Structured Temporal Composition of Multimedia Data

Andrzej Duda, Chérif Keramane

CNRS-IMAG,

2, rue Vignate, F-38610 Gières, France

Abstract. Many existing models for temporal composition of multimedia data are based on the Allen relations [1]. However, the relations are not suitable for composition: they are descriptive (do not reflect causal dependency between intervals), they depend on interval durations (changing duration may modify relations), and they may lead to temporal inconsistency (contradictory relations specified for intervals). We define a set of operators expressing causal relations between intervals. They can be used to form nested *interval expressions.* Interval expressions are independent of interval durations and allow encapsulation and structured nesting, so that it is possible to define complex multimedia presentations in a well-structured manner.

1 Introduction

Multimedia integrates different media having inherent temporal behavior. Taking into account the diverse time-dependent data types result in new issues that do not arise in traditional databases. The essential issues are the management of different time-dependent data types and specification of their temporal composition. Temporal composition consists of defining presentations of complex multimedia data that relies on synchronization among different media as well as within each medium. Interactive multimedia add complexity because user interactions may modify and control presentations.

Integration of interactive multimedia presentations within databases must address the following requirements:

- *Structuring:* it should be possible to define complex multimedia presentations in a well-structured manner,

- *Encapsulation:* we need a means for encapsulation and nesting of structured entities,
- *Interactivity:* we must take into account user interactions; this requirement results in a model with time intervals of possibly unknown durations,
- *Content-based access:* we should provide flexible associative access based on the content, structure and temporal composition of multimedia data,
- *Content-sensitivity:* an interesting feature would be to make a presentation sensitive to the content—the flow of a presentation can be changed according to the content of media entities.

Composition of interactive multimedia presentations may involve expressing such relations as:

1. *start video segment C after the end of both video segment A and B,*
2. *display image X when the user clicks on the person presented in the video segment A,*
3. *present a choice to the user when the beginning of "Dollars and Sense" is detected,*
4. *find all scenes with Sharon Stone.*

In this paper, we will focus on temporal specification of interactive multimedia presentations. So, we are interested in expressing composition relations such as in Example 1. The requirements above imply that the desired temporal composition model must provide high-level constructs for defining relations between time intervals which have possibly unknown durations. For example, user interactions make the durations of media segments unknown or unpredictable. In addition to that, the constructs must allow encapsulation and structured nesting.

Many existing models for temporal composition of multimedia data are based on the Allen relations [1]. However, the relations are not suitable for temporal composition. First, the relations are descriptive: they do not reflect causal dependency between intervals, but they rather represent temporal coincidence. Second, since the relations depend on interval duration, changing duration may modify the relation that exists between the intervals. Third, composition based on the relations may lead to temporal inconsistency, because contradictory relations may be specified for intervals.

We propose a composition model based on intervals of unknown duration and on operators that express causal relations between intervals. Operators take time intervals as arguments and yield another interval as a result. They can be used to form nested *interval expressions* allowing specification of temporal compositions in a well-structured way. Interval expressions provide a means of encapsulation and structuring: compound encapsulated intervals can be specified in terms of elementary media objects as building blocks.

We build upon the previous work on *Algebraic Video* [14] that uses a set of basic operations on video segments to create a desired video stream. The video algebra consists of operations for temporally and spatially combining video segments, and for attaching attributes to these segments. Algebraic Video focus on content-based access to video libraries with query based discovery and algebraic combination of video segments of interest. We adopt the approach consisting of defining a set of operators to form nested expressions.

In the remainder of this paper, we present existing models (Section 2), analyze possible arrangements between two time intervals (Section 3), present the proposed composition model (Section 4), and outline conclusions (Section 5).

2 Existing models

Existing temporal models for multimedia can be divided into two classes: *point-based* and *interval-based* [13]. In point-based models, the elementary units are points in a time space. Each event in the model has its associated time point. The time points arranged according to some relations such as *precede*, *simultaneous* or *after* form complex multimedia presentations. An example of the point-based approach is *timeline*, in which media objects 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. The model is applied in HyTime [9] and in several other propositions [8], [7]. The model is well suited for temporal composition of media segments of known durations, however it falls short for unknown durations.

Other authors have proposed to use relations between interval end points for temporal composition of multimedia (temporal point nets [3], MME [6]). However, their use is difficult and results in complicated, unstructured graphs. In addition to that, their use may lead to an inconsistent specification in which contradictory conditions are specified for intervals. In this case, a verification algorithm (called sometimes a temporal formatter) must check for inconsistency [3].

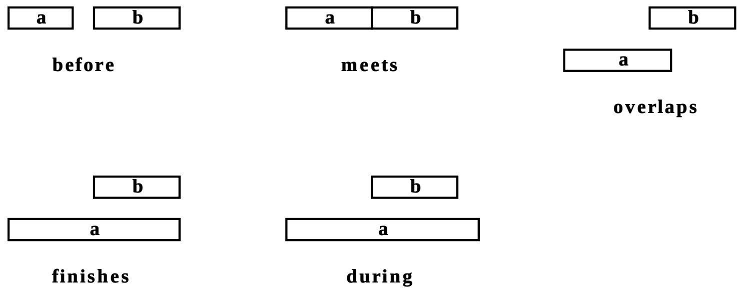

Fig. 1. Relations change when the interval duration is modified.

Interval-based models consider elementary media entities as time intervals ordered according to some relations. Existing models are mainly based on the relations defined by Allen for expressing the knowledge about time [1]. Giving any two intervals, they can be arranged according to seven relations : *before, meets, overlaps, finishes, during, starts, equals.* However, using Allen relations for multimedia composition faces several problems. First, the relations were designed to express existing relationships between intervals of fixed duration and not for specifying relationships that must be always satisfied even when interval durations are changed. Consider for example an existing relation *before* between intervals a and b (see Figure 1). When we increase the duration of interval a, the relation changes from *before* to *during* passing through intermediate relations *meets*, *overlaps*, and *finishes*. This drawback makes the Allen relations not suitable for specifying composition of intervals with unknown duration.

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. For example, relation *meets* only states that the end of the first interval coincides with the end of the second one, but it does not say whether the first interval *starts* the second one, whether the second interval *stops* the first one or whether it is a pure coincidence (see Figure 2). So, the Allen relations can be useful for characterizing an existing, *instantiated* presentation (a presentation for which

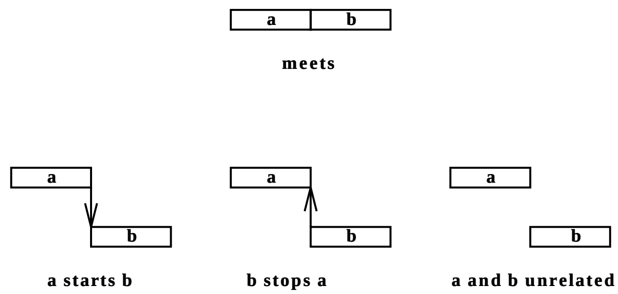

Fig. 2. *meets* represents temporal coincidence not a functional relationship.

all start and termination instants of media segments are known).

The third problem with the Allen relations is related to inconsistent specifications that can be introduced to a multimedia presentation. Detecting inconsistent specification requires algorithms of complexity $[O(N^2)]$, where N is the number of intervals [1].

Some authors propose to use the Allen relations to specify a multimedia database schema. For example, Little and Ghafoor propose an OCPN model equivalent to the Allen relations [11, 12]. They do not take into account possible unknown durations of intervals and to prepare an instantiated presentation (a presentation in which all interval end points are determined), they must traverse the tree of interval relations to get deadlines used to schedule the presentation. They propose an algorithm for constructing a database schema from temporal relations, however, when interval durations are modified, the schema must be modified too.

King proposes a different formalism based on an interval temporal logic [10]. He shows how the Allen relations can be expressed using temporal logic formula. Although his formalism has solid mathematical bases, composition of multimedia presentations using declarative formula is awkward—logic formula do not correspond to the mental image that an author uses in conception. Moreover, to be useful, the formalism must be supported by a consistency checker and an interpreter to execute a given temporal specification.

Unlike previous models, *interval expressions* offer a different paradigm that allow specification in a structured manner of causal (or functional) relations between intervals of unknown duration. We explore the same approach as *Algebraic Video* [14] that uses a set of basic operations on video segments. The video algebra defines three temporal operators for combining video segments: *concatenation*, *parallel* and *parallel-end*. Other operators such as *union*, are related to attributes and content-based access.

3 Analysis of interval relations

Our elementary entities are *time intervals*. Time interval a is defined by the end points ($\underline{a} \leq \bar{a}$) as $a = \{t \mid \underline{a} \leq t \leq \bar{a}\}$. The duration of interval a is $\bar{a} - \underline{a}$ and can be constant (e.g. *5 seconds*), dependent on the intrinsic playing time of the medium (e.g. *playing time of a video segment*) or unspecified (e.g. *user interaction* or *live feed*). Causal relations or actions are attached to interval end points. This assumption does not limit our model, because if it is necessary to perform an action in the middle of an interval, we can divide it into two intervals and attach the action to some of the end points. Each end point of an interval may perform an action on any of the end points of another interval: the action aimed at the beginning of an interval corresponds to *start* or activation and the action aimed at the end of an interval corresponds to *stop* or termination.

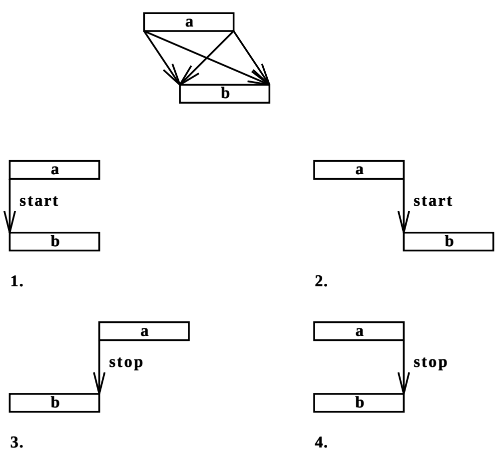

Fig. 3. Four possible relations between intervals.

Let us consider relationships between any two time intervals (see Figure 3). The arrows in the figure represent actions performed at the end points (note that the actions represent causal relations that have real system interpretation which is not the case for the Allen relations).

There are four possible relations with a single action involving a and b (shown in Figure 3) and four inverse relations involving b and a. In addition, we can consider relations with more than one action. Many of such relations are trivial, for example relations $\underline{a} \longrightarrow \underline{b}$ and $\underline{a} \longrightarrow \bar{b}$ make the duration of b null. There are two non trivial relations concerned with equality of interval durations (see Figure 4). In the first relation, *a starts b* and *a stops b*. In the second one, *a starts b* and *b stops a*. The distinction between the two relations is important, because any of the intervals may be of unknown duration, for example, the duration can be controlled by the user or can be determined externally by a live feed.

Fig. 4. Two relations for equality.

We could have used a model in which actions are attached to interval end points, but we want to define a higher-level model that provides encapsulation and structuring.

4 Temporal composition model

Our model for structured temporal composition of multimedia is based on binary operators that represent some of the previously described relations between intervals of unknown duration. An operator takes two intervals as arguments and returns an interval as a result:

$$interval \ \texttt{operator} \ interval \mapsto interval$$

The operators can be used to form nested *interval expressions* allowing specification of temporal compositions by means of encapsulation in a well-structured way. The structure defined by an interval expression is invariant with respect to the duration of component intervals.

Forming interval expressions provides encapsulation boundaries. From the external point of view, the result interval has some end points on which actions can be performed. From the internal point of view, the operator defines the internal temporal behavior of the result interval completely—the activation of the beginning end point in the result interval implies the activation of all its internal components. In this way, the result interval forms an autonomous entity separated from outside. For example, when *a starts b* (Figure 3), the result interval is defined by the end points $(\underline{a}, \bar{b})$. The activation of the result interval implies activation of a that in turn causes activation of b. The two relations involving *start* in Figure 3 and the equal relations in Figure 4 satisfy this encapsulation condition. However, relations 3 and 4 (Figure 3) in which *a stops b* cannot be encapsulated, because inside the encapsulation boundaries, we are not able to determine at which instant b must be activated. For this reason, we do not define operators to express these relations.

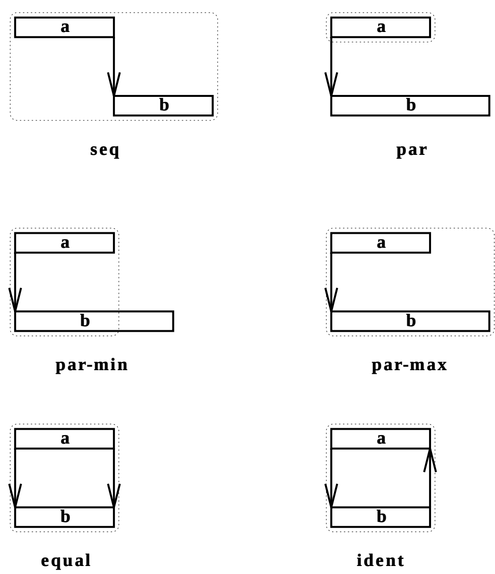

Fig. 5. Temporal composition operators

We propose a sequential operator for starting one interval after another one and several variants of a parallel operator for specifying different termination conditions. The operators are presented in Figure

5. Thin boxes show the result interval for each operator (encapsulation boundary). The semantics of the operators is defined below.

a `seq` $b \mapsto (\underline{a}, \bar{b})$	defines composition in which the end of interval a *starts* interval b.
a `par` $b \mapsto (\underline{a}, \bar{a})$	defines composition in which the beginning of interval a *starts* interval b.
a `par-min` $b \mapsto (\underline{a}, \min(\bar{a}, \bar{b}))$	defines composition in which the beginning of interval a *starts* interval b; the result interval is stopped when the first of the two interval terminates.
a `par-max` $b \mapsto (\underline{a}, \max(\bar{a}, \bar{b}))$	defines composition in which the beginning of interval a *starts* interval b; the result interval is stopped when the last of the two interval terminates.
a `equal` $b \mapsto (\underline{a}, \bar{a})$	defines composition in which interval a *starts* and *stops* interval b.
a `ident` $b \mapsto (\underline{a}, \bar{b})$	defines composition in which the beginning of interval a *starts* b and the end of interval b *stops* a.

4.1 Discussion

Operators `par-min` and `par-min` are commutative and the other operators are not. The equality operators are in fact variants of a parallel operator.

The arguments of the operators and their result are intervals, so the operators can be used to form nested expressions. Composition consists of considering intervals as building blocks that can be put together, encapsulated and nested. Nesting allows building more complex hierarchical well structured presentations. For example, the following complex scenario (see Figure 6) can be specified as:

(a `seq` e) `par-max` (b `seq` (c `equal` d))

As nothing comes for free, there is a price to pay for duration independence and for structured composition in our model. Not all scenarios can be expressed using our operators. For example, the scenario presented in Figure 7 [6] cannot be expressed, because of interleaved start and stop actions on parallel branches. Temporal point nets ([3], [6]) can express such scenarios, however, the resulting graph

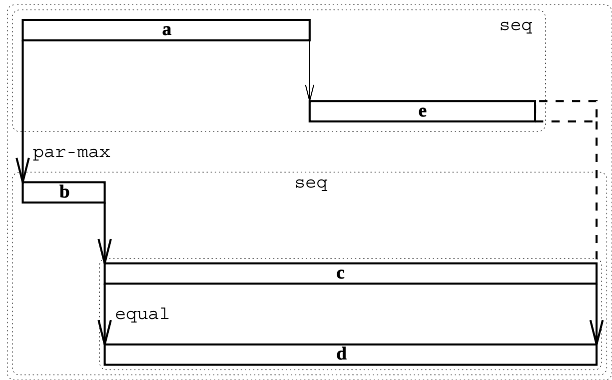

Fig. 6. Example scenario.

becomes complicated and difficult to manipulate or to modify. In addition, composition using temporal point nets does not avoid temporal inconsistencies difficult to detect and to correct. Interval expressions guarantee composition consistency, so there is no need for any verification.

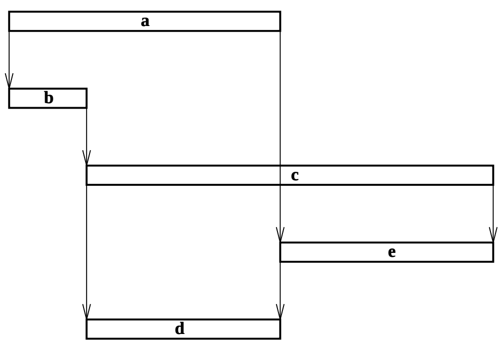

Fig. 7. Another example scenario.

This issue is similar to the debate on the structured approach to programming or on parallel programming constructs (*fork-join* [4] versus *cobegin-coend* [5]). In the structured approach to programming, some constructs such as *goto* are eliminated in order to simplify program structure. However, some programs that use *goto* statement cannot be expressed using structured programming constructs. Similarly, *fork-join* allows more flexible programming compared to *cobegin-coend*, however, the latter makes the structure of the program simpler and more readable. Restricted flexibility is compensated by simplicity and semantic consistency. Composition using interval expressions restricts the class of temporal scenarios that can be expressed, but forces methodological structure and encapsulation.

We can also compare our approach to structured document processing as opposed to traditional text formatters such as TEX. Structured documents are treated as compound objects corresponding to the document's logical composition [2]. Their composition may be restricted to fit the structure. On the opposite, TEX allows almost not constrained composition resulting in confusing documents that are difficult to modify and maintain.

Our temporal composition model fits well object-oriented databases. Each elementary medium entity can be represented as an object having temporal behavior. Its duration may be left unspecified. Interval expressions allow definition of compound encapsulated objects that are nested to form a well structured hierarchy of objects. The database schema represents the structure of multimedia presentations. Because of the invariance of the structure with respect to object durations, the schema does not need to be updated if object durations are modified. The structure of multimedia presentation can be used for classification and reuse of subcomponents.

5 Conclusions

We have presented a new model for structured temporal composition of multimedia data. Our work differs from earlier works in two major aspects. Interval expressions are independent of durations and provide a means for composing well-structured multimedia presentations.

The model is a part of a larger project that integrates user interactions, content-based access and content-sensible interactions within a unified framework. The framework will allow association of various attributes and events with media intervals: user interactions, events that depend on the media contents, and attributes that characterize the media contents. We want to separate the multimedia data and their temporal and content composition from other kinds of information such as presentation specification (geometry of windows, spatial layout) and resource allocation (channel placement). Our goal is to provide a framework for integration of live feeds, video and audio collections into databases of interactive multimedia documents.

Acknowledgments

We are grateful to Nabil Layaida and Stéphane Perret for useful comments and fruitful discussions.

References

1. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11), November 1983.
2. J. André, R. Furata, and V. Quint. *Structured Documents.* Cambridge University Press, 1989.
3. M. C. Buchanan and P. T. Zellweger. Automatic temporal layout mechanisms. In *Proc. First ACM International Conference on Multimedia.*, pages 341–350, Anaheim, CA, August 1993.
4. M. Conway. A multiprocessor system design. In *Proc. 1963 Fall Joint Computer Conference*, volume 24, pages 139–146, Baltimore, Maryland, 1963.
5. E.W. Dijkstra. Co-operating sequential processes. In *Programming Languages, F.Genuys ed.*, pages 43–112, 1968.
6. D. Dingeldein. Modeling multimedia-objects with MME. In *Proc. Eurographics Workshop on Object-Oriented Graphics*, Sintra, Portugal, May 1994.
7. G. D. Drapeau. Synchronization in the MAEstro multimedia authoring environment. In *Proc. First ACM International Conference on Multimedia.*, pages 331–339, Anaheim, CA, August 1993.
8. S. Gibbs, C. Breiteneder, and D. Tsichritzis. Audio/Video databases: An object-oriented approach. In *Proc. 9th IEEE Int. Data Engineering Conference*, pages 381–390, 1993.
9. ISO. Information technology hypermedia/time-based structuring language (HyTime). *ISO International Standard*, (ISO/IEC IS 10744), August 1992.
10. P. R. King. Towards a temporal logic based formalism for expressing temporal constraints in multimedia documents. Technical Report 942, LRI, Université de Paris-Sud, Orsay, France, December 1994.
11. T.D.C Little and A. Ghafoor. Synchronization and storage models for multimedia objects. *IEEE Journal on Selected Areas in Communication*, 8(3):413–427, 1990.

12. T.D.C Little and A. Ghafoor. Interval-based temporal models for time-dependent multimedia data. In *Proc. IEEE Conference on Data and Knowledge Engineering*, Lake Buena Vista, FL, August 1993.

13. T. Wahl and K. Rothermel. Representing time in multimedia systems. In *Proc. IEEE International Conference on Multimedia Computing and Systems.*, Boston, MA, May 1994.

14. R. Weiss, A. Duda, and D.K. Gifford. Content-based access to algebraic video. In *Proc. 1994 International Conference on Multimedia Computing and System, also to appear in IEEE Multimedia*, Boston, MA, May 1994.