Chapter 3
Theoretical Foundations: Formalized Temporal Models for Hyperlinked Multimedia Documents

Britta Meixner

Abstract Consistent linking and accurate synchronization of multimedia elements in hypervideos or multimedia documents are essential to provide a good quality of experience to viewers. Temporal models are needed to define relationships and constraints between multimedia elements and create an appealing presentation. However, no commonly used description language for temporal models exists. This makes existing temporal models harder to understand, compare, and transform from one to another temporal model. Using a formal description is more accurate than commonly used textual descriptions or figures of temporal models. This abstract representation makes it is easier to precisely define algorithms and constraints for delivery and buffering, as well as behavior of user and/or multimedia document. The use of a common formalism for all temporal models makes it possible to define synchronization constraints and media management. The same variables and terminology can then be used for describing algorithms that are applied to the documents, for example, to implement pre-fetching or download and cache management in order to increase the quality of experience for users. In this chapter, we give an overview of different existing temporal models for linked and temporally synchronized multimedia documents, like point-based, event-based, or interval-based temporal models. We analyze their common features and formally define their elementary components. We then give formal definitions for each temporal model covering essential features. These can then be used to computationally solve existing problems. We show this by defining basic functions that can be used in algorithms. We also show how user interaction and resulting video behavior can be precisely defined.

Keywords Multimedia document · Hypervideo · Temporal model
Formal definition

B. Meixner (✉)
CWI, Science Park 123, 1098 XG Amsterdam, Netherlands
e-mail: britta.meixner@cwi.nl

M. Montagud et al. (eds.), *MediaSync*,
https://doi.org/10.1007/978-3-319-65840-7_3

3.1 Introduction

Recent Web technologies like HTML5 allow the creation of appealing and interactive presentations consisting of various linked multimedia elements like images, videos, audio, and text. Users can interact and explore contents, find additional information about topics, and maybe even add their own contents. With growing Internet bandwidths and fast end user devices like smartphones and tablets, even synchronized and interactive multiscreen presentations are possible. Users can, for example, watch sports broadcasts on TV and, at the same time, follow their favorite athlete on a smartphone. They may get statistics and current standings in an object-based way that they can enable and disable on one of the screens. However, while it is technically possible to create hyperlinked single-device multimedia presentations or synchronized multidevice services, they are not popular.

The major challenge in this area is to ensure a good quality of experience by avoiding stalling event and synchronization issues between devices during playback. Work trying to optimize the user quality of experience by pre-fetching or adapting multimedia elements in hyperlinked or synchronized environments is needed in the future. This chapter helps researchers trying to create and optimize algorithms for temporal synchronization in multimedia presentations by providing models which can be used for computation. This approach was, for example, used for video centered hypervideos during the standardization process of HTML5. While a large part of the standard was already implemented in the commonly used browsers, the standard definitions and browser implementation constantly changed making it impossible to implement and test the developed algorithms. Using the formal temporal model and definitions enabled us to describe our algorithms in a platform-independent way and implement them in a test framework as described in [38]. After the HTML5 standard was finished and implemented in browsers, the findings were transformed to match the constraints given by the standards and could be easily reimplemented in HTML5 and JavaScript as shown in [35].

Giving formal definitions for the existing temporal models, algorithms can be defined using them. In order to explain basic parts and concepts of a multimedia document, we give and explain an example of a hypervideo. Figure 3.1 shows an exemplary structure of a hypervideo representing a tour through the ground floor of a house. Navigation between the rooms is realized via user-selection events. The lower left part of the image depicts the layout of the ground floor. The video scenes shown in the scene graph in the rest of the figure are filmed paths through rooms of the house, from one door to another. Viewers are asked where they want to go at certain points and are able to choose their own unique way through the house. The structure of the scenes defines a scene graph. The 16 scenes of the video are represented as labeled rectangles. The diamond symbolizes a fork in the flow where the viewer can choose a scene. Possible targets of a scene are other scenes, a fork, or the end of the video. The decision, which path is followed, depends on the click of the appropriate button. Here, the scene graph has a source (yellow triangle) and a sink (red circle). It is directed, weighted, and possibly cyclic.

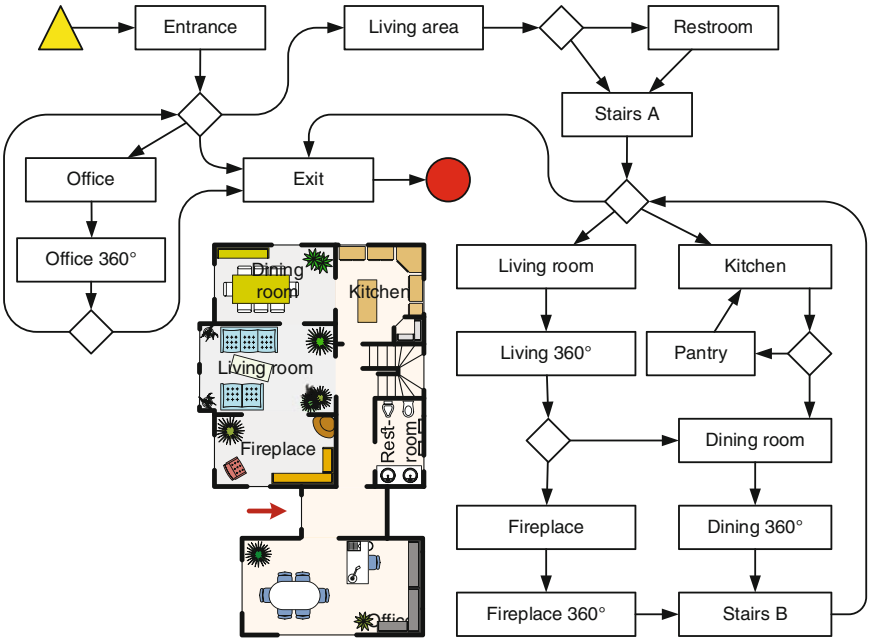

Fig. 3.1 Scene graph of a tour through the ground floor of a house with six rooms

Individual scenes are annotated with detailed images of furniture or flooring adding parallel multimedia elements to the main video. Doing this, a plain video can be enriched with additional information that may not be relevant to all viewers. The viewer may interact and get more information if desired. Text annotations (for example, showing prices and contact information), images (for example, showing alternative floor tiles), and audio files (for example, playing background music) describe room specifics or items shown in a scene that reach beyond the information provided by the video. Images provide detailed views of objects in the video. Figure 3.2 shows the time spans for displaying annotations of the entrance scene in a detailed view. This scene consists of a video with 710 frames (solid green boxes). During the playback of the video, four annotations are shown and hidden as exemplified by the diamond patterned boxes.

However, mixing different types of media and displaying them in parallel as done in the previous example may lead to QoE issues when, for example, videos are not properly synchronized or long buffering times after user interactions appear. Models help to define synchronization constraints and to create and specify algorithms for pre-fetching, download and buffer management. These algorithms can be pre-evaluated in simulations in test beds. A transformation of successfully pre-evaluated algorithms into real-world implementations helps solving synchronization and buffering issues, especially when underlying technologies and standards change.

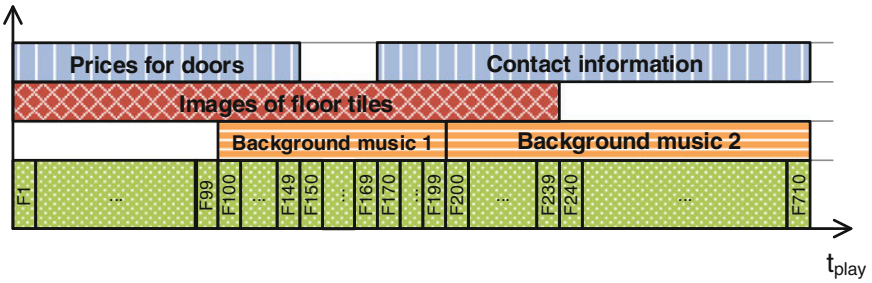

Fig. 3.2 Schedule for six annotations of one scene in detail

In this chapter, we first introduce fundamentals and terms in Sect. 3.2. We then provide universal definitions and descriptions for different temporal models (point-based, event-based, and interval-based) as described in literature in Sect. 3.3. We then give definitions of basic elements of multimedia documents in Sect. 3.4. After that, we provide formals descriptions for each temporal model in Sect. 3.5. Exemplary applications of definitions and models can be found in Sect. 3.6. This chapter ends with a conclusion in Sect. 3.7.

3.2 Fundamentals and Terms

Depending on used media, temporal model, and allowed interaction, we are talking about different forms of multimedia documents, depending if they are hyperlinked and/or synchronized. These can be described by the following terms (definitions beyond the below mentioned can be found in [39]):

- **multimedia element**: A multimedia element is an image, a video, an audio, a text, or any other type of audiovisual medium. It is the atomic object of any multimedia document.
- **annotation**: An annotation is additional information displayed with a main medium. It consists of an anchor attaching it to the main medium and a body. The body of an annotation is a multimedia element that can be shown in a player [39].
- **static multimedia element**: Static multimedia elements are time independent and always show the same content, like images and/or text.
- **continuous multimedia element**: Continuous multimedia elements are time dependent showing/playing different contents over time, like videos or audios.
- **hyperlinked media**: Hyperlinked media are multimedia elements which are linked with each other by hyperlinks (as known from hypertext). Static media may be clickable or have clickable areas. Continuous media may provide links depending on the media time.

- **media synchronization**: Synchronization of multimedia elements requires mechanisms to prepare the media for display (i.e., pre-fetch, buffering, rendering) and to ensure that timing constraints are met.
- **multimedia document**: A multimedia document is a *self-contained* presentation of linked and synchronized multimedia elements which allows user interaction and navigation. Usually, it is about a certain topic.

Many subcategories of multimedia documents providing more or less interactive features exist. They may focus on one main medium like video (called "video centered") or allow mixing different media files. Regardless of the differences, for each form of multimedia document, a description of some sort is necessary to define possible interactions as well as temporal and spatial relationships between multimedia elements. For example, a video description may specify sequences of scenes and the point in time at which the viewer can interact with the video, as well as the points in time where annotations are displayed or hidden. In addition, the description defines relationships and outlines their structure. Two multimedia elements may have a sequential, a conditional, or a parallel relationship, which needs to be defined in order to create a multimedia document. Describing the relationships and links between multimedia elements creates a structure which may be a linear, a tree, or a graph.

Depending on the desired output and use case of the multimedia document, different temporal models may be used. They are different in the way in which temporal relationships and control are specified. They may be using points, events, or intervals which will be explained hereafter:

- **point**: A (time) point is a precise moment in time [41]. It is synchronized with a clock.
- **event**: An event is something that happens or takes place [41]. It may be triggered by a clock or by a user interaction.
- **interval**: A (time) interval is the time between start and end of a time span.

Distinguishing among hypermedia, passive multimedia (presentations), and active multimedia (presentations) is useful for the definition of temporal models, because it limits their scope. The terms can be defined as follows:

- **hypermedia**: Hypermedia is an extension to hypertext providing multimedia facilities, such as those handling sound and video [41]. Keeping the hyperlink structure from hypertext, multimedia elements of different types are added.
- **multimedia**: Multimedia uses a variety of artistic or communicative media that are presented in one presentation [41].
  - **passive multimedia**: Passive multimedia presentations are started and then watched with little to no interaction. Available forms of interaction are starting, pausing, and stopping the presentation.
  - **active multimedia**: Active multimedia presentations allow more interaction compared to passive multimedia presentations. They may have hyperlinks or other interactive control elements.

According to Hirzalla et al., hypermedia "implies store-and-forward techniques where user actions, typically mouse-selections on hotspots, cause the system to retrieve a new 'page' of data which could be an image, text, video etc. There are usually no temporal relationships between media. Passive multimedia implies a fully synchronized document that 'plays itself back' in time, synchronizing all multimedia elements together. Active multimedia implies that there are hypermedia-type choices presented to users during the playback of a multimedia document which allow the user's interaction to 'drive' the playback" [22].

3.3 Temporal Models

Hereafter, we describe and analyze related work for point-based, event-based, interval-based, and other less common temporal models and show common features of the temporal models that will be formalized in Sect. 3.5. As an example, we use the scene from Fig. 3.2 in Sect. 3.1. An overview of all variables is given in Table 3.4 in the Appendix.

3.3.1 Point-Based Temporal Models

Using point-based temporal models, each event is triggered by its point in time. The points in time are ordered on a timeline [15]. The points in time form a total order [15]. "For any two points in time [,] one of the relationships before (<), after (>), or equals (=) holds" [7]. "An example of the point-based approach is [a] timeline, in which multimedia elements are placed on several time axes called tracks, one per each media type. All events such as the beginning or the end of a segment are totally ordered on the time line" [15]. Figure 3.3 shows an exemplary presentation with a point-based temporal model. The presentation consists out of six multimedia elements. Three of them start at t_0, one starts at t_7, one starts at t_{13}, and the last one starts at t_{16}. One element is displayed over the whole presentation while the others start later or end earlier. The presentation ends at t_{29}.

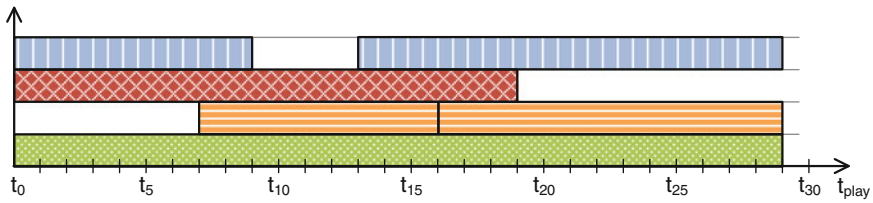

Fig. 3.3 Exemplary timeline of a presentation with six multimedia elements and time points for starting/stopping their display

Other approaches that incorporate this temporal model are point nets [9]. This temporal model requires media segments with known durations, and it is not possible to use it for unknown durations [15]. Wahl et al. call the previously described relationships before (<), after (>), or equals (=) "basic point relations (basic PRs)" [49]. They also differentiate between relations in the past and relations between future events which might be indefinite and it might not be known which one out of two possible relations will become true. "Typically, indefinite relations are represented as disjunctions of basic PRs. Since there are three basic PRs, $2^3 = 8$ disjunctions exist each representing an indefinite relation. Any of the eight indefinite relations has an associated symbolic notation. The eight indefinite relations are as follows: $\oslash, \leq, <, =, >, \geq, \neq, ?$, where ? is the full set of basic PRs $<, =, >$, $\oslash$ is the empty set $\{\}$" [49].

Blakowski and Steinmetz call the timeline "axis". They differentiate between "synchronization based on a global timer" and "synchronization based on virtual axes". They describe the use of different clocks depending on the underlying content. They point out that the temporal model using the global timer is easy to understand, it supports hierarchies, it is easy to maintain, it provides a good abstraction for multimedia elements, and the integration of time-dependent objects is easy. Disadvantages are that objects need a previously defined duration and additional effort is necessary to implement QoS. The temporal model using virtual axes allows in addition to add specifications according to a certain problem space. Time-independent multimedia elements can be integrated, and interactive objects are possible. However, the additional axes may lead to complex specifications and the mapping of the axes during runtime may be complex and time-consuming [5].

Different works use the point-based temporal model. "Prior to 1993, few multimedia systems had reached the 'document formatter' stage. They typically required authors to create temporal layouts manually by positioning media at absolute points on a document timeline [14, 34, 40, 43], a tedious and error-prone process" [10]. Further systems and standards using this temporal model are, for example: the Firefly multimedia document system [9] and the works by [14, 20].

3.3.2 Event-Based Temporal Models

In event-based temporal models, synchronization events trigger presentation actions. Synchronization events aim at a target (like a multimedia element or a point in time) and contain an associated action (like show/hide a medium) that must be triggered at a given time [44]. Typical actions are starting, stopping, and preparing a presentation. Events can be external (e.g., generated by a timer) or internal (e.g., generated by time-dependent multimedia element) [5]. Figure 3.4 shows an exemplary presentation with an event-based temporal model. The presentation consists out of six multimedia elements. Three of them start at the beginning, triggered by e_0, e_1, and e_2, and another one is started at event e_3. At e_4, the first element is stopped/hidden, and then another one is started at e_5. Another element is hidden at e_6, immediately after-

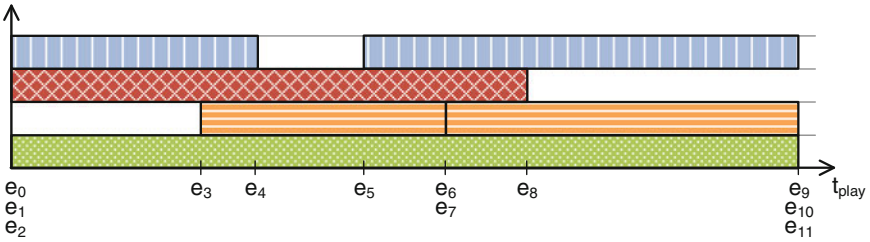

Fig. 3.4 Exemplary timeline of a presentation with six multimedia elements and starting and stopping events

ward, an element is started at e_7. The events e_8, e_9, e_{10}, and e_{11} are stopping/hiding multimedia elements which finally finish the presentation.

According to Blakowski and Steinmetz, event-based temporal models are easily extensible with new events, they are easy to integrate interactive objects, and they are flexible, because any event can be specified [5]. Disadvantages are that they are difficult to handle, have a complex specification, are hard to maintain, and time-dependent objects can only be integrated by using additional timers [5]. Boll et al. describe event-based temporal models as follows: "In an event-based model of time, events determine the temporal course of the presentation. An event is connected to actions and when an event occurs, e.g., a video reaches a certain point in time, the corresponding actions, typically start and stop of the presentation of other media elements, is carried out" [7].

Works using the event-based temporal model are, for example, HyTime [26], HyperODA [3], MHEG-5 [25], or the SIVA Suite [36–38]. "Events are defined in HyTime as presentations of media objects along with the playout specifications and finite coordinate system (FCS) coordinates. HyperODA events happen instantaneously and mainly correspond to start and end of media objects or timers. All these approaches suffer from poor semantics conveyed by the events. Moreover, they don't provide any scheme for composition and consumption architectures" [48]. The temporal model of the SIVA Suite has several layers where events can occur. The hypervideo is video-based and divided into smaller units called scenes. A scene has one single main video and may have several annotations which can either be triggered by time or by a user interaction. Interaction with the annotations is also possible. In a scene, showing and hiding of media or interactive elements that can show annotations is triggered when the main video reaches a certain point in time. Scenes are linked with each other, and the follow-up scene is triggered by a user-selection event.

3.3.3 Interval-Based Temporal Models

Interval-based temporal models are based on intervals which are defined by two points on a timeline. "Interval-based models consider elementary media entities as

time intervals ordered according to some relations" [48]. Depending on the complexity and possible interactions with the temporal model, one can differentiate between basic and enhanced interval-based temporal models. However, enhanced interval-based temporal models are always extensions of basic interval-based temporal models.

3.3.3.1 Basic Interval-Based Temporal Models

Following the definitions of Allen ("Assuming a model consisting of a fully ordered set of points of time, an interval is an ordered pair of points with the first point less than the second" [2]) and Wahl and Rothermehl ("As any interval can be characterized by its beginning and end, any basic [interval relations] can be represented by a conjunction of [point relations] on its margins" [49]), 13 basic relations between two intervals can be defined [2]:

- *X equals Y* (meaning start and end point of both intervals are the same, parallel intervals);
- *X before Y* and *Y before X* (meaning one interval has no overlap with the other);
- *X meets Y* and *Y meets X* (a sequence of intervals);
- *X overlaps Y* and *Y overlaps X* (overlap of the intervals but also parts where only one interval is played);
- *X during Y* and *Y during X* (first interval shorter than second, all of first is parallel with second);
- *X starts Y* and *Y starts X* (both intervals have the same start point and are parallel, but the first is shorter than the second); and
- *X finishes Y* and *Y finishes X* (both intervals have the same end point and are parallel, but the first is shorter than the second).

Figure 3.5 shows an exemplary presentation with an interval-based temporal model using Allen's temporal relationships. The presentation consists out of six multimedia elements. Three of them start at the beginning, (*M*1 *starts M*3, *M*2 *starts M*3), three end at the same time (*M*5 *finishes M*3, *M*6 *finishes M*3), and two form a sequence of intervals (*M*4 *meets M*6).

Little and Ghafoor extend this temporal model to n-ary temporal relations [30]. Wahl and Rothermehl propose an enhanced interval-based temporal model with 29

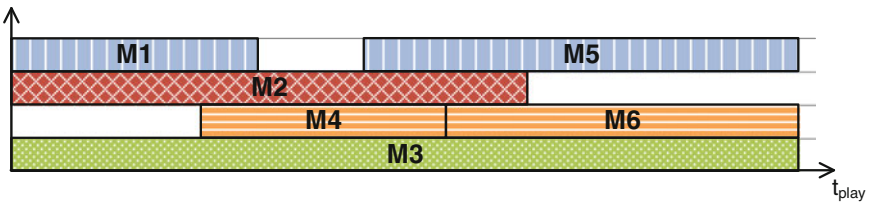

Fig. 3.5 Exemplary timeline of a presentation with six multimedia elements

interval relations in [49], which extends the 13 basic relations of [2]. Advantages of the interval-based temporal model are that logical objects can be kept, there is a good abstraction for media content, it is easy to integrate time-independent and/or interactive objects, and the specification of indeterministic temporal relations is supported. Disadvantages are a complex specification, the necessity of additional specifications for QoS that it is only possible to specify relations between whole multimedia elements, not subparts of multimedia elements, and that resolving indeterminism at runtime may lead to inconsistencies [5].

Interaction with interval-based temporal models is discussed by Wahl et al. [50] and Little and Ghafoor [30]. Wahl et al. define two different temporal interaction forms: context interaction (start, stop, and selection (jump to new document)) and speed interaction (for example, pause/continue, faster, slower, forward, backward, reverse, and set speed). These interactions always affect the whole presentation, not single elements [50]. Little and Ghafoor state that "uncertainty created by random user interaction is an additional complexity in managing time in multimedia information systems" [30].

Works using the interval-based temporal model are in addition to the already mentioned the following: The work of Fujikawa et al. describes "Harmony" which uses a combination of events and intervals [19]. Euzenat et al. propose a semantic framework for multimedia adaptation for heterogeneous devices resulting in various constraints like different display sizes and available bandwidths. Thereby, a temporal "model of a multimedia document is a potential execution of this document and a context defines a particular class of models" [17]. Little and Ghafoor use a Petri Net definition where intervals are represented by places and relations by transitions to deal with the interval-based temporal model [32]. A later work of Little and Ghafoor proposes a solution to store a temporal model in a database [30]. Wahl and Rothermehl analyze path expressions [11, 23] which include three operators to represent temporal relations: sequence, parallel-first, and parallel-last deal with interval-based temporal models. They also discuss MHEG [25, 29] which uses two temporal operators sequential and parallel similarly to the path expression model [49].

3.3.3.2 Enhanced Interval-Based Temporal Models

There are several issues with Allen's relations [2], according to Duda and Keramane [15]. The relations were designed for intervals with fixed durations. Changes in durations may transform one relation into another. "Another problem with the Allen relations is their descriptive character, they allow expression of an existing, a posteriori arrangement of intervals, but they do not express any causal or functional relation between intervals" [15]. This makes Allen's relations useful for characterizing existing, instantiated presentations where all start and termination points of media segments are known. The third problem is, that the relations may lead to inconsistent specifications that may occur in a multimedia presentation. The detection of the third problem requires algorithms of complexity [$O(N^2)$], where N is the number of intervals [2]. To deal with these issues, enhanced interval-based

temporal models use the 13 basic relations of [2], or similar definitions but extend them with, for example, mechanisms to deal with unknown durations of intervals, semantics on the intervals, or substitutions of elements in certain contexts.

Benbernou et al. use techniques called "Augmentation and Substitution" [4]. They find alternatives for multimedia elements that are "semantically closed" using definitions of semantic constraints between media elements. They also substitute "unwanted media" using alternatives which take the spatiotemporal coherence of the presentation into account [4]. Boll et al. introduce intervals of unknown durations for user interaction in multimedia presentations. "With the Interval Expressions [15] we find a temporal model for multimedia presentations on the level of intervals with a set of temporal operators to relate time intervals which possibly have an unknown duration, that also overcomes the problem of temporal inconsistencies by construction. The Interval Expressions form the basis of the underlying temporal model of the ZyX data model" [8]. SMIL also uses an enhanced interval-based temporal model.

3.3.4 Other

Other temporal models are mentioned in the literature, but are less important and significant compared to the already mentioned temporal models. Furthermore, it may be possible to categorize specific implementations in one of the other categories. The two major subcategories are script-based and tree-based temporal models. According to Blakowski and Steinmetz, scripts have the following advantages: they have a good support for hierarchies, logical objects can be kept, it is easy to integrate time-independent and interactive objects, these are easily extensible with new constructs and flexible due to their programmability. However, they are not easy to handle, can have complex specifications, additional timers are necessary as well as constructs to implement QoS. Fiume et al. use temporal scripting languages [18].

Kim et al. use temporal relation trees [27]. Hirzalla et al. propose a timeline-tree temporal model which introduces choice elements into the timelines known from point-based temporal models [22]. Courtiat and De Oliveira use presentation and constraint objects in a hierarchical composition which is then translated into a complete RT-LOTOS formal specification (extension of [16]) [13].

According to Blakowski and Steinmetz, control flow-based specifications may use basic hierarchical descriptions [1, 45] (serial/parallel), reference points [6, 47] or timed Petri Nets [31, 33]. For further descriptions, see [5].

3.3.5 Summary

In this section, we showed differences for existing temporal models. Point-based temporal models are the easiest to define, but are based on one timeline which requires fixed start and end points for multimedia elements. They allow only basic

VCR actions for the overall presentation. Event-based temporal models are more advanced, allowing different types of events and interactions. They are more flexible than point-based temporal models. However, no commonly valid definition of events or the temporal model itself could be found in existing literature. The interval-based temporal models are the most discussed and researched in related work. All of the works are based on the basic interval relationships defined by Allen in 1983 [2]. However, due to their possible nondeterministic presentation, several methods are proposed to synchronize the media for playback. Adding interactivity to the temporal models makes them more complicated, especially with regard to synchronization. If interactivity is possible on single multimedia elements, it gets even more complicated. Table 3.1 shows an overview of the different temporal models with regard to possible relationships between objects/media and events, interactivity, and timing, which are the most distinctive differences between the temporal models. Depending on the given task, one or the other temporal model may have its advantages or disadvantages. Table 3.1 may help to pick the right temporal model given relationships between media and events, timing constraints, or requirements regarding interactivity.

Table 3.1 Comparison of temporal models

Temporal model	Relationships between objects/media	Relationships between events	Interactivity	Timing
Point-based	Sequential, parallel	Before (<), after (>), or equals (=)	Start/stop/ pause whole presentation	Display/hiding and start/stop of media are triggered by points in time on a timeline, total order for points on timeline
Event-based	Sequential, parallel, conditional, linked	Events are not clearly defined throughout literature	Depending on model	Events are triggered by other events
Basic interval-based	13 relationships defined by Allen [2] or 29 interval relations in [49]		VCR actions for whole presentation	Based on relations, not necessarily deterministic
Enhanced interval-based	13 relationships defined by Allen [2] or 29 interval relations in [49]	Before (<), after (>), or equals (=) for start and end points of intervals	Interactions on multimedia elements	Based on relations, not necessarily deterministic

3.4 Definition of Basic Multimedia Elements

As shown in the previous sections, existing temporal models are capable of describing relationships between elements, timing, and interactivity. However, no commonly used mechanism for describing them is available. The authors of each paper use their own formalisms making it hard to compare the temporal models or transform one of the temporal models into another. Hereafter, we give formal definitions of all basic elements that may be part of a multimedia document, namely, static (like text, images) and continuous (like audio, video) multimedia elements. Static and continuous media files are fundamental elements of a multimedia presentation and appear in each of the temporal models in some way.

A static annotation is a static multimedia element which contains a multimedia element α (see Definition 1) and has a priority Λ (see Definition 2). The multimedia element may be any type of additional content like text, image, video, or audio files and is defined as a sequence of bits. These usually have to be transmitted over a network. For simplification purposes, the content of a continuous multimedia element may be handled as a single block whether it is a continuous or a static medium. More precisely, a static multimedia element a is a pair of the multimedia element α and a priority Λ (see Definition 3). We also define the set of static multimedia elements $\mathcal{A}_D$ (see Definition 4) of a multimedia document $\mathcal{D}$.

Definition 1 (*Content of a Static Multimedia Element* α) The content of a static multimedia element α is an n-tuple of bits representing a multimedia element; $\alpha := (\chi_1, \ldots, \chi_n) \in \{0,1\}^n, n \in \mathbb{N}^+$.

Definition 2 (*Priority of a Static Multimedia Element* Λ) The priority of a static multimedia element is Λ, $\Lambda \in \mathbb{N}^+$. The higher Λ is, the lower is the priority of the static multimedia element.

Definition 3 (*Static Multimedia Element* a) A static multimedia element $a := (\alpha, \Lambda)$, $\Lambda \in \mathbb{N}^+$ is a pair of the content of the static multimedia element α and the priority of the static multimedia element Λ.

Definition 4 (*Set of Static Multimedia Elements* $\mathcal{A}_D$) $\mathcal{A}_D$ is a finite set of static multimedia elements of the multimedia document $\mathcal{D}$;
$\mathcal{A}_D := \{a_i | a_i \text{ is a static multimedia element.}\}$

A continuous multimedia element consists of frames (video) or samples (audio). These consist of a sequence of bits as defined in Definition 5.

Definition 5 (*Frame/Sample* f) A frame/sample f is an n-tuple of bits representing an image or audio sample; $f := (\chi_1, \ldots, \chi_n) \in \{0,1\}^n, n \in \mathbb{N}^+$.

We define the set of frames/samples $\mathcal{F}_D$ (see Definition 6) of a multimedia document as well as an ordered n-tuple of frames/samples F. Similar to the set of static multimedia elements $\mathcal{A}_D$, we also define a set of continuous multimedia elements $\mathcal{C}_D$.

Definition 6 (*Set of Frames* $\mathcal{F}_D$) $\mathcal{F}_D$ is a finite set of frames/samples of the multimedia document $\mathcal{D}$.

Definition 7 (*Continuous Multimedia Element F*) A continuous multimedia element $F := (f_\sigma, f_1, \ldots, f_n, f_\epsilon)$, $n \in \mathbb{N}^+, f_i \in \mathcal{F}_D, 1 \leq i \leq n$ is an n-tuple of frames/samples f with start frame/sample f_σ and end frame/sample f_ϵ.

Definition 8 (*Set of Continuous Multimedia Elements* C_D) C_D is a finite set of continuous multimedia elements of the multimedia document $\mathcal{D}$;
$C_D := \{F_i | F_i$ is a continuous multimedia element.$\}$

3.5 Formalized Temporal Models

Now we use the definitions of basic multimedia elements from Sect. 3.4 to describe more complex structures in point-based, event-based, and interval-based temporal models.

3.5.1 Point-Based Temporal Model

With the definitions in Sect. 3.4, all multimedia elements are already defined. Additional definitions are necessary for the timeline and events which show or hide a multimedia element. Interaction with single multimedia elements is usually not possible in point-based temporal models and does not need to be defined accordingly.

3.5.1.1 Additional Definitions

Points in time t (see Definition 9) as well as time intervals T (see Definition 10) need to be defined to control when events are applied to multimedia elements. The timeline is given as an abstract external clock.

Definition 9 (*Point in Time t*) The point in time of a presentation is t, $t \in \mathbb{N}$.

Definition 10 (*Time Interval T*) A time interval T is a n-tuple of discrete points in time t_i; $T := (t_0, \ldots, t_n)$, $n \in \mathbb{N}^+, t_i \in \mathbb{N}, 1 \leq i \leq n$.

In order to define what happens to a multimedia element at a certain point in time, an executable action *ea* (see Definition 11) for a multimedia element e is defined. Possible executable actions vary between static and continuous media. Static media can be shown and hidden. Continuous media can be started, paused, and stopped. Sets for executable actions for static multimedia elements $\mathcal{EA}_a$, dynamic multimedia elements $\mathcal{EA}_F$ and all executable actions $\mathcal{EA}$ are defined in Definitions 12, 13, and 14.

Definition 11 (*Executable Action ea*) An executable action *ea* is an action that is applied to a multimedia element *e*, $e \in \mathcal{A}_D \cup C_D$.

Definition 12 (*Set of Executable Actions for Static Multimedia Elements* $\mathcal{EA}_a$) For static multimedia elements *a*, the finite set of actions is $\mathcal{EA}_a := \{show_a, hide_a\}$.

Definition 13 (*Set of Executable Actions for Dynamic Multimedia Elements* $\mathcal{EA}_F$) For continuous multimedia elements, the finite set of actions is $\mathcal{EA}_F := \{start_F, pause_F, stop_F\}$.

Definition 14 (*Set of Executable Actions* $\mathcal{EA}$) The finite set of executable actions $\mathcal{EA}$ of a presentation is defined as $\mathcal{EA} := \mathcal{EA}_a \cup \mathcal{EA}_F$.

After defining executable actions, these need to be linked to multimedia elements in order to indicate what should happen with the multimedia element. Therefore, we define a pair *ee* called element event in Definition 15. The set of element events $\mathcal{EE}$ is defined in Definition 16.

Definition 15 (*Element Event ee*) An element event *ee* is a pair containing an element *e* and an executable action *ea*; $ee_j := (e_j, ea_n),\ e_j \in \mathcal{A}_D \cup \mathcal{F}_D,\ ea_n \in \mathcal{EA}$.

Definition 16 (*Set of Element Events* $\mathcal{EE}$) $\mathcal{EE}$ is a finite set of element events *ee*.

At each point in time, one or more events may occur. For that reason, we link a set of element events to a point in time. The resulting pair *ep* is called event point (see Definition 17), the set of event points $\mathcal{EP}$ is defined in Definition 18.

Definition 17 (*Event Point ep*) An event point *ep* is a pair containing a point in time t_i, and a set of element events EE_i; $ep_i := (t_i, EE_i),\ t_i \in T,\ EE_i \subset \mathcal{EE} \cup \emptyset$.

Definition 18 (*Set of Event Points* $\mathcal{EP}$) $\mathcal{EP}$ is a finite set of event points *ep*.

The whole presentation can be described as an n-tuple of event points *ep* and is called a timeline *tl* (see Definition 19).

Definition 19 (*Timeline tl*) A timeline *tl* is an n-tuple of event points; $tl := (ep_0, \ldots, ep_n),\ n \in \mathbb{N}^+, ep_i \in \mathcal{EP},\ 1 \leq i \leq n$.

3.5.1.2 Example

Definition 19 is now illustrated with a small example as shown in Fig. 3.6 and specified in Eq. 3.1. This timeline has 30 time points. The presentation consists of three continuous multimedia elements (two videos (F_1 and F_2) and one audio (F_3)) and four static multimedia elements (one text (a_1) and three images (a_2, a_3, and a_4)). The text is shown over the whole time. The images are shown one after the other with a break in-between. First, video F_1 is shown, then video F_2, which is muted and audio F_3 is used to replace its sound.

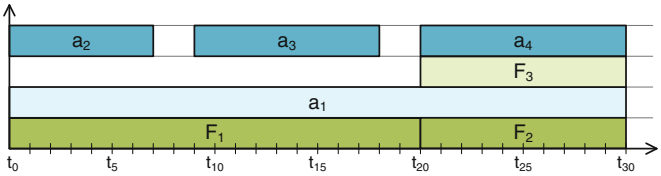

Fig. 3.6 Exemplary timeline of a presentation with continuous and static multimedia elements

The **set of continuous media** from Eq. 3.1 is set to $C_D = \{F_1, F_2, F_3\}$ and contains two videos, F_1 and F_2, and one audio F_3. The **set of static media** from Eq. 3.1 contains four elements. It is defined as $\mathcal{A}_D = \{a_1, a_2, a_3, a_4\}$ and contains one text a_1 and three images a_2, a_3, and a_4.

$$
\begin{aligned}
tl = &((t_0, \{(F_1, start_F), (a_1, show_a), (a_2, show_a)\}),\\
&(t_1, \emptyset),\\
&\dots,\\
&(t_6, \emptyset),\\
&(t_7, \{(a_2, hide_a)\}),\\
&(t_8, \emptyset),\\
&(t_9, \{(a_3, show_a)\}),\\
&(t_{10}, \emptyset),\\
&\dots,\\
&(t_{17}, \emptyset),\\
&(t_{18}, \{(a_3, hide_a)\}),\\
&(t_{19}, \emptyset),\\
&(t_{20}, \{(F_1, stop_F), (F_2, start_F), (F_3, start_F), (a_4, show_a)\}),\\
&(t_{21}, \emptyset),\\
&\dots,\\
&(t_{29}, \emptyset),\\
&(t_{30}, \{(F_2, stop_F), (F_3, stop_F), (a_1, hide_a), (a_4, hide_a)\})).
\end{aligned}
\tag{3.1}
$$

3.5.2 *Event-Based Temporal Model*

While no clear definition and common understanding of the event-based temporal model exists, we show one form of hypervideo where this temporal model is used, the so-called Annotated Interactive Nonlinear Video $\mathcal{V}$ [38]. This video centered type of hypervideo always has a video as main medium, which also defines the clock during

playback. All other multimedia elements are called annotations and are treated as single block of data, no matter if the annotation is a static or continuous medium. In addition to video scenes and annotations, control information is defined which is needed during playback to select the successor scene at a fork. With the definitions in Sect. 3.4, we can define videos scenes. These can then be extended and used to formulate the definition of annotated interactive nonlinear video $\mathcal{V}$ as a deterministic finite state machine.

3.5.2.1 Additional Definitions

An n-tuple containing pairs of a frame and a set of annotations is representing a scene p (see Definition 20). The set of annotations attached to the frame indicates that all annotations in the set are displayed with the frame. An annotated interactive nonlinear video furthermore has a start scene p_σ and an end scene p_ϵ with just one frame and an empty set of annotations (see Definitions 21 and 22). All scenes can be combined to a set of scenes $\mathcal{P}_\mathcal{V}$ (see Definition 23).

Definition 20 (*Scene p*) A scene p is an n-tuple of pairs each containing a frame and a set of annotations which are displayed with the frame; $p_x := ((f_{x,1}, A_{x,1}), \ldots, (f_{x,n}, A_{x,n}))$, $x, n \in \mathbb{N}^+, f_{x,i} \in \mathcal{F}_\mathcal{V}, A_{x,i} \subseteq \mathcal{A}_\mathcal{V}, 1 \leq i \leq n$.

Definition 21 (*Start Scene p_σ*) The start scene p_σ is a 1-tuple containing a pair representing a single frame without an annotation; $p_\sigma := ((f_{\sigma,1}, \{\}))$.

Definition 22 (*End Scene p_ϵ*) The end scene p_ϵ is a 1-tuple containing a pair representing a single frame without an annotation; $p_\epsilon := ((f_{\epsilon,1}, \{\}))$.

Definition 23 (*Set of Scenes $\mathcal{P}_\mathcal{V}$*) The set of scenes $\mathcal{P}_\mathcal{V}$ of the annotated interactive nonlinear video is defined as $\mathcal{P}_\mathcal{V} := \{p_\sigma, p_1, \ldots, p_x, p_\epsilon\}, x \in \mathbb{N}^+$.

The whole annotated interactive nonlinear videos can be described with the elements and sets defined so far as a deterministic finite state machine $\mathcal{N}_\mathcal{V}$ (see [24, p. 14 et seq.]). The finite set of states is represented by the set of scenes $\mathcal{P}_\mathcal{V}$. The input symbols Σ are defined as a set of Boolean functions as defined in [21, p. 40]. The transition function is δ, it takes a scene as the current state and a button click as input and returns a scene as next state. The start state is p_σ, the set of end states only contains one element, $\{p_\epsilon\}$.

Definition 24 (*Annotated Interactive Nonlinear Video $\mathcal{V}$*) An annotated interactive nonlinear video $\mathcal{V}$ is defined as a deterministic finite state machine $\mathcal{N}_\mathcal{V} := (\mathcal{P}_\mathcal{V}, \Sigma, \delta, p_\sigma, \{p_\epsilon\})$ with $\Sigma := \{w_{i,j} | w_{i,j}$ is a button triggering the selection of a successor scene, $i \in \{1, \ldots, |\mathcal{P}_\mathcal{V}| - 2, \sigma\}, j \in \{1, \ldots, |\mathcal{P}_\mathcal{V}| - 2, \epsilon\}\}$ and $\delta : \mathcal{P}_\mathcal{V} \times \Sigma \rightarrow \mathcal{P}_\mathcal{V}$.
The following restrictions are applied: $\exists! k : \delta(p_\sigma, w_{\sigma,k}) \rightarrow p_k \wedge \nexists k : \delta(p_k, w_{k,\sigma}) \rightarrow p_\sigma \wedge \exists k : \delta(p_k, w_{k,\epsilon}) \rightarrow p_\epsilon \wedge \nexists k : \delta(p_\epsilon, w_{\epsilon,k}) \rightarrow p_k$.

The deterministic finite state machine $\mathcal{N}_{\mathcal{V}}$ defines possible successors of a scene and which buttons have to be clicked to access a designated successor scene. The transition $(p_m, w_{i,j}) \rightarrow p_n \in \delta$ implies that scene p_n is successor of scene p_m [38]. Applied restrictions define that there is exactly one transition from the start scene to the first scene ($\exists! k : \delta(p_\sigma, w_{\sigma,k}) \rightarrow p_k$), that once the video is started, the start scene cannot be reached ($\nexists k : \delta(p_k, w_{k,\sigma}) \rightarrow p_\sigma$), that there is at least one scene connected to the end scene ($k : \delta(p_k, w_{k,\epsilon}) \rightarrow p_\epsilon$), and once the end scene is reached, the video ends ($\nexists k : \delta(p_\epsilon, w_{\epsilon,k}) \rightarrow p_k$).

Restrictions need to be applied for the start and the end scene.

3.5.2.2 Example

Definition 24 will now be illustrated with a small example as shown in Fig. 3.7. This annotated interactive nonlinear video has six scenes, including start and end scene, and five annotations. The **set of scenes** is defined as $\mathcal{P}_{\mathcal{V}} = \{p_\sigma, p_1, p_2, p_3, p_4, p_\epsilon\}$. The different scenes can be described as follows:

- $p_\sigma = ((f_{\sigma,1}, \{\}))$,
- $p_1 = ((f_{1,1}, \{a_1\}), \ldots, (f_{1,750}, \{a_1\}), (f_{1,751}, \{\}), \ldots, (f_{1,1500}, \{\}))$,
- $p_2 = ((f_{2,1}, \{a_2\}), \ldots, (f_{2,500}, \{a_2\}), (f_{2,501}, \{a_3\}), \ldots, (f_{2,1500}, \{a_3\}))$,
- $p_3 = ((f_{3,1}, \{a_5\}), \ldots, (f_{3,1000}, \{a_5\}))$,

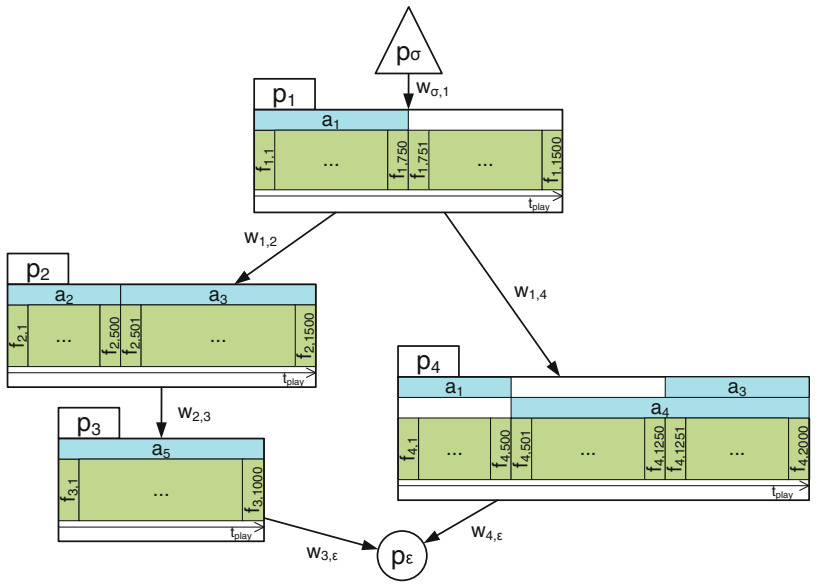

Fig. 3.7 Example of an annotated interactive nonlinear video with six scenes (including start and end scene) and five annotations

- $p_4 = ((f_{4,1}, \{a_1\}), \ldots, (f_{4,500}, \{a_1\}), (f_{4,501}, \{a_4\}), \ldots, (f_{4,1250}, \{a_4\}), (f_{4,1251}, \{a_3, a_4\}), \ldots, (f_{4,2000}, \{a_3, a_4\}))$, and
- $p_\epsilon = ((f_{\epsilon,1}, \{\}))$.

The **set of frames** is set to $\mathcal{F}_D = \{f_{\sigma,1}, f_{1,1}, \ldots, f_{1,1500}, f_{2,1}, \ldots, f_{2,1500}, f_{3,1}, \ldots, f_{3,1000}, f_{4,1}, \ldots, f_{4,2000}, f_{\epsilon,1}\}$ and contains 6002 frames which are divided up into the six scenes. The first and second scene, p_1 and p_2, each consist of 1500 frames, the third scene p_3 consists of 1000 frames and the fourth scene p_4 consists of 2000 frames. The **set of annotations** contains five elements. It is defined as $\mathcal{A}_D = \{a_1, a_2, a_3, a_4, a_5\}$.

The **transition function** δ defines where and under what conditions transitions from one scene to another are allowed. The transition $\delta(p_\sigma, w_{\sigma,1}) \rightarrow p_1$ sets the first scene of the video. Transitions $(p_3, w_{3,\epsilon}) \rightarrow p_\epsilon$ and $(p_4, w_{4,\epsilon}) \rightarrow p_\epsilon$ indicate two different last scenes of the video followed by the end scene. A linear transition is also defined from scene p_2 to p_3 with $\delta(p_2, w_{2,3}) \rightarrow p_3$. In these cases, the follow-up scenes start immediately after the predecessor scenes end. The remaining two transitions, $\delta(p_1, w_{1,2}) \rightarrow p_2$ and $\delta(p_1, w_{1,4}) \rightarrow p_4$, describe a selection panel at the end of scene p_1. The viewer in this example selects button $w_{1,2}$ or button $w_{1,4}$. Only one of the buttons/paths can be selected [38].

3.5.3 Interval-Based Temporal Model

Interval-based temporal models are based on intervals where multimedia elements are shown. Hereafter, we use Allen's 13 basic dual relationships [2] between individual intervals.

3.5.3.1 Additional Definitions

For the definition of relationships between intervals (see Definition 28), intervals have to be defined (see Definition 26). Both are then linked in interval relations as defined in Definition 29. The set of interval relations then defines the whole presentation (see Definition 30). While continuous media are already representing intervals, static media need a display duration to show them in interval-based temporal models. This is defined in Definition 25.

Definition 25 (*Display Duration A*) The display duration of a static multimedia element a is defined as a pair A_i consisting of a multimedia element a_i and its duration t_i; $A_i := (a_i, t_i),\ i \in \mathbb{N}^+, a_i \in \mathcal{A}_D, t_i \in \mathbb{N}^+$.

Definition 26 (*Interval I*) An interval I_e for a multimedia element e is defined as a pair consisting of the start point t_{s_e} and the end point t_{e_e} of the interval;

$$I_e := (t_{s_e}, t_{e_e}), \begin{cases} t_{s_e} - t_{e_e} = t_i & \text{if } e \in \mathcal{A}_D \\ t_{s_e} - t_{e_e} = dim(F) & \text{if } e \in \mathcal{F}_D \end{cases} \tag{3.2}$$

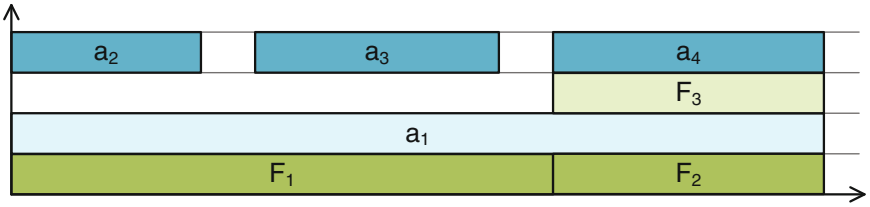

Fig. 3.8 Exemplary relations between intervals with continuous and static multimedia elements

Definition 27 (*Set of Intervals* $\mathcal{I}$) $\mathcal{I}$ is a finite set of intervals I.

Definition 28 (*Set of Relationships* $\mathcal{RS}$) For two intervals, the finite set of interval relationships is $\mathcal{RS} := \{equal, before, meets, overlaps, during, starts, finishes\}$.

Definition 29 (*Interval Relation* R) A relationship between two intervals can be defined as a 4-tuple R consisting of the first interval I_j, the second interval I_k, the relationship between the two intervals rs, and the offset t (in case the relationship requires an offset);

$$R := \begin{cases} (I_j, I_k, rs, \emptyset), & I_j, I_k \in \mathcal{I}, rs \in \mathcal{RS} \\ & \text{if } rs \in \{equal, meets, starts, finishes\} \subset \mathcal{IR} \\ (I_j, I_k, rs, t), & I_j, I_k \in \mathcal{I}, rs \in \mathcal{RS}, t \in \mathbb{N}^+ \\ & \text{if } rs \in \{before, overlaps, during\} \subset \mathcal{IR} \end{cases} \tag{3.3}$$

Definition 30 (*Set of Interval Relations* $\mathcal{IR}$) $\mathcal{IR}$ is a finite set of interval relations R representing a presentation.

3.5.3.2 Example

We now use the example from Sect. 3.5.1.2 in an interval-based temporal model (see Fig. 3.8). As in the point-based example, the presentation has two videos (F_1 and F_2), one audio (F_3), one text (a_1), and three images (a_2, a_3, and a_4). The text is shown all of the time. The images are shown one after the other with a break in-between. First, video F_1 is shown, then video F_2, which is muted and audio F_3 is used to replace its sound.

The elements are defined as follows:

$$\begin{aligned} A_1 &= (a_1, 30) \\ A_2 &= (a_2, 7) \\ A_3 &= (a_3, 9) \\ A_4 &= (a_4, 10) \end{aligned}$$

$$\begin{aligned} F_1 &= (f_{1,1}, \ldots, f_{1,20}) \\ F_2 &= (f_{2,1}, \ldots, f_{2,10}) \\ F_3 &= (f_{3,1}, \ldots, f_{3,10}). \end{aligned} \tag{3.4}$$

Accordingly, the set of intervals is $\{I_{A_1}, I_{A_2}, I_{A_3}, I_{A_4}, I_{F_1}, I_{F_2}, I_{F_3}\}$.

Possible definitions of the set of interval relations for the given example are as follows:

$$\begin{aligned} IR_1 = \{&(I_{A_2}, I_{A_3}, before, 2), (I_{A_3}, I_{A_4}, before, 2), (I_{A_4}, I_{F_3}, equal, \emptyset), \\ &(I_{A_2}, I_{A_1}, starts, \emptyset), (I_{A_2}, I_{F_1}, starts, \emptyset), (I_{F_1}, I_{F_2}, meets, \emptyset)\} \\ IR_2 = \{&(I_{A_2}, I_{A_3}, before, 2), (I_{A_3}, I_{A_4}, before, 2), (I_{A_4}, I_{F_3}, equal, \emptyset), \\ &(I_{A_1}, I_{F_3}, ends, \emptyset), (I_{A_2}, I_{F_1}, starts, \emptyset), (I_{F_1}, I_{F_2}, meets, \emptyset)\} \\ IR_3 = \{&(I_{A_2}, I_{F_1}, starts, \emptyset), (I_{F_1}, I_{A_1}, starts, \emptyset), (I_{A_3}, I_{F_1}, during, 9), \\ &(I_{A_4}, I_{F_3}, equal, \emptyset), (I_{A_4}, I_{F_2}, equal, \emptyset), (I_{F_3}, I_{A_1}, ends, \emptyset)\}. \end{aligned}$$

3.6 Exemplary Applications

With the definitions given in the previous sections, we can now calculate values for algorithms (see Sect. 3.6.1) and give precise definitions of user interactions and resulting video behavior (see Sects. 3.6.2 and 3.6.3). These can also be used to calculate timing information in multimedia documents as, for example, done by Meixner in [38].

3.6.1 Basic Calculations and Definitions

Further definitions and various basic functions are useful for calculations in algorithms, which will be defined hereafter. Thereby, $\mathcal{X} \in \{\mathcal{A}_D, \mathcal{C}_D\}$ or $\mathcal{X} \in \{\mathcal{A}_V, \mathcal{C}_V\}$ and $\mathcal{X}^k$ is a tuple where k is the number of elements in the tuple.

The **frame rate** r of the video may either be a constant or variable over time t (as described in [12, 28, 42, 46]). A constant frame rate c_r is defined in Function 3.5. It is set to a fixed value c_r for all calculations in this work. Usually, c_r is set to 25 or 30 fps. We use $c_{r_{SFW}}$ as a constant for the slow-forward frame rate, $c_{r_{SBW}}$ as a constant for the slow rewind frame rate, $c_{r_{FFW}}$ as a constant for the fast-forward frame rate, and $c_{r_{FBW}}$ as a constant for the fast rewind frame rate. $\mathbb{R}^+$ is the set of positive real numbers (without zero).

$$r : \mathbb{R}^+ \mapsto \mathbb{N}^+, t \mapsto r(t) := c_r \tag{3.5}$$

A **dimension function** *dim* is needed to get the size/length of a tuple. This basic function is defined in Function 3.6.

$$dim : \mathcal{X}^k \mapsto \mathbb{N}^+, (x_1, \ldots, x_k) \mapsto dim(x_1, \ldots, x_k) := k \tag{3.6}$$

A **projection function** π_i is needed to get a specific value from a tuple. This basic function is defined in Function 3.7.

$$\pi_i : \mathcal{X}^k \mapsto \mathcal{X}, k \in \mathbb{N}^+, (x_1, \ldots, x_k) \mapsto \pi_i(x_1, \ldots, x_k) := x_i,\ 1 \leq i \leq k \tag{3.7}$$

A second **projection function** π_{i_1,i_2} can be used to get a part of a tuple. This basic function is defined in Function 3.8.

$$\begin{aligned} \pi_{i_1,i_2} : \mathcal{X}^j \mapsto \mathcal{X}^k, j, k \in \mathbb{N}^+, k \leq j, (x_1, \ldots, x_j) \mapsto \pi_{i_1,i_2}(x_1, \ldots, x_j) \\ := (x_{i_1}, \ldots, x_{i_2}), 1 \leq i_1 < i_2 \leq j, i_2 - i_1 + 1 = k \end{aligned} \tag{3.8}$$

Furthermore, a generalization from frames/samples and static multimedia elements to "downloadable objects" may simplify some calculations. The static multimedia elements $\mathcal{A}_D/\mathcal{A}_\mathcal{V}$ and the frames/samples $\mathcal{F}_D/\mathcal{F}_\mathcal{V}$ of a multimedia document $\mathcal{D}$ or an annotated interactive nonlinear video $\mathcal{V}$ are joined to a **set of (downloadable) elements** of a multimedia document $\mathcal{E}_D$ (see Definition 31) or of an annotated interactive nonlinear video $\mathcal{E}_\mathcal{V}$ (see Definition 32).

Definition 31 (*Set of Downloadable Elements $\mathcal{E}_D$ of a Multimedia Document*) $\mathcal{E}_D$ is a set of downloadable elements of a multimedia document $\mathcal{D}$, which is defined as the union $\mathcal{E}_D = \mathcal{A}_D \cup \mathcal{F}_D$ of the set of frames/samples $\mathcal{F}_D$ and the set of static multimedia elements $\mathcal{A}_D$ of the multimedia document.

Definition 32 (*Set of Downloadable Elements $\mathcal{E}_\mathcal{V}$ of an Annotated Interactive Nonlinear Video*) $\mathcal{E}_\mathcal{V}$ is a set of downloadable elements of an annotated interactive nonlinear video $\mathcal{V}$, which is defined as the union $\mathcal{E}_\mathcal{V} = \mathcal{A}_\mathcal{V} \cup \mathcal{F}_\mathcal{V}$ of the set of frames/samples $\mathcal{F}_\mathcal{V}$ and the set of static multimedia elements $\mathcal{A}_\mathcal{V}$ of the multimedia document.

In the following functions, the set of downloadable elements $\mathcal{E}_\mathcal{V}$ of an annotated interactive nonlinear video and the set of downloadable elements $\mathcal{E}_D$ of a multimedia document are used interchangeable.

A **size function** s is defined in Function 3.9. It returns the size of an element by returning the length of the n-tuple of bits representing the content of the multimedia element. This function is needed to get the amount of data that has to be downloaded from the server or has to be stored in the cache for each multimedia element.

$$s : \mathcal{E}_D \to \mathbb{N}^+, e_i \mapsto s(e_i) := \begin{cases} dim(e_i) & \text{if } e_i \text{is a frame/sample} \\ dim(\pi_1(e_i)) & \text{if } e_i \text{ is an static multimedia element} \end{cases} \tag{3.9}$$

Function 3.10 returns the **priority** q of a static multimedia element by a projection on the second component of the static multimedia element pair (α_o, Λ). The higher the priority is, the lower is its number. The highest priority is "1". If no priorities are

used, all static multimedia elements are set to priority "1" and are treated with the same priority with which other elements are downloaded.

$$q : \mathcal{A}_D \to \mathbb{N}^+, a_o \mapsto q(a_o) := \pi_2(a_o) = \Lambda \tag{3.10}$$

The **duration of a continuous medium** F in seconds $l(F)$ is calculated by the division of the number of frames/samples of the continuous multimedia element $dim(F)$ by the frame/sample rate c_r at a fixed frame/sample rate. This is expressed by Function 3.11.

$$l : \mathcal{F}_D \to \mathbb{N}^+, F \mapsto l(F) := \frac{dim(F)}{c_r} \tag{3.11}$$

3.6.2 VCR Actions for Continuous Multimedia Elements

Besides play, pause, and stop, several other VCR actions are possible. It is also possible to play the medium backward with the frame rate used for playing it forward. Besides slow- and fast-forward or rewind, it is furthermore possible to jump to a certain frame/sample forward or backward in the currently played medium. Table 3.2 enlists all actions which may be considered in a single medium. For each action, first the current frame is given, and then the current frame rate is stated. Thereby, f_m is a frame in a scene. If the current frame rate is 0, the playback is currently stopped/paused, and if it is c_r, the video is playing with a constant frame rate. After the user interaction, either frame or frame rate changes. Depending on the current state of the video, some restrictions may apply, see remark column in Table 3.2.

3.6.3 Extended Interactivity and Navigation

Annotated interactive nonlinear videos (as described in Sect. 3.5.2) provide additional features besides the VCR actions described in Sect. 3.6.2. The following facts are summarized in Table 3.3. As in Table 3.2, pre- and post-interaction frame and frame rate are shown. Because scene changes may occur, also an index for the scene is given for current and new frame. We assume that the selection panels or quizzes are usually displayed after a scene ends. The user decides which scene should be displayed next either by selecting it directly in a button panel or by solving a quiz. In the latter case, the follow-up scene is chosen by the score reached in the quiz. Each score is assigned to a point range of a scene which is then selected accordingly. Jumps in the whole video which are not depending on the underlying graph structure between the scenes are selections in a table of contents or a selection in search results. When a user opens the table of contents, the video may stop and continue playing after a user selection, or it may continue playing, depending on the positioning of the table of contents (side area of the player or overlay on the video). The selected entry starts the

Table 3.2 Different intrascene VCR actions

Action	Current frame	Current frame rate	New frame	New frame rate	Remark
Stop	f_m	c_r	f_σ	0	
Pause	f_m	c_r	f_m	0	
Play forward	f_m	0	f_{m+1}	c_r	f_m not last frame of a scene
	f_m	c_r	f_{m+1}	c_r	f_m not last frame of a scene
Play backward	f_m	0	f_{m-1}	c_r	f_m not first frame of a scene
	f_m	c_r	f_{m-1}	c_r	f_m not first frame of a scene
Jump forward	f_m	c_r	f_{m+x}	c_r	$x \in \mathbb{N}^+, f_m, f_{m+x}$ in same scene
Jump backward	f_m	c_r	f_{m-x}	c_r	$x \in \mathbb{N}^+, f_m, f_{m-x}$ in same scene
Slow-forward	f_m	c_r	f_{m+1}	$c_{r_{SFW}}$	$c_r < c_{r_{SFW}}, f_m$ not last frame of a scene
Slow rewind	f_m	c_r	f_{m-1}	$c_{r_{SBW}}$	$c_r < c_{r_{SBW}}, f_m$ not first frame of a scene
Fast-forward	f_m	c_r	f_{m+1}	$c_{r_{FFW}}$	$c_r < c_{r_{SFW}} < c_{r_{FFW}}, f_m$ not last frame of a scene
Fast rewind	f_m	c_r	f_{m-1}	$c_{r_{FBW}}$	$c_r < c_{r_{SBW}} < c_{r_{FFW}}, f_m$ not first frame of a scene
Pan/tilt/zoom	f_m	c_r	f_{m+1}	c_r	User interaction to modify the presentation of the following frames

Table 3.3 Interactive and navigational actions which are possible in a single scene (intrascene) or in-between scenes (interscene)

Action	Current frame	Current frame rate	New frame	New frame rate	Remark
Selection/quiz	$f_{a,m}$	0	$f_{b,1}$	c_r	$f_{a,m}$ last frame of a scene, $f_{b,1}$ first frame of selected successor scene
TOC	$f_{a,m}$	c_r	$f_{a,m}$	0	User interaction to invoke table of contents at frame $f_{a,m}$
	$f_{a,m}$	0	$f_{a,1} \vee f_{b,1}$	c_r	Selection in overlay table of contents at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other selected scene; thereby, the video resumes playing at the new position
	$f_{a,m}$	c_r	$f_{a,1} \vee f_{b,1}$	c_r	Selection in side area table of contents at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other selected scene
Keyword search	$f_{a,m}$	c_r	$f_{a,m}$	0	User interaction to invoke keyword search at frame $f_{a,m}$
	$f_{a,m}$	0	$f_{a,k} \vee f_{b,k}$	c_r	Select annotation in search results at frame $f_{a,m}$, jump to frame $f_{a,k}$ of same scene or $f_{b,k}$ of other scenes where selected annotation is displayed
	$f_{a,m}$	0	$f_{a,1} \vee f_{b,1}$	c_r	Select scene in search results at frame $f_{a,m}$, jump to first frame $f_{a,1}$ of same scene or $f_{b,1}$ of other scenes where selected annotation is displayed

playback of a scene at its beginning. A search is usually carried out during the playback of a scene. It is possible to jump to the beginning of a scene or to an annotation in a scene. Interactive functions like pan, tilt, and zoom have no influence on the order of the displayed frames or the frame rate. They may rather increase the download volume, because higher resolutions of single frames or other camera positions are needed at client side [38].

3.7 Conclusion

In this chapter, we propose formal definitions of temporal models as well as other functions that are important for multimedia synchronization, scheduling, and management in applications. Thereby, we focus on the most important existing temporal models, namely, point-based, event-based, and interval-based temporal models which have been proposed and described in previous related work. We summarize the descriptions given in related literature, show advantages and disadvantages of the temporal models, and point out issues that originate from these temporal models. While the temporal models have been widely used, no commonly used formal and precise way of defining them is available. This makes transforming one temporal model into another or performing calculations on one of the temporal models tedious. To overcome this issue, we present formal definitions for commonly used relationships, timing, and interactivity of each temporal model which can easily be extended with additional features. We use each definition in a small example to show its practical usage for defining presentations. After that, we give further definitions of basic functions that are useful for calculations on the temporal models and for the definition of algorithms. We also show how the proposed formalized temporal models and definitions can be used to describe possible user interaction and the following reaction of the video.

In the era of Web with technologies like HTML5, Adaptive Streaming (e.g., DASH), user-generated live video, and Netflix, these temporal models can be used to describe synchronization constraints, to describe scheduling algorithms, or to manage elements during download or streaming and while they reside in the (client) cache. The temporal models make calculations in theoretical frameworks possible which can then be transformed into real-world technologies after initial tests. This is especially useful for standards that are not finished yet or where browser implementations are still in development.

Appendix

Table 3.4 List of symbols

Symbol	Explanation
A	Symbol for the display duration of a static multimedia element
A_i	Symbol for the ith display duration
F	Symbol for a continuous media element
F_i	Symbol for the ith continuous media element
I	Symbol for an interval
I_e	Symbol for an interval for multimedia element e
R	Symbol for a relationship between two intervals
T	Symbol for a time interval
a	Symbol for an annotation
a_i	The ith annotation of a video
c	Symbol for constant values
c_r	Constant for the frame rate (normal speed)
$c_{r_{SFW}}$	Constant for the frame rate (slow-forward)
$c_{r_{SBW}}$	Constant for the frame rate (slow rewind)
$c_{r_{FFW}}$	Constant for the frame rate (fast-forward)
$c_{r_{FBW}}$	Constant for the frame rate (fast rewind)
dim	Symbol for the function returning the length of a tuple
e	Symbol for a (downloadable) element
ea	Symbol for an executable action
ee	Symbol for an element event
ep	Symbol for an event point
e_i	The ith element of set $\mathcal{E}_D$
f	Symbol for a frame/sample
f_σ	Symbol for the start frame/sample
f_ϵ	Symbol for the end frame/sample
f_i	Symbol for the ith frame/sample
$f_{i,m}$	The mth frame/sample of scene i
i	Index
j	Index
j_i	Last frame index of scene i
k	Index
l	Symbol for the duration function
n	Index

(continued)

Table 3.4 (continued)

Symbol	Explanation
p	Symbol for a scene
p_σ	Symbol for the start scene
p_ϵ	Symbol for the end scene
p_i	Scene i in set $\mathcal{P}_\mathcal{V}$
q	Symbol for the function returning the priority of an element
r	Symbol for the frame rate function
s	Symbol for the function returning the size of an element
t	Symbol for the time
t_i	The ith point in time
tl	Symbol for a timeline
w	Symbol for a button
w_j	The jth button
x	Index
α	Symbol for the content of an annotation
α_o	The content of the oth annotation in set $\mathcal{A}_D$
δ	Symbol for the transition function of the DFA
ϵ	Symbol for the end of the video
Λ	Priority of an annotation
π	Symbol for the projection function
π_i	Symbol for the projection function on the ith element of a tuple
$\pi_{i,j}$	Symbol for the projection function on the ith and jth element of a tuple
σ	Symbol for the start of the video
Σ	Input symbols of the DFA
$\mathcal{D}$	Symbol for a multimedia document
$\mathbb{N}$	Set of natural numbers
$\mathbb{N}^+$	Set of positive natural numbers (without zero)
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^+$	Set of positive real numbers (without zero)
$\mathcal{A}_D$	Set of static multimedia elements of $\mathcal{D}$
$\mathcal{A}_\mathcal{V}$	Set of static multimedia elements of $\mathcal{V}$
$\mathcal{C}_D$	Set of continuous multimedia elements of $\mathcal{D}$
$\mathcal{E}_D$	Set of (downloadable) elements of $\mathcal{D}$
$\mathcal{E}_\mathcal{V}$	Set of (downloadable) elements of $\mathcal{V}$
$\mathcal{EA}$	Set of executable actions
$\mathcal{EA}_a$	Set of executable actions for static multimedia elements
$\mathcal{EA}_F$	Set of executable actions for dynamic multimedia elements
$\mathcal{EE}$	Set of element events
$\mathcal{EP}$	Set of event points

(continued)

Table 3.4 (continued)

Symbol	Explanation
$\mathcal{F}_D$	Set of frames/samples of D
$\mathcal{F}_\mathcal{V}$	Set of frames/samples of $\mathcal{V}$
$\mathcal{I}$	Set of intervals
$\mathcal{IR}$	Set of interval relations
$\mathcal{N}_\mathcal{V}$	Set of transitions of $\mathcal{V}$
$\mathcal{P}_\mathcal{V}$	Set of scenes of $\mathcal{V}$
$\mathcal{RS}$	Set of relationships
$\mathcal{X}$	Random set of single elements
$\mathcal{X}^k$	Random set of k-tuples
$\mathcal{V}$	Symbol for an interactive nonlinear video
$\exists$	Existential quantifier ("there exists")
$\exists!$	Existential quantifier ("there exists exactly one")
$\nexists$	Existential quantifier ("there does not exist")
$\forall$	Universal quantifier ("for all")
$\cap$	Intersection of sets
$\cup$	Union of sets
$\lvert\circ\rvert$	Cardinality of a set

Definitions

Multimedia element A multimedia element is an image, a video, an audio, a text or any other type of audiovisual medium. It is the atomic object of any multimedia document.

Annotation An annotation is additional information displayed with a main medium. It consists of an anchor attaching it to the main medium and a body. The body of an annotation is a multimedia element that can be shown in a player [39].

Static multimedia element Static multimedia elements are time independent and always show the same content, like images and/or text.

Continuous multimedia element Continuous multimedia elements are time dependent showing/playing different contents over time, like videos or audios.

Hyperlinked media Hyperlinked media are multimedia elements which are linked with each other by hyperlinks (as known from hypertext). Static media may be clickable or have clickable areas. Continuous media may provide links depending on the media time.

Media synchronization Synchronization of multimedia elements requires mechanisms to prepare the media for display (i.e. pre-fetch, buffering, rendering) and to ensure that timing constraints are met.

Multimedia document A multimedia document is a *self-contained* presentation of linked and synchronized multimedia elements which allows user interaction and navigation. Usually it is about a certain topic.

Point A (time) point is a precise moment in time [41]. It is synchronized with a clock.

Event An event is something that happens or takes place [41]. It may be triggered by a clock or by a user interaction.

Interval A (time) interval is the time between start and end of a time span.

Hypermedia Hypermedia is an extension to hypertext providing multimedia facilities, such as those handling sound and video [41]. Keeping the hyperlink structure from hypertext, multimedia elements of different types are added.

Multimedia Multimedia uses a variety of artistic or communicative media that are presented in one presentation [41].

Passive multimedia Passive multimedia presentations are started and then watched with little to no interaction. Available forms of interaction are starting, pausing, and stopping the presentation.

Active multimedia Active multimedia presentations allow more interaction compared to passive multimedia presentations. They may have hyperlinks or other interactive control elements.

References

1. AFNOR Expert Group: Multimedia synchronization: Definitions and model, input contribution on time variant aspects and synchronization in oda-extensions. ISO IE JTC 1/SC 18/WG3 (1989)
2. Allen, J.F.: Maintaining knowledge about temporal intervals. Commun. ACM **26**(11), 832–843 (1983)
3. Appelt, W., Scheller, A.: HyperODA–going beyond traditional document structures. Comput. Stand. Interfaces **17**(1), 13–21 (1995)
4. Benbernou, S., Makhoul, A., Hacid, M.S., Mostefaoui, A.: A spatio-temporal adaptation model for multimedia presentations. In 7th IEEE International Symposium on Multimedia (ISM'05), pp. 8–pp. Dec (2005)
5. Blakowski, G., Steinmetz, R.: A media synchronization survey: reference model, specification, and case studies. IEEE J. Sel. Areas Commun. **14**(1), 5–35 (1996)
6. Blakowski, G., Hübel, J., Langrehr, U., Mühlhäuser, M.: Tool support for the synchronization and presentation of distributed multimedia. Comput. Commun. **15**(10), 611–618 (1992)
7. Boll, S., Klas, W., Westermann, U.: A comparison of multimedia document models concerning advanced requirements. Technical Report UIB-1999–01, DBIS (1999)
8. Boll, S., Klas, W.: ZYX—A Semantic Model for Multimedia Documents and Presentations, pp. 189–209. Springer, US, Boston, MA (1999)
9. Buchanan, M.C., Zellweger, P.T.: Automatic temporal layout mechanisms. In: Proceedings of the First ACM International Conference on Multimedia, MULTIMEDIA '93, pp. 341–350. ACM, New York, NY, USA (1993)
10. Buchanan, M.C., Zellweger, P.T.: Automatic temporal layout mechanisms revisited. ACM Trans. Multimedia Comput. Commun. Appl. **1**(1), 60–88 (2005)
11. Campbell, and A. N. Habermann. The specification of process synchronization by path expressions. In: Operating Systems, Proceedings of an International Symposium, pp. 89–102. Springer, London, UK (1974)

12. Chen, J.-J., Hang, H.-M.: Source model for transform video coder and its application ii. variable frame rate coding. IEEE Trans. Circuits Syst. Video Technol. **7**(2), 299–311 (1997)
13. Courtiat, J.P., De Oliveira, R.C.: Proving temporal consistency in a new multimedia synchronization model. In: Proceedings of the 4th ACM International Conference on Multimedia, MULTIMEDIA '93, pp. 141–152. ACM, New York, NY, USA (1996)
14. Drapeau, G.D.: Synchronization in the maestro multimedia authoring environment. In: Proceedings of the First ACM International Conference on Multimedia, MULTIMEDIA '93, pp. 331–339. ACM, New York, NY, USA (1993)
15. Duda, A., Keramane, C.: Structured temporal composition of multimedia data. In: 1995 Proceedings International Workshop on Multi-Media Database Management Systems, pp. 136. Aug 1995
16. Eijk, P.V., Diaz, M.: Formal Description Technique Lotos: Results of the Esprit Sedos Project. Elsevier Science Inc., New York, NY, USA (1989)
17. Euzenat, J., Layaïda, N., Dias, V.: A semantic framework for multimedia document adaptation. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'2003, pp. 31–36. Morgan Kauffman, Acapulco, Mexico, Aug 2003
18. Fiume, E., Tsichritzis, D., Dami, L.: A temporal scripting language for object-oriented animation. In: EG 1987-Technical Papers Eurographics Association (1987)
19. Fujikawa, K., Shimojo, S., Matsuura, T., Nishio, S., Miyahara, H.: Multimedia presentation system "harmony" with temporal and activemedia. In: Multimedia for Now and the Future, USENIX (1991)
20. Gibbs, S.J., Breiteneder, C., Tsichritzis, D.: Audio/video databases: an object-oriented approach. In: Proceedings of the 9th International Conference on Data Engineering, pp. 381–390. IEEE Computer Society, Washington, DC, USA (1993)
21. Gotthardt, K.: Grundlagen der Informationstechnik. LIT Verlag Münster, Einführungen–Informatik (2001)
22. Hirzalla, N., Falchuk, B., Karmouch, A.: A temporal model for interactive multimedia scenarios. IEEE Multimedia **2**(3), 24–31 (1995)
23. Hoepner, P.: Synchronisation der Präsentation von Multimedia-Objekten, pp. 455–464. Springer, Berlin Heidelberg (1991)
24. Illik, J.: Formale Methoden der Informatik: Von der Automatentheorie zu Algorithmen und Datenstrukturen. Expert, Reihe Technik (2009)
25. ISO/EIC: Iso/iec 13522-6:1998(en) information technology—coding of multimedia and hypermedia information. Website (1998)
26. ISO/IEC: Information technology—hypermedia/time-based structuring language (hytime) (iso/iec jtc 1/sc 34) (1997)
27. Kim, W., Kenchammana-hosekote, D., Lim, E.P., Srivastava, J.: Synchronization relation tree: a model for temporal synchronization in multimedia presentations. Technical Report (1992)
28. Kim, J.W., Kim, Y.-G., Song, H., Kuo, T.-Y., Chung, Y.J., Kuo, C.-C.J.: Tcp-friendly internet video streaming employing variable frame-rate encoding and interpolation. IEEE Trans. Circuits Syst. Video Technol. **10**(7), 1164–1177 (2000)
29. Kretz, F., Colaitis, F.: Standardizing hypermedia information objects. Comm. Mag. **30**(5), 60–70 (1992)
30. Little, T.D.C., Ghafoor, A.: Interval-based conceptual models for time-dependent multimedia data. IEEE Trans. Knowl. Data Eng. **5**(4), 551–563 (1993)
31. Little, T.D.C., Ghafoor, A.: Scheduling of bandwidth-constrained multimedia traffic. In: Proceedings of the 2nd International Workshop on Network and Operating System Support for Digital Audio and Video, pp. 120–131. Springer, London, UK (1992)
32. Little, T.D.C., Ghafoor, A.: Synchronization and storage models for multimedia objects. IEEE J. Sel. Areas Commun. **8**(3), 413–427 (1990)
33. Little, T.D.C., Ghafoor, A.: Spatio-temporal composition of distributed multimedia objects for value-added networks. Computer **24**(10), 42–50 (1991)
34. Macromind Director: Version 3.0: Overview Manual. MacroMind (1991)

35. Meixner, B., Einsiedler, C.: Download and cache management for HTML5 hypervideo players. In: Proceedings of the 27th ACM Conference on Hypertext and Social Media HT '16, pp. 125–136. ACM, New York, NY, USA (2016)
36. Meixner, B., John, S., Handschigl, C.: Siva suite: framework for hypervideo creation, playback and management. In: Proceedings of the 23rd ACM International Conference on Multimedia, MM '15, pp. 713–716. ACM, New York, NY, USA (2015)
37. Meixner, B., Kosch, H.: Interactive non-linear video: definition and xml structure. In: Proceedings of the 2012 ACM Symposium on Document Engineering, DocEng '12, pp. 49–58. ACM, New York, NY, USA (2012)
38. Meixner, B.: Annotated interactive non-linear video–software suite, download and cache management. Ph.D. thesis, Universität Passau (2014)
39. Meixner, B.: Hypervideos and interactive multimedia presentations. ACM Comput. Surv. **50**(1), 9:1–9:34 (2017)
40. Ogawa, R., Harada, H., Kaneko, A.: Scenario-based hypermedia: a model and a system. In: Rizk, A., Streitz, N., Andre, J. (eds.) Hypertext: Concepts, Systems and Applications–Proceeding the First European Conference on Hypertext, pp. 38–51. Cambridge University Press, Cambridge (1990)
41. Oxford University Press: British & world english. Website https://en.oxforddictionaries.com/ (2017). Accessed 03 May 2017
42. Pan, F., Lin, X., Rahardja, S., Lim, K.P., Li, Z.G., Wu, D.J., Wu, S.: Proactive frame-skipping decision scheme for variable frame rate video coding. In: 2004 IEEE International Conference on Multimedia and Expo 2004. ICME '04, Vol. 3, pp. 1903–1906. IEEE (2004)
43. Poole, L.: Quicktime in motion. MacWorld. **8**(9), 154–159 (1991)
44. Rousseau, F., Duda, A.: An execution architecture for synchronized multimedia presentations, pp. 42–55. Springer, Berlin, Heidelberg (1998)
45. Shepherd, D., Salmony, M.: Extending osi to support synchronization required by multimedia applications. Comput. Commun. **13**(7), 399–406 (1990)
46. Shue, J.-S., Hsieh, C.-H., Tsai, H.-S., Wang, C.-C.: Variable-rate video codec using frame adaptive finite-state vector quantization. In: 1993 IEEE International Symposium on Circuits and Systems, 1993 ISCAS '93, vol. 1, pp. 28–31. IEEE (1993)
47. Steinmetz, R.: Synchronization properties in multimedia systems. IEEE J. Sel. A. Commun. **8**(3), 401–412 (1990)
48. Vazirgiannis, M., Kostalas, I., Sellis, T.: Specifying and authoring multimedia scenarios. IEEE Multimedia **6**(3), 24–37 (1999)
49. Wahl, T., Rothermel, K.: Representing time in multimedia systems. In: Proceedings of the International Conference on Multimedia Computing and Systems, pp. 538–543. May 1994
50. Wahl, T., Wirag, S., Rothermel, K.: Tiempo: temporal modeling and authoring of interactive multimedia. In: Proceedings of the International Conference on Multimedia Computing and Systems, pp. 274–277. May 1995