel to be allocated, without any consequence on the other media synchronization. In the second case, we need to introduce a new synchronization primitive, as illustrated in figure 3.

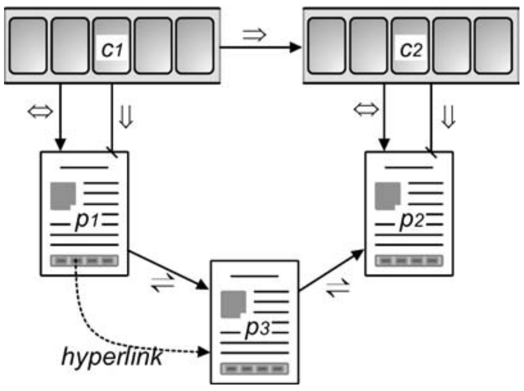


Figure 3. Following a hyperlink to a static object.

Definition 12. Let a and b be two media objects of the same type (i.e. two continuous media or two non continuous media) that can use the same channel. We define "a is replaced by b", written $a \rightleftharpoons b$, the relation such that the activation of object b causes the forced termination of object a. The channel held by a is released to be used by b.

If the presentation at time i is in state s such that $a \in \mathcal{AM}_s$, $b \notin \mathcal{AM}_s$, and *usedBy* $(\textit{channel}(a), i) = a$, $\textit{channel}(a) = \textit{channel}(b)$, and the event *start*$(b)$ occurs, then at time $i+1$ the presentation will be in state s' such that $a \notin \mathcal{AM}_{s'}$, $b \in \mathcal{AM}_{s'}$, and $\textit{usedBy}(\textit{channel}(a), i+1) = b$.

In figure 3 the relationship $p_1 \rightleftharpoons p_3$ allows page p_3 to be displayed in the same window of page p_1, that is therefore terminated. In the same way, $p_3 \rightleftharpoons p_2$ when later clip c_1 ends and c_2 starts playing.

If page p_1 contains a link to a large document or to another video clip the user should need to pause the initial presentation in order to pay attention to the new document. Figure 4(a) pictorially shows this case: page p_1 contains a link to a new video clip (c_3) that is associated with text page p_3.

The user also can decide to stop the presentation and to continue reading the new one, or the author could have designed a set of multimedia documents with this behavior in mind. Going back to the abandoned presentation in this case would be possible only by restarting it from some defined starting point. It should be clear that the user or author behavior depends on the meaning of the linked documents. In principle the synchronization model should be able to describe both cases.

Definition 13. Let a and b be two generic media objects. We define "a has priority over b with behavior α", written $a \overset{\alpha}{>} b$, the relation such that the activation of object a (by the user, or according to the presentation's design) forces object b to release the channel it occupies, so that object a can use it if needed .

Label α denotes object b's behavior once it has released the channel. If $\alpha = p$ then object b goes into an inactive state (i.e. it *pauses*), waiting for being resumed. If $\alpha = s$, object b is forced to terminate (it *stops*), releasing the resource it uses.

If event *start*(a) occurs at time i when the presentation is in state s such that $a \notin \mathcal{AM}_s$, $b \in \mathcal{AM}_s$, and $\textit{usedBy}(\textit{channel}(a),i) = \_$ (if $\textit{channel}(a) = \textit{channel}(b)$), $\textit{usedBy}(\textit{channel}(b), i) \neq b$, then at time $i + 1$ the presentation will be in state s' such that:

1. if $\alpha = s$, $a \in \mathcal{AM}_{s'}$, $b \notin \mathcal{AM}_{s'}$;
2. if $\alpha = p$, $a \in \mathcal{AM}_{s'}$, $b \in \mathcal{FM}_{s'}$;

in both cases, $\textit{usedBy}(\textit{channel}(a), i+1) = a$, $\textit{usedBy}(\textit{channel}(b), i+1) = \_$ (if $\textit{channel}(b) \neq \textit{channel}(a)$).

In the case illustrated by figure 4(a), we draw the relation $c_3 \overset{p}{>} c_1$ so that following the hyperlink activates c_3 that leads c_1 into an inactive state. The channel used by c_1 is released to be used by c_3. From the relation $c_3 \Leftrightarrow p_3$ we can assume that page p_3 must be displayed into p_1's channel. Therefore an additional relationship $p_1 \rightleftharpoons p_3$ must be added.

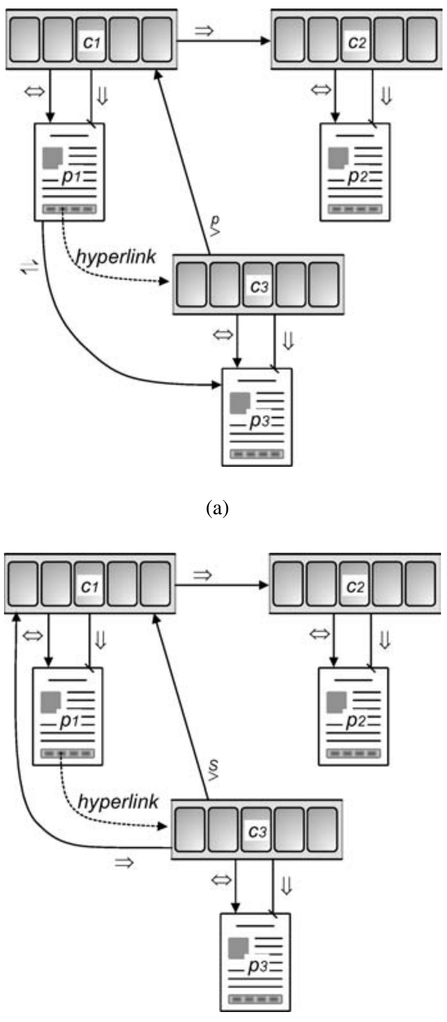


Figure 4. Following hyperlinks to continuous objects: use of $\overset{p}{>}$ and $\overset{s}{>}$ relationships.

When c_3 terminates the user can resume c_1 from the point where it was suspended. Since the relation $c_1 \Leftrightarrow p_1$ is active for the duration of c_1, page p_1 is also activated, so the presentation goes back to the state it was before the hyperlink activation. The channel that was used by p_3 is free due to the two relationships $\Leftrightarrow$ and $\Downarrow$ between c_3 and p_3.

If the author decides to stop the first document before leaving it, the relationship $c_3 \overset{s}{>} c_1$ must be used instead of relation $\overset{p}{>}$, as illustrated in figure 4(b) The relation $p_1 \rightleftharpoons p_3$ introduced in figure 4(a) is not necessary because, since c_1 is forced to terminate, by relation $c_1 \Downarrow p_1$ also page p_1 is forced to terminate, releasing the channel that can be used by p_3.

In figure 4(b) it is assumed that when c_3 terminates, the clip c_1 starts again, as described by the relationship $c_3 \Rightarrow c_1$. A different behavior could be to let the user start again the stopped presentation.

6. Comparison with other approaches

Some works, reviewed in Section 2, discuss issues close to the ones approached by our model.

Allen's approach [1] defines a set of thirteen relationships between temporal intervals of known length. In our model the length of an object is the time span from its beginning to its natural end, but its duration is known only at run-time. Therefore the Allen model cannot be used to describe the actual relationships between any two media in the general case. For example, the relation $a \Leftrightarrow b$ of our model corresponds to two different cases in Allen's notation:

– if the duration of a is less or equal to the duration of b, the corresponding Allen relation is *a equal b*;
– otherwise the corresponding Allen relation is *a starts b*.

Other problems could arise when dealing with user interactions: as an example, the relationship *a meets b*, which naturally translates our synchronization primitive $a \Rightarrow b$, does not capture the possibility of the user to stop a thus preventing object b from starting. The same issues can be addressed for the works discussed in [15] and [22]. More generally, Allen's model captures the relationships between two media items when their execution is known, therefore cannot be used as a design model.

The main differences between FLIPS [19, 20] and 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 a section of a presentation is not possible.

In FLIPS synchronization is defined between the object 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. For example, the beginning of an object, which corresponds to its activation in our model, depends upon the absence of barriers that can be set or reset by complex conditions. Our model is simpler since a state change in an object is caused only the user or by events associated to a related object, independently from other presentation components. Moreover, FLIPS does not address presentation layout but it only deals with synchronization issues.

The Amsterdam Hypermedia Model [12] describes the temporal behavior of hypermedia documents inside the objects structure. Differently from our model, synchronization is achieved through object composition and synchronization arches, permitting the insertion of offsets into timing relationships. The authoring paradigm is therefore different: AHM integrates information about the presentation structure with information related to its temporal behavior, that our model keeps separate.

The main difference between SMIL [21] and our model concerns the lack of a reference model for the data structure in SMIL. Our model organizes media objects into a hierarchical structure, which is useful to design complex presentation. For example, it can be used to infer temporal relationships between media (e.g., the scenes of a clip are played sequentially).

The XML language which will be presented in Section 8, clearly separates spatial and temporal relations from references to media objects in three separate sections. A media item can be referenced several times without redundancy by addressing its *id*. Thus, an author can easily reuse the structure of another presentation, a single media item, an entire document or a part if it. In SMIL, instead, the two types of information are interleaved in the document, possibly generating redundancy.

Other differences between SMIL and our model can be found in the way actions directed to end media executions are managed. Like Allen's relationships, SMIL native features do not distinguish between natural and forced termination of a media, therefore the effects of a user interaction on a single media component are not always easy to describe.

7. Model testing: The Maya's sun temple

In this section we discuss a quite long example in order to prove that the model is able to describe non trivial presentations with many synchronization requirements. We analyze a multimedia presentation designed for a virtual exhibition, namely the *Maya Sun Temple* section of the *Maya* exhibition at Palazzo Grassi, Venice, Italy, which is available on the World Wide Web [18].

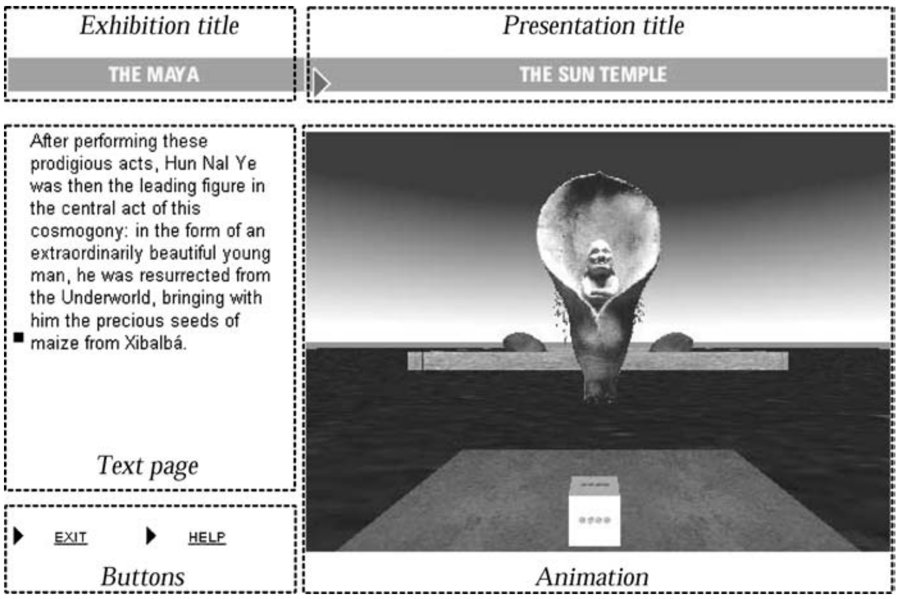
The presentation is a narration of the Maya cosmogony illustrated by a tour in a 3D virtual world. The myth, engraved on the temples of Palenque (the Sun Temple) tells how the First Father, called Hun Nal Ye (A Grain of Maize) was born in 3114 BC. At that time, the Sun did not exist and darkness reigned over everything. Hun Nal Ye built himself a house in a place called the Higher Heaven.

As the narration goes on, a 3D reconstruction of the Sun Temple is presented. By clicking on a dice engraved with Maya numbers the user can step through the building, exploring pictures and handicrafts inside it. Text pages explain habits and customs of Mayan civilization.

The user interface is divided into five regions (figure 5). Two regions are dedicated to the exhibition and to the presentation titles. Since the information contained does not change during presentation playback, we ignore them. A region in the lower left corner of the screen contains two buttons: *help* and *exit*. Help information is displayed in the text pages area. Exiting the presentation takes the user back to Maya exhibition's home page. In order to limit the size of the discussion we ignore these two elements, that do not change during the whole presentation.

We focus our analysis on the animation, the text pages that are displayed in the left side of the screen, and the sound tracks. We consider four channels: *an* for the animation, *text* for the text pages, *sound* for the sound tracks and *noise* for some audio effects that are played during the presentation.[6]

Maya sun temple's presentation contains twenty animations, each of which is a clip in our model's structure. Some clips are played alone, i.e., they are not associated with any



After performing these prodigious acts, Hun Nal Ye was then the leading figure in the central act of this cosmogony: in the form of an extraordinarily beautiful young man, he was resurrected from the Underworld, bringing with him the precious seeds of maize from Xibalbá.

Figure 5. Maya Sun Temple presentation interface [18].

text document. Different sound tracks are played during the presentation, in order to make more evident the progression of the narration. With reference to the model, text documents are pages, while the sound tracks are clips. The presentation is organized in five modules. We also consider an initial module, M_0, which contains only a text page, p_0, which begins to tell the cosmogony story, and the image of a dice playing the role of a button, ui_0, acting as a user interaction device to step through the narration. The dice is placed in the animation channel an. Channel $sound$ is initially free, while channel $text$ and an are busy. Tables 1 and 2 lists the elements of modules M_0–M_2, which are illustrated in this section.

Table 1. Maya Sun Temple presentation elements, Modules 0 and 1.

M_0	Initial module.
p_0	The myth engraved on the temples of Palenque recounts how the First Father…
ui_0	(Maya number 0)
M_1	Hun Nal Ye house in Higher Heaven.
s_1	Wind sound track.
a_1	House foundation.
p_1	At the site of the house, the mythical ancestor also set up three stones that symbolized the three vertical levels …
ui_1	● (Maya number 1)

Table 2. Maya Sun Temple presentation elements, Module 2.

M_2	Hun Nal Ye brings the life in the world (representation of monumental art).
s_2	Birds singing sound track.
n_1	Noise due to God's resurrection.
a_2	Tree and stones raising.
$\vdots$	$\vdots$
a_8	House view.
ui_2	●● (Maya number 2)
$\vdots$	$\vdots$
ui_8	●●● ▬ (Maya number 8)
p_2	After performing these prodigious acts, Hun Nal Ye was then the leading figure ...
$\vdots$	$\vdots$
p_7	Depending upon the occasion, these deities would appear in different forms ...
p_b	Blank page.

When the user starts the presentation by clicking on the dice, he or she starts module M_1: we set the relation $M_0 \rightleftharpoons M_1$ so that the first module can use the channel *text*. As a consequence of M_0 termination, page p_0 must terminate too, so we set $M_0 \Downarrow p_0$.

The first animation begins by building Hun Nal Ye house foundations, while the user hears wind blowing. Calling the animation clip a_1 and the sound track s_1, we model this behavior with the relations $M_1 \Leftrightarrow a_1$ and $M_1 \Leftrightarrow s_1$, where a_1 occupies the channel *an* and s_1 the channel *sound*. The wind sound is a short audio file that loops, so we set $s_1 \Rightarrow s_1$.

At the end of the first animation a text page p_1 is displayed in channel *text*. The relation $a_1 \Rightarrow p_1$ models this behavior. Wind sound track s_1 continues playing, while channel *an* is released.

At the end of each animation the presentation pauses, waiting for user interaction: a dice engraved with Maya numbers is displayed and the user must click on it to enter the next animation. When a_1 terminates, ui_1 begins by virtue of the relation $a_1 \Rightarrow ui_1$. When the user clicks on the dice, ui_1 ends: the next animation begins playing as a consequence of relations $ui_1 \Rightarrow M_2$ and $M_2 \Leftrightarrow a_2$.

If the user decides to exit the module, using the button "Exit", all active objects should stop: they are a_1 and s_1, during animation's playback, and s_1 and p_1, if a_1 is already ended. The relations $M_1 \Downarrow a_1$, $M_1 \Downarrow s_1$ and $s_1 \Downarrow p_1$ describe this behavior. Figure 6 shows the first two modules.

In all the modules the dice with Maya numbers represents an activation point for the next animation, or the next module if the animation currently playing is the last one of the module. For each module M_k the end of animation a_i activates the user interaction entity ui_i, which starts the next animation when clicked: relationships $a_i \Rightarrow ui_i$ and $ui_i \Rightarrow a_{i+1}$ model this behavior.

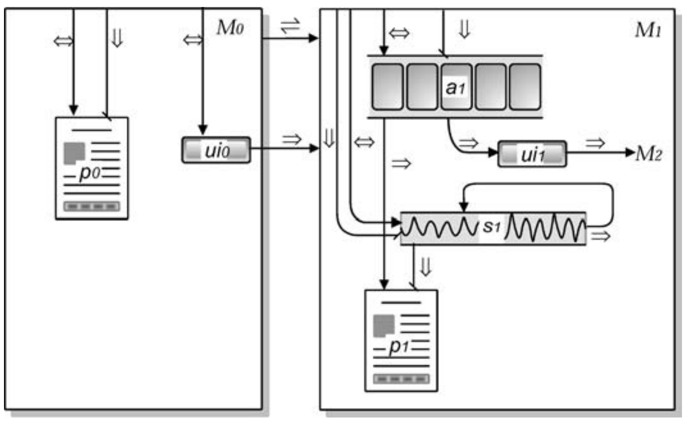


Figure 6. Synchronization in the first two modules.

At the end of the last animation of a module M_k the user goes to module M_{k+1}. The relation $M_k \rightleftharpoons M_{k+1}$, set for every module but the last one, forces the whole module M_k to terminate in order to release its channels, as defined by the relations $\Downarrow$ between the module and its components.

The narration of Maya cosmogony continues in module M_2 with the beginning of the life in the world. Hun Nal Ye comes in the world like a beautiful young man, bringing with him the seeds of maize. To emphasize this event, the sound track plays a sound of singing birds. Then the presentation gives the user some information about Maya monumental art and religious view of life and world.

Module M_2 has a much more complex structure. It contains seven animations, seven pages and two sound clips. They share the same channels used by module M_1.

When the user begins M_2's playback, animation clip a_2 adds some stones and a tree to the god's house, and a sound of singing birds begins (s_2). When a_2 ends, page p_2 is displayed. Sound track s_2 plays for the entire duration of M_2, because it loops continuously. The following relationships model this situation:

$$\begin{array}{lll} M_2 \Leftrightarrow a_2 & a_2 \Rightarrow a_2 & M_2 \Downarrow a_2 \\ M_2 \Leftrightarrow s_2 & s_2 \Rightarrow s_2 & M_2 \Downarrow s_2 \\ & & s_2 \ \Downarrow p_2 \end{array}$$

At the end of animation a_2, clicking on the dice starts animation a_3, which uses channel *an* released by a_2. Page p_2 remains active.

As a_3 begins the user hears, together with sound s_2, also a noise, which represents the resurrection from the underworld of Hun Nal Ye. Denoting this sound clip with n_1, we establish the relation $a_3 \Leftrightarrow n_1$. n_1 uses channel *noise*, while the soundtrack s_2 continues playing in channel *sound*.

As for animation a_2, if the user stops the module's playback, both animation and noise should stop, therefore $M_2 \Downarrow a_3$ and $a_3 \Downarrow n_1$. The same relation has to be set between the

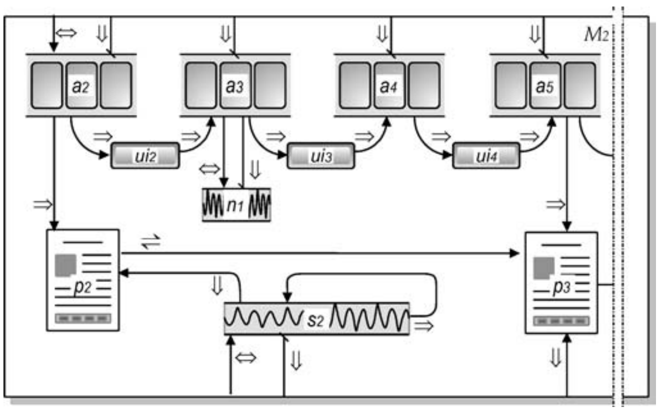


Figure 7. Synchronization in module M_2 (first part).

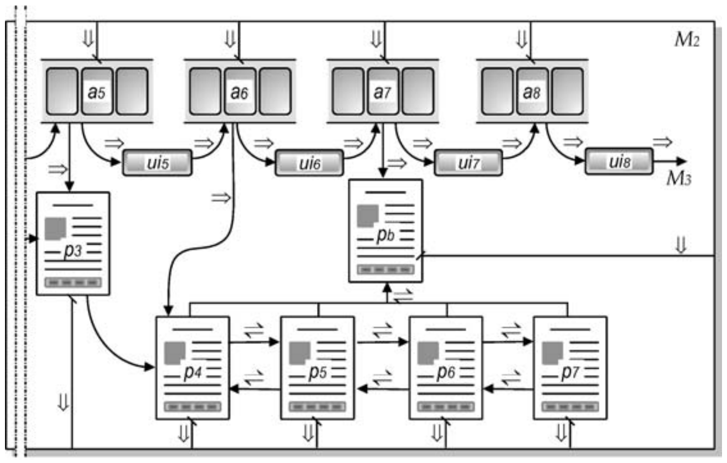


Figure 8. Synchronization in module M_2 (second part).

module and all the animations that make up the story, and $M_1 \Downarrow a_8$, a_4–a_8, as figures 7 and 8 pictorially show.

Page p_2 is displayed until animation a_5 ends, then it is replaced by page p_3, as described by the relations $a_5 \Rightarrow p_3$ and $p_2 \rightleftharpoons p_3$.

When animation a_6 ends, it activates page p_4 by virtue of the relation $a_6 \Rightarrow p_4$. Page p_4 contains a link to another text page, p_5, which contains two hyperlinks, one back to p_4, and one to p_6 (see Table 2). Pages p_4 to p_7, and their hyperlinks, form a bidirectional list, as shown in figure 8. We introduce the relationship "is replaced by" ($\rightleftharpoons$) whenever there is a link between two objects that share the same channel, to describe that the playing object must release the current content. Relations $p_4 \rightleftharpoons p_5$, $p_5 \rightleftharpoons p_4$, and the others shown in the figure, manage the use of the channel *text*. Relations $M_1 \Downarrow p_4$ up to $M_1 \Downarrow p_7$ are introduced to terminate the active page when the user stops the presentation.

Animation a_7 ends by showing the god physical aspect, then the page displayed in channel *text* has to be removed. When a page is terminated, the channel is released but the page content remains visible, i.e., a Web browser displays a page until a new content is downloaded. Therefore a blank page p_b is displayed in channel *text*, as described by relation $a_7 \Rightarrow p_b$.

8. An XML schema for multimedia presentations description

The hierarchical structure and the temporal relationships defined in our presentation model are described in an XML schema [8]. An XML source document describing a hypermedia presentation is generated by a visual authoring tool in which the author defines the layout and the synchronization relationships [9].

We store hypermedia presentation structure and spatio-temporal information separately from reference to multimedia data. In most existing model this information is often intermixed. For example, in SMIL spatial information are separated in the *head* section, but the temporal axis includes the media objects declaration. This integration does not encourage objects reuse, which is useful mainly when documents structure becomes complex, and often generates redundancy.

The XML document contains three types of data: the spatial layout of the document, the media involved in the presentation and their temporal behavior. They are described in three different sections, the layout section, the components section and the relationships section.

The layout section contains the definition of the channels used by the presentation and the size of the presentation windows. Figure 9 shows the layout section of the presentation illustrated in Section 7.

The an and the text channels are portions of the user screen delimited by the coordinates of the corners, SupX, SupY, InfX and InfY. Sound and noise are audio channels, therefore they have no visible layout. Each channel has a unique name.

The components section (figure 10) contains the description of all the media objects involved in the presentation organized around the concepts of module, story, clip, scene and page, as described in Section 3. Each element has a unique identifier id, which is used to reference it from other sections, and a type (but for modules), which is one of video, audio or animation for continuous media, and text or image for the pages. The type is inherited from a story to its clips and scenes.

The attribute channel is required for clips and pages. Since a story contains at least one clip, the channel can be defined in the story or in each clip of the story. A scene inherits the channel from the clip.

```
<layout width="500" height="400">
  <channel name="an" SupX="158" SupY="2" InfX="495" InfY="394"/>
  <channel name="text" SupX="0" SupY="2" InfX="155" InfY="325"/>
  <channel name="sound"/>
  <channel name="noise"/>
</layout>
```

Figure 9. Layout section of the Maya Sun Temple presentation.

Table 3. XML representation of the synchronization primitives.

$A \Leftrightarrow B$	<play><master><cont_object id="A"/></master> <slave><object id="B"/></slave></play>
$A \Rightarrow B$	<act><ended><cont_object id="A"/></ended> <activated><object id="B"/></activated></act>
$A \rightleftharpoons B$	<repl><before><object id="A"/></before> <after><object id="B"/></after></repl>
$A \Downarrow B$	<stop><first><object id="A"/></first> <second><object id="B"/></second></stop>
$A \overset{\alpha}{>} B$	<link behaviour = α ><from><object id="A"/></from> <to><object id="B"/></to></link>

```
<components>
  ...
  <module id="M2">
     <clip id="an2" channel="an" type="animation" file="an2.swf" >
     <clip id="an3" channel="an" type="animation" file="an3.swf" >
      ...
     <clip id="n1" channel="noise" type="audio" file="n1.wav" >
     <clip id="s2" channel="sound" type="audio" file="s2.wav" >
     <page id="p2" channel="text" type="text" file="p2.txt" />
   </module>
   ...
<components>
```

Figure 10. An excerpt of the component section of the Maya Sun Temple presentation.

Clips and pages always correspond to a file, and the attribute file refers to the actual multimedia data URL.

The relationships section describes the temporal behavior of objects, by defining the list of synchronization relationships of the presentation.

Each relationship type is coded with a different tag as shown in Table 3, enclosing two children tags for the two sides of the relationship; an optional attribute from_struct indicates whether the relationship is derived from the hierarchical structure of the presentation or explicitly set by the author.

The complete schema of the XML language for the model is described in http://www.dsi.unive.it/~ogaggi/xml/model.xsd.

9. A prototyping environment

We have implemented an authoring environment called *LAMP* (LAboratory for Multimedia presentations Prototyping) which allows an author to set up and test a complex multimedia presentation by defining the synchronization relationships among media according to the synchronization model presented in this paper.

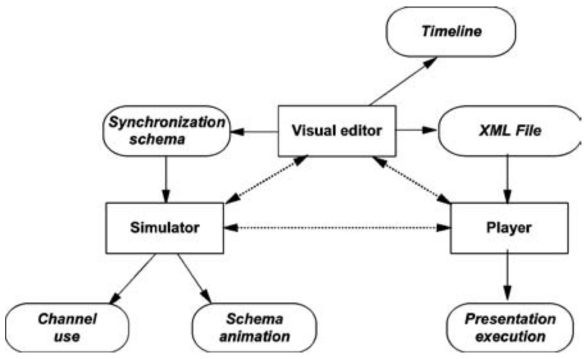


Figure 11. The *LAMP* authoring environment.

The authoring system components are a visual editor, an execution simulator to test the presentation behavior on different media-related and user-related events, and a player for the actual execution of the complete presentation (figure 11). A detailed description is in [9].

The visual editor is based on a graph notation very similar to the one used in figures 2–4, in which the nodes are media objects and the edges are synchronization relations between them. Screen layout and playback channels are visually arranged by drawing rectangles inside the presentation window.

An execution simulator interprets the synchronization graph and allows the author to check the temporal behavior of the presentation. It does not require the actual media file to be available, using placeholders if they are missing. Placeholders are allocated to the channels the corresponding media would use in the real execution. Then, without being compelled to follow a real time scale, the author can generate all media related events, both internal (e.g., the end of an object play), and external (e.g., a user-executed stop or a hyperlink activation) to see how the synchronization relationships are fired and how the presentation evolves. In order to help the author to understand the relationships between events and media, the simulator animates also the graph of the presentation: when the user activates an object, the corresponding node in the graph is highlighted. The relations triggered by the activation of the object are also highlighted, and their effect is propagated to the related objects, giving the author, in a short animation, the visual perception of how media are synchronized.

The visual editor can generate several external representations: an XML-based description is used by the player, while a representation close to a timeline highlights media sequencing, showing the overall execution structure at a high level of detail.

A player reproduces the presentation to final users. It can be used as a stand-alone application or during the authoring phase, since it can interact with the simulator by visually relating the execution evolution with an animation of the synchronization schema.

10. Conclusion

In this paper we have presented a model for describing hypermedia presentations for distributed environments like the World Wide Web. A presentation is modelled along two directions: document structure and temporal synchronization. The model defines a hierarchy of components: a continuous media stream is a story, which is made of clips, divided into scenes; a clip, and the pages related to its scenes, build up a section. Synchronization is achieved with a set of relationships among the components of a multimedia presentation, while spatial positioning is obtained through the definition of playback and display channels.

Our model is oriented to describe applications based on a main video or audio narration, to which static and dynamic objects are synchronized. As a verification of the model potentialities, we have modelled a non trivial web-based presentation.

An XML language is defined to store hypermedia documents in an application-independent and machine-understandable format dividing temporal and spatial relationships from data definition.

An authoring and prototyping tool for generating hypermedia presentation descriptions according to the model has been developed, which allows the designer to simulate the presentation behavior under different combinations of media related events, e.g., different durations of media objects, user actions on media playback, hyperlink activations, etc.

Besides authoring, the model has been used in two applications. The first addresses the automatic generation of *multimedia reports*, i.e., presentations defined according to a repeating template, whose content is defined by querying one or more multimedia repositories [6]. The second application is in the area of retrieval of distributed multimedia presentations [7]. We show how this model can be used for building coherent answers to information retrieval queries to multimedia data repositories where complex presentations are stored.

A promising research direction for our future work is the extension of the model to the mobile applications, where distributed hypermedia presentations can be played on portable devices and change their behavior according to the user context, like the user location or the set of resources available.

Acknowledgments

The authors acknowledge the contribution of Diego Medici, Daniele Dal Mas and Alessandro De Faveri for the implementation of the authoring environment. Maria Luisa Sapino has contributed to the formal definition of the model. The term “video-centered hypermedia” was suggested by Alberto Colorni of Politecnico di Milano.

Notes

1. In the sequel we’ll refer implicitly to version 2.0.
2. The difference between the last two events will be described in Definition 8.
3. The modelling of the whole presentation requires the notion of *link* that will be introduced later.
4. In the figures we use a different arrow style for the relation *terminates*($\Downarrow$) to improve visibility.
5. The reader could note that this behavior is consistent with the one defined in the Amsterdam Hypermedia Model [12].
6. The implementation mixes several virtual audio channels to the same audio device.

References

1. J.F. Allen, “Maintaining knowledge about temporal intervals,” Comm. ACM, Vol. 26, No. 11, pp. 832–843, 1983.
2. E. Bertino and E. Ferrari, “Temporal synchronization models for multimedia data,” IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 4, pp. 612–631, 1998.
3. S. Boll and W. Klas, “ZYX, a multimedia document model for reuse and adaption of multimedia content,” IEEE Transaction on Knowledge and Data Engineering, DS–8 Special Issue, Vol. 13, No. 3, pp. 361–382, 2001.
4. L.A. Carr, D.W. Barron, H.C. Davis, and W. Hall. “Why use HyTime?” Electronic Publishing—Origination, Dissemination, and Design, Vol. 7, No. 3, pp. 163–178, 1994.
5. A. Celentano and O. Gaggi, “Querying and browsing multimedia presentations,” in Second International Workshop on Multimedia Databases and Image Communication, M.Tucci (Ed.), number 2184 in LNCS, Springer Verlag: Amalfi, Italy, Sept. 2001, pp. 105–116.
6. A. Celentano and O. Gaggi, “Template-based generation of multimedia presentations,” International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 4, pp. 419–445, 2002.
7. A. Celentano, O. Gaggi, and M.L. Sapino, “Retrieval in multimedia presentations,” Multimedia Systems, Vol. 10, No. 1, pp. 146–154, 2004.
8. D.C. Fallside (Ed.), “XML Schema Part 0: Primer,” http://www.w3.org/TR/xmlschema-0/, May 2001.
9. O. Gaggi and A. Celentano, “A visual authoring environment for multimedia presentations on the world wide web,” in IEEE International Symposium on Multimedia Software Engineering (MSE2002), Newport Beach, California, Dec. 2002, pp. 206–213.
10. L. Hardman, “Using the Amsterdam hypermedia model for abstracting presentation behavior,” in Electronic Proceedings of the ACM Workshop on Effective Abstractions in Multimedia, San Francisco, CA, Nov. 1994.
11. L. Hardman, “Modelling and Authoring Hypermedia Documents,” PhD thesis, CWI, University of Amsterdam, 1998.
12. L. Hardman, D.C.A. Bulterman, and G. van Rossum, “The Amsterdam hypermedia model: Adding time, structure and context to hypertext,” Communications of the ACM, Vol. 37, No. 2, pp. 50–62, 1994.
13. L. Hardman, J. van Ossenbruggen, L. Rutledge, K. Sjoerd Mullemder, and D.C.A. Bulterman, “Do you have the time? Composition and linking in time-based hypermedia,” in ACM Conference on Hypertext and Hypermedia’99, Darmstadt, Germany, Feb. 1999, pp. 189–196.
14. M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismail, and L. Tardif, “Madeus, an authoring environment for interactive multimedia documents,” in ACM Multimedia 1998, Bristol, UK, Sep. 1998, pp. 267–272.
15. P. King, H. Cameron, H. Bowman, and S. Thompson, “Synchronization in multimedia documents,” in Electronic Publishing, Artistic Imaging, and Digital Typography, Lecture Notes in Computer Science, J. Andre (Ed.), Springer-Verlag, May 1998, Vol. 1375, pp. 355–369.
16. MPEG ISO/IEC Joint Technical Committee. MPEG-4 Overview, ISO/IEC JTC1/SC29/WG11 N4668, March 2002.
17. S.R. Newcomb, N.A. Kipp, and V.T. Newcomb, “The HyTime: Hypermedia/ Time-Based document structuring language,” Communications of the ACM, Vol. 34, No. 11, pp. 67–83, 1991.
18. Palazzo Grassi, The Maya Exhibition, Venice, Italy, 1999. http://www.palazzograssi.it/eng/mostre/mostre_maya.html
19. J.A. Schnepf, J.A. Konstan, and D. Hung-Chang Du, “Doing FLIPS: FLexible interactive presentation synchronization,” IEEE Journal on Selected Areas of Communications, Vol. 14, No. 1, pp. 114–125, 1996.
20. J. Schnepf, Y. Lee, D. Du, L. Lai, and L. Kang, “Building a framework for flexible interactive presentations,” in Pacific Workshop on Distributed Multimedia Systems (Pacific DMS 96), Hong Kong, June 1996, pp. 296–305,
21. Synchronized Multimedia Working Group of W3C, Synchronized Multimedia Integration Language (SMIL) 2.0 Specification, http://www.w3.org/TR/smil20, August 2001.
22. M. Vazirgiannis, Y. Theodoridis, and T. Selling, “Spatio-temporal composition and indexing for large multimedia applications,” Multimedia Systems, Vol. 6, No. 4, pp. 284–298, 1998.
23. Jin Yu, “A Simple, intuitive hypermedia synchronization model and its realization in the Browser/Java environment,” in Asia Pacific Web Conference, Hong Kong, Sept. 1998, pp. 209–218.

Ombretta Gaggi received her PhD in Computer Science from the Consortium of Universities of Bologna, Padova and Venice in 2003, and is currently a Post-Doc fellow at the Department of Computer Science of Ca' Foscari University in Venice. Her research interests include multimedia document and application design, mobile context-aware information systems, and distributed multimedia modelling.She is serving as reviewer and program committee member in international conferences.

Augusto Celentano is full professor of Information Systems at Ca' Foscari University in Venice. He received a Master Degree in Electronic Engineering from the Politecnico di Milano. He has been Head of the Department of Computer Science of Ca' Foscari University, and Deputy Rector for information technology. Augusto Celentano has been member of scientific committees in research and educational centers of the Politecnico di Milano, and has been consultant and scientific coordinator of research projects of the European Union. His current research interests are in mobile, pervasive and context-aware information systems, in multimedia systems and in Human Computer Interaction. He is co-author of more than 80 scientific papers published in international journals and conference proceedings, and is serving as reviewer and program committee member in international conferences.