Representing Time in Multimedia Systems

Thomas Wahl, Kurt Rothermel

Universität Stuttgart
Institut für Parallele und Verteilte Höchstleistungrechner

{rothermel, wahl}@informatik.uni-stuttgart.de

Keywords

multimedia, synchronization, presentation, temporal model, specification, multimedia documents

Abstract

As multimedia systems deal with a variety of temporally interrelated media items, synchronization is an important issue in those systems. One part of synchronization is the representation of temporal information. In contrast to traditional computing tasks, multimedia imposes new requirements on the representation of time. Specifically, a fine-grained and a flexible temporal model is required. Therefore, a number of temporal models have been suggested by various authors. This paper evaluates and classifies a selection of the most common existing models applying fundamental statements of the time theory and temporal logic. Learning from the deficits of the existing models, a new temporal model based on interval operators is proposed for multimedia systems.

1. Introduction

Multimedia systems integrate a variety of media with different temporal characteristics, e.g. time dependent media, such as video, audio or animation, and time independent media, such as text, graphics and images [Stei90]. In monomedia environments, all media show the same basic temporal behavior. Time does not need any particular attention. Now with the arising multimedia systems, various temporal interrelations between media items become more and more important.

Assuring the correct temporal appearance of the media items is called *synchronization*. The issue of synchronization is twofold. First, the temporal appearance including the interrelations of

presentation items have to be specified. The temporal specification has to be represented for reviewing by the user, presentation planning by the system and storing purposes. Secondly, the multimedia system has to guarantee the temporal constraints when presenting the media items. This is done by providing sufficient resources and real-time processing [BDH+93]. This paper focuses on the first issue of representing time in multimedia environments.

The representation of time has been examined in the context of concurrent processes. Several temporal models have been developed, e.g. CSP [Hoar78], [Hoar85], path expressions [CaHa74]. When applying those models for multimedia, it is observed that temporal relationships are specified in a very coarse-grained manner. To elaborate this, look at a temporal model whose expressive power is restricted to allow processes to be 'sequential' or 'parallel'. Let us examine two multimedia scenarios to determine the interrelations of the presentation items. The first scenario consists of a video sequence that is presented simultaneously with a corresponding audio sequence. The second scenario comprises a video that fades over to a subsequent video. Both scenarios describe parallel actions because both include temporal intervals during which two actions are active. Unfortunately, these types of parallelism cannot be distinguished in the given temporal model. Of course for multimedia, they should be distinguished because a presentation in which the video overlaps the audio sequence in an unspecified manner generally does not satisfy the user. Therefore, a *fine-grained model* of temporal relations including various types of parallelism is necessary.

A second requirement addresses the *flexibility* of temporal models which is needed when not all events are known in advance. Typically, when specifying a multimedia presentation, not all events are known before the presentation is started. Asynchronous events caused by the system or by user interaction often result in a rescheduling of presentation items. E.g. a student might pause a video lecture, look up a definition in a data base and then take a note for his term paper. All the actions are highly indeterministic and cannot be predicted by the supporting multimedia system. Therefore, many temporal relations are unspecified or only partially restricted. To express unspecified or partial relations, a multimedia time model has to be very flexible. Temporal models with totally ordered events generally do not satisfy this criteria.

Finally, the multimedia user needs intuitive abstractions of temporal relations to ease authoring of multimedia presentations. Therefore, *high-level temporal relations* are needed [BuZe93]. E.g. for a synchronous presentation of a video 'v' and its audio 'a', we would like to specify 'v synchronous a' instead of specifying all the details, such as 'a and v start together, are displayed with the same constant speed and end at the same time'.

Since the temporal models developed for process synchronisation do not meet the specific requirements of multimedia, several models have been proposed, and it has been discussed which of the models are appropriate for multimedia. However, this question cannot be answered in general because simple multimedia environments need weak temporal models whereas sophisticated systems require more complex models. To find the appropriate temporal model, we would like to know the expressive power of the existing temporal models. Before assessing the temporal models, it is very helpful to understand the two basic temporal frameworks given in Section 2 and the temporal characteristics of multimedia presentations described in Section 3. Then, we describe, classify and evaluate the most important existing temporal models in Section 4. Learning from the deficits of the existing models, we introduce a new powerful temporal model based on interval operators in Section 5.

2. Basic Temporal Frameworks

Before examining multimedia time models, a basic understanding of the fundamental temporal frameworks is necessary. Depending on their elementary units, two basic classes of time models can be distinguished [vBee92]. In the first class, time is expressed by means of points in a one-dimensional time space [ViKa86] whereas, in a second model class, intervals are the atomic units of the time space [Alle83]. This section introduces the basic models, their elementary units and the relations between them.

2.1 Point-Based Framework

In point-based temporal models, the elementary units are events, which are points in a time space. Given two events in history, only three relations can hold between them. An event can be *before* (<), *simultaneous to* (=) or *after* (>) a second event. The relations <, =, > are called the *basic point relations* (basic PRs).

In contrast to relations in the past, relations between future events might be indefinite. For example, we know that an event e_1 cannot occur after an event e_2. This means that e_1 is before or simultaneous to e_2. This is denoted as $e_1 < e_2 \vee e_1 = e_2$ or as $e_1 \{<,=\} e_2$. Note that e_1 is before or simultaneous to e_2, and it is not known which of the relations will become true. Typically, *indefinite relations* are represented as disjunctions of basic PRs. Since there are 3 basic PRs, $2^3 = 8$ disjunctions exist each representing an indefinite relation. Any of the 8 indefinite relations has an associated symbolic notation. For example, instead of $e_1 \{<,=\} e_2$, we use $e_1 \leq e_2$. The 8 indefinite relations are: $\emptyset$, $\leq$, <, =, >, $\geq$, $\neq$, ?, where '?' is the full set of basic PRs {<, =, >}, $\emptyset$ is the empty set { } and the others are self-explaining. In this paper, we identify the basic relations <, =, > with the indefinite relations {<}, {=} and {>}. Therefore, the basic PRs are a subset of the indefinite PRs.

2.2 Interval-Based Framework

Intervals are the basic units of a time model class suggested by [Alle83], [Bruc72]. There are 13 basic interval relations (basic IRs). Table 1 summarizes the basic interval relations showing the name, the symbol, the inverse and an example for each relation. In this context, x and y represent intervals. Also, a point notation exists for each IR. It is given in the fourth column with Bx denoting the beginning and Ex the end of the interval x.

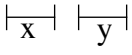
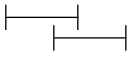
relation	symbol	inverse	conjunctions of point-relations	example	class
x before y	<	>	Bx<Ex<By<Ey		sequential
x meets y	*m*	*mi*	Bx<Ex=By<Ey		sequential
x overlaps y	*o*	*oi*	Bx<By<Ex<Ey		parallel

Table 1: Basic interval relations

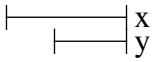
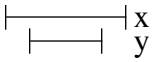
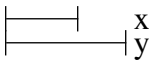
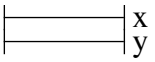
relation	symbol	inverse	conjunctions of point-relations	example	class
y finishes x	*f*	*fi*	Bx<By<Ex=Ey		parallel
y during x	*d*	*di*	Bx<By<Ey<Ex		parallel
x starts y	*s*	*si*	Bx=By<Ex<Ey		parallel
x equals y	=	=	Bx=By<Ex=Ey		parallel

Table 1: Basic interval relations

In analogy to the point relations, 2^{13} indefinite interval relations can be defined as disjunctions of the basic IRs. E.g. if two presentation actions x and y are sequential, we know that 'x is before y' or 'x meets y' or one of the two inverse relations holds. This relation is denoted as {<, m, mi, >}. We also identify the basic IRs with their corresponding indefinite IRs such that the basic IRs are a subset of the indefinite IRs. Table 2 compares the two frameworks in terms of the number of possible relations.

	number of PRs	number of IRs
basic	3	13
indefinite	$2^3 = 8$	$2^{13} = 8192$

Table 2: Number of point and interval relations

2.3 Translations between Representations

As we will show in Section 4, some temporal models proposed for multimedia are point-based, others are interval-based or hybrid. To compare temporal models of different frameworks, we need to translate temporal specifications from one framework to the other. Doing this, we can benefit from essential results proved in temporal logics. This section presents some important statements from temporal theory [Rich89].

Generally, temporal intervals describe the duration of a media item in a presentation environment. So, we use the relations that a temporal model can represent between two intervals to evaluate its expressive power. In a point-based framework, some relations between two intervals are represented as conjunctions of PRs between the four end-points of the two intervals. Four relations between the four end-points of the a pair of intervals can be specified (Figure 1).

By labelling the end-point relations with basic or indefinite PRs, we can find out how many IRs are representable in a point-based framework. Table 2 shows the number of consistent IRs that can be expressed by conjunctions of the given PR set. E.g., conjunctions of the basic PRs set <, = and > just create the basic interval relations. If the larger PR base <, =, >, ? is used, 29 consistent interval relations can be represented. Although the basic PRs generate all basic IRs, an equivalent statement for indefinite relations does not hold. The full set of indefinite PRs gener-

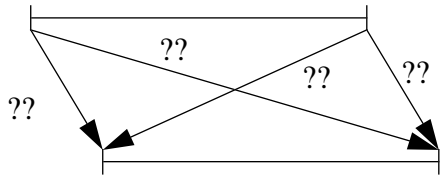


Figure 1: Computing the number of consistent IRs

ates only the small subset of 187 out of 8191 consistent indefinite IRs. For this reason, the expressive power of point-based temporal models is very limited compared to interval-based approaches. In the next section, we will see that the 29 relations generated by the relations <, =, > and ? are especially important in multimedia environments.

point relation base	number of consistent disjunctive IRs
<, =, >	13
<, =, >, ?	29
≤, <, =, >, ≥, ?	82
≤, <, =, >, ≥, ≠, ?	187

Table 3: Disjunctive IRs generated by point relations

3. Characteristics of Multimedia Time Models

Some temporal characteristics observed in multimedia systems are inherent to processing media items. Taking into account the temporal behavior of the media items, specific temporal models tailored to multimedia applications can be defined avoiding complex universal models. However, before adjusting temporal models to multimedia, it is helpful to know which are the relevant relations in multimedia applications.

A point-based model should obtain a representation form for all PRs that have to be specified when composing a multimedia presentation. So, it is interesting to know which PRs do occur in multimedia. Obviously, the basic PRs <, =, > occur because presentation events might be before, simultaneous to or after other events.

To evaluate the indefinite PRs, we have to consider the fact that small inaccuracies are tolerated in multimedia. E.g. in a video-audio presentation, the audience does not notice the skew introduced if the audio is presented too early or too late by some milliseconds [Stei92], [LiKo92], [RoDe92], [BDF+92]. So, we do not need to specify the temporal behavior at exactly one point in time rather it is sufficient to specify the temporal behavior close to each point in time. This implies that there is not any perceptible difference in the presentation if somebody specifies for two events e_1 and e_2 that $e_1 < e_2$ or in the second case $e_1 \leq e_2$. This holds because the audience cannot distinguish whether e_1 is simultaneous to e_2 or e_1 is 1 millisecond before e_2. Therefore, it is sufficient to be able to express only one of the relations $\prec$ or $\leq$. In this paper, we operate with the relations $\prec$ and $\succ$, and do not need the relations $\leq$ and $\geq$. Analogically, the relation $\neq$

differs from the ?-relation only in one point in time. Since there is not perceptible difference between the two relations, we do not need the relation ≠ if we have the relation ?. Observe that we need the relation ? if any basic PR can hold between two events. This indefinite often occurs during the specification and planning process when not all events are known yet. Generally, the ?-relation is responsible for the flexibility of a temporal model because it includes all possible basic PRs.

To summarize, the relations <, = , > and ? are the most important relations in multimedia environments. Powerful point-based temporal models should be able to express at least this set of relations. According to Section 2.3, the PRs <, =, > and ? generate the 29 interval relations. In Annex A, Table 5 enumerates the 29 IRs, and gives a point, an interval and an operator representation for each IR. The operator representation will be explained in Section 5.

A commonly applied temporal model is the time-line by which only the 13 basic IRs are representable. Some authors [HyTi92], [Hoep91], [LiGh90] assure that their temporal models are at least as powerful as the time-line by showing that the 13 basic IRs are expressible within the model. However, it was omitted to determine the power of temporal model, i.e. to show how many and which types of relations can be represented in the model.

Temporal specifications that are restricted to the 13 basic IRs are often over-constraint. Indefinite IRs are needed to avoid this problem. It is observed that indefinite IRs occur frequently in multimedia systems. For example, if we do not care about the end of the presentation components x and y, we issue a 'cobegin' for x and y. The result might be that x ends before, after, simultaneously to y. Note that this cannot be expressed by a single IR because then the relation between the end-points of x and y would be known. We conclude that multimedia needs indefinite IRs.

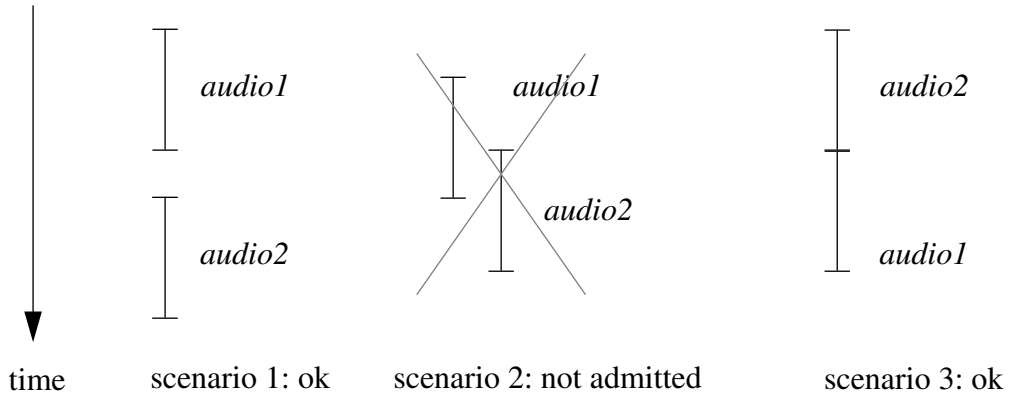


Figure 2: 'Not parallel'

As it was shown in the previous section, some indefinite IRs cannot be represented as conjunctions of PRs. This fact is a major handicap of point-based systems because disjunctions of conjunctions of PRs cannot be represented by most point-based systems. One of these indefinite IRs is the relation 'not parallel' which is needed when limited resources are shared. For example, if there is only one loudspeaker, then two audio sequences should not be presented simultaneously (Figure 2). Therefore, we would like to specify that the audio sequences are not parallel. This is expressed by the indefinite IR {<, m, mi, >}. Represented by PRs, a disjunction is needed: $Ex \leq By \vee Ey \leq Bx$. Consequently, 'not parallel' cannot be represented in point-based systems that do not allow disjunctions.

4. Evaluation of Multimedia Time Models

In the context of multimedia, various temporal models have been proposed by many authors. The temporal models are hard to compare because they are based on fundamentally different approaches of time modelling. This section analyzes the expressive power of the most important temporal models. Specifically, the number of indefinite IRs that can be represented in this model are determined and a classification is given whether a model is mainly point- or interval-based. The latter question is not always easy to answer because some temporal models use intervals as their basic units but their relations address at most one end-point of each interval. Essentially, those models have the same characteristics as point-based approaches.

4.1 Time-Line

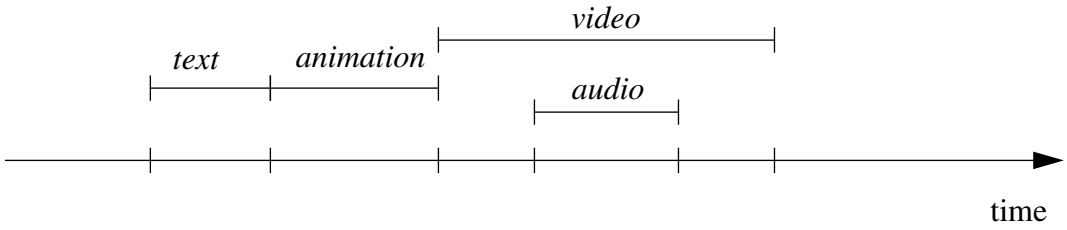


Figure 3: Time line model

The time line model is applied by [BHL91], [Gibb91], [Appl91], [Drap93] and in HyTime [HyTi92]. In the time line model, all events are aligned on a time axis (time line) as it is shown in Figure 3. Since events are the atomic units, the time line model is point-based. All events are totally ordered on a time line. So, exactly one of the PRs <, =, > holds between any pair of events. As all events are totally ordered, it is impossible not to define a relation between any two events. This means that the relation '?' cannot be expressed in the time-line model. This lack of flexibility is a major disadvantage of the time-line model. With <, = and > being the only possible PRs in the time-line model, we can conclude that the 13 basic IRs are the only IRs that are expressible in the time-line model.

4.2 Temporal Point Nets

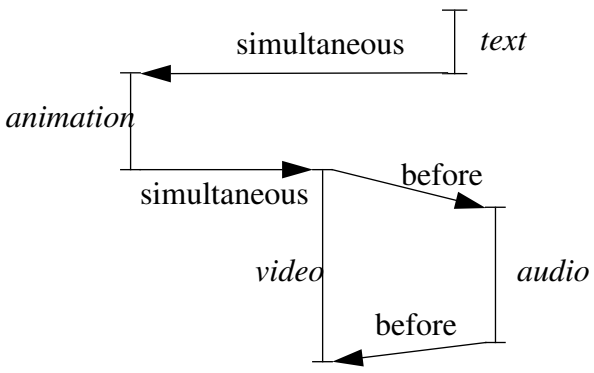


Figure 4: Temporal point net

[BuZe92], [BuZe93] use a point net to represent time specifications (Figure 4). Relations address events establishing temporal equalities (=) and temporal inequalities (<, >). Although [BuZe92] does not mention it, a fourth relation (?) can be specified meaning: The relation between two time points is not restricted. The ?-relation adds a flexibility to the model that cannot be found in the time-line model. Using the PRs <, =, > and ?, 29 IRs can be represented including the 13 basic IRs.

[BuZe92] also defines a relation construct 'before by at least δ' where δ is a delay parameter describing the temporal distance between two events. For δ=0, the point relations ≤ and ≥ can be specified. Then, the PRs <, ≤, =, ≥, >, ? are representable in the point net model generating 82 IRs.

4.3 Timed Petri-Nets

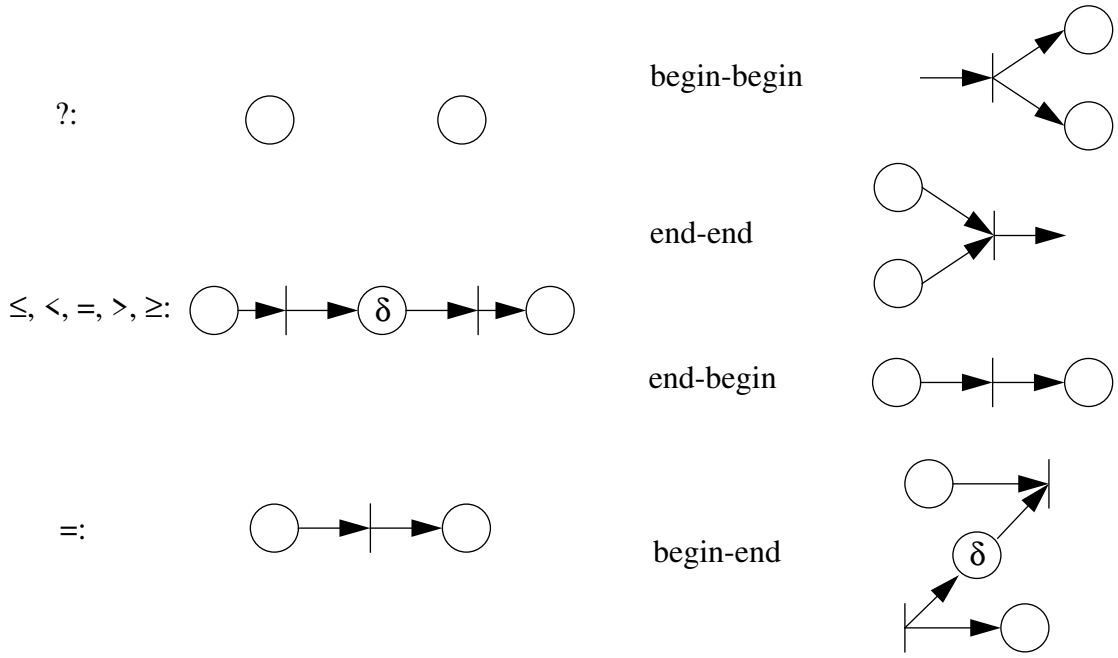


Figure 5: Petri nets

A timed petri net model is proposed by [LiGh90] and [Hoep91]. The petri net of [Hoep91] is a mapping of the path notation on petri nets and will be analyzed together with the path notation in Section 4.4. In this section, we essentially follow the petri net definition of [LiGh90]. There, intervals are represented by places and relations by transitions. In order to avoid ambiguities, we need the additional assumption that petri nets in this context are conflict-free. The basic units of the model are intervals. Therefore, this model is classified as interval-based although transitions refer only to end-points of intervals.

The relation '?' is specified if two places are not connected by any transition. As shown in Figure 5, <, =, > can be modelled by a transition in conjunction with a delay place δ. The delay place represents an idle time $\delta \subseteq \Re^+_0$. If δ is in $\Re^+$, the corresponding relations are < and >. The relation = is modelled if $\delta = \{0\}$. In this case, the place can be omitted as it is done in Figure 5. If δ is unrestricted in $\Re^+_0$, then ≤ or ≥ is expressed.

In petri nets, the PRs ≤, <, =, >, ≥ can be represented. Since Figure 5 assures that any combination of interval end-points can be connected by a relation, the petri net model is as powerful as the point net model. This means that 82 IRs can be expressed although [LiGh90] described only the 13 basic IRs.

4.4 Path expressions

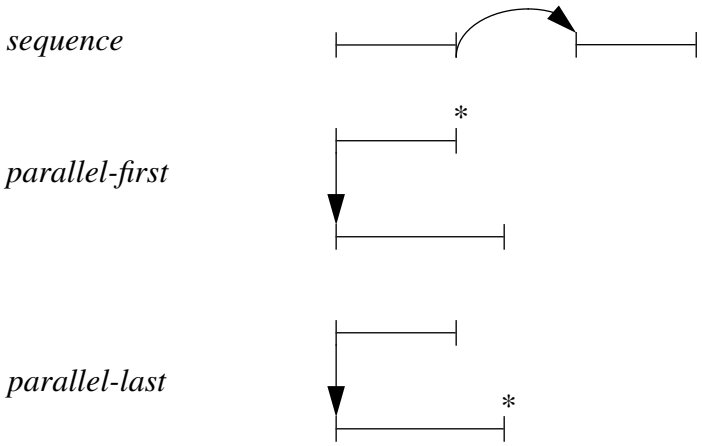


Figure 6: Path expressions

Path expressions were introduced by [CaHa74] for procedure level synchronization and adapted by [Hoep91] for multimedia presentation systems. Path expressions include three operators to represent temporal relations: *sequence*, *parallel-first* and *parallel-last*.

The basic units of path expressions are intervals. However, all three express only IRs that can be described by a single PR. The *sequence* operator models a relation between the end-point of the first and the beginning of the second interval. The IRs that can be expressed by the *sequence* operator are {m} and {mi}. Using a delay interval [Hoep91], it is also possible to represent {<} and {>}.

For this classification, the operators *parallel-first* and *parallel-last* are identical because the attributes first and last give reference points for subsequent operators, which do not have any impact on our relation analysis. The parallel operators establish a relation between the start-points of two intervals. Three indefinite IRs are expressible by the parallel operators: {s, =, si}, {di, o, fi, m, <} and {>, mi, oi, f, d}.

To summarize, path expressions are only able to represent 7 IRs: 4 basic IRs {m}, {mi}, {<}, {>} and 3 non-basic indefinite IRs {s, =, si}, {di, o, fi, m, <}, {>, mi, oi, f, d}.

4.5 MHEG

MHEG (Multimedia Hypermedia Expert Group) [MHEG92] [KrCa92] [Mark91] is a standardization group to establish a new standard for multimedia objects. MHEG uses a time model similarly to the path expression model. Additionally, MHEG allows not to specify any temporal relation between two intervals represented as multimedia objects. Therefore, MHEG has 8 possible IRs, one more than the original path expression model.

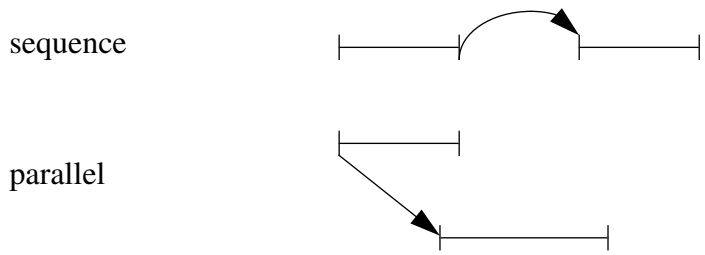


Figure 7: MHEG time model

4.6 Resume of the Evaluation

time model	type	number of interval relations		
		total	basic	representable by the PRs <, =, >, ?
time-line	point-based	13	13	13
point nets	point-based	82	13	29
petri nets	interval-based	82	13	29
path expressions	interval-based	7	4	7
MHEG	interval-based	8	4	8

Table 4: Summary: Multimedia time models

Table 4 summarizes the multimedia time models, their basic types and the corresponding IRs that can be represented. Assessing the temporal models, it is not only important how many relations are expressible in a specific model but also which relations are representable. As we showed in Section 3, not all relations are equally important. Specifically, the 29 IRs generated by the PRs <, =, > and ? are very important including the 13 basic IRs. So, Table 4 also shows how many basic and how many of the 29 relevant IRs can be expressed in each of the temporal models.

It can be observed that non of the examined temporal models exceeds the expressive power of the point-based framework, not even those models that operate on intervals. All temporal relations in the examined models can be denoted within the PR system ≤, <, =, >, ≥, ?. To provide the full expressive power of the interval-based framework for multimedia, we will develop an interval operator system in the next section.

The more relations a temporal model is able to represent, the more general it is and less prerequisites have to be met when it is applied. However in some multimedia environments, only a limited number of relations can occur. Then, only a simple temporal model is needed. So, when choosing a temporal model for multimedia, the context and the restrictions have to be respected. It seems that there is not a universal temporal model for all multimedia applications. There are simple models for simple environments and more universal models for complex environments.

The question of a most suited temporal model for multimedia became especially important since a standardized temporal model is needed for exchanging and storing multimedia information.

With the emerging standards HyTime and MHEG, it has been discussed which of their temporal models is more general. Concluding from our analysis, MHEG models less relations but has more flexibility due to the ?-relation, whereas HyTime using the time-line model has more possibilities. But neither MHEG nor HyTime is a superset of the other.

5. An Interval-Based Time Representation

Since all real presentation actions (video, audio, text, etc.) have a non-zero, finite duration, it seems to be natural to model multimedia actions as intervals. Also, point-based systems have some inherent disadvantages that are due to the limitations of the point-based time model. To overcome the disadvantages of point-based systems, we will systematically develop an interval-based model in this section.

5.1 Modelling Presentation Actions

Before developing this model, we have to introduce the notion of a presentation action. Any multimedia presentation is composed of single media items. The process of presenting a single media item is called a *presentation action*. Any action can be characterized by two significant end-points, the beginning and the ending, and the duration δ which describes the time is required when presenting a media item. The duration δ has a specific fixed value for any real presentation. However in the process of planning a presentation, the final duration might not be known. Therefore, the duration is described as a subset of the non-negative numbers $\Re^+_0$ [KeLo91] indicating the potential values of the duration. So, the duration can be a single real number, a range within the real numbers or totally unrestricted in $\Re^+_0$. E.g. the duration of a 90-minute video that might by interrupted by a user interaction is written as [0 min, 90 min] $\subseteq \Re^+_0$ because the real duration is 90 minutes or less depending on the user interaction. In the other case, the duration is denoted as [90 min, 90 min] = {90 min} $\subseteq \Re^+_0$ meaning the duration cannot be modified and has a fixed value.

A *delay* is a time span which passes without presenting any audio-visual output, and thus it is distinct from a presentation action with a perceivable output. On the other hand, the temporal characteristics are similar to those of presentation actions. So, a delay can be described as a subset of the non-negative real numbers $\Re^+_0$.

Note that, in this paper, it suffices to characterize a presentation action only by its temporal behavior. Other attributes including those specifying the location, the quality or associated media of a presentation are not subject of our investigation.

5.2 Primitive Interval-Based Models

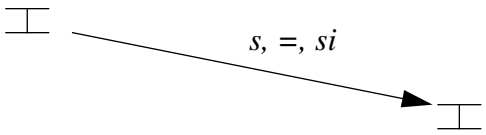


Figure 8: Expressing 'cobegin' by disjunction of basic IRs

For specifying temporal interrelations between media actions, two extreme approaches can be considered. In the first approach, disjunctions of the 13 basic IRs are used as a method to specify interval relations. E.g., a 'cobegin' of the presentation action can be denoted as a disjunction of 'starts', 'equals' and 'starts inverse' {s, =, si}. The obvious drawback of the approach is the high number of IRs required to represent a single PR: Up to 11 IRs are needed to represent a single PR [Rich89] (Table 5). Of course, this is not acceptable as a user interface because users need single and intuitive relations. Consequently, we require that at least the 29 IRs relevant to multimedia and generated by the PRs <, = and > should be represented by a single relation operator.

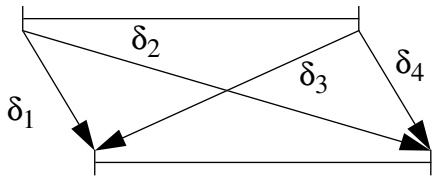


Figure 9: Totally generic operator: genericIR(δ_1, δ_2, δ_3, δ_4)

The other extreme is a model based on a totally generic operator. The operator can be derived from Figure 9 as: *genericIR(δ_1, δ_2, δ_3, δ_4)*, where δ_i, $i \in \{1,..,4\}$, is the delay for each of the possible end-point relations. The delay can be any subset of the real numbers. In this model, the delay may have negative values to indicate which of the corresponding time points is the first one. The trade-off of this approach is the huge number of inconsistent specifications that can be created by this operator. Moreover, consistency checking would be as expensive as in a point-based temporal model.

5.3 Enhanced Interval-Based Model

Though very flexible, both of the above models are not applicable as they do not represent temporal relations intuitively. Therefore, we define an alternative model by using the IRs generated by <, =, > and ? (Table 5). Constructing an operator for each of the relations, 29 operators are needed. This number seems to confuse the user of a presentation system. Fortunately, the number of operators can be reduced by exploiting regularities between the IRs. Then, several IRs can be combined to one operator.

Using the regularities, the number of operators can be reduced from the original 29 to 10. Figure 10 shows the generic pattern for each of the operators. Formal definitions can be derived from the patterns. For example, the operator *x before(δ_1) y* is defined by $Ex + \delta_1 = By$, i.e. the beginning of the interval y is δ_1 time units after the end of the interval x.

The first regularity is that some relations are inverse to each other. E.g., 'x meets y' is the inverse of 'y meets x'. So, we can use the operator *before(δ_1)* to specify both relations: *x before(0) y* for 'x meets y' and *x before^{-1}(0) y* for 'y meets x'. In graphical notations, the inverse is expressed by an inverted edge.

The second regularity is that some relations differ only by an offset from others. E.g., 'x meets y' and 'x < y' are only in so far distinct as there is a non-zero time span between x and y in the case of 'x < y' and a zero time span in the case of 'x meets y'. IRs that differ only in offsets are combined to the same operator. Then, the IRs can be distinguished by the delay pa-

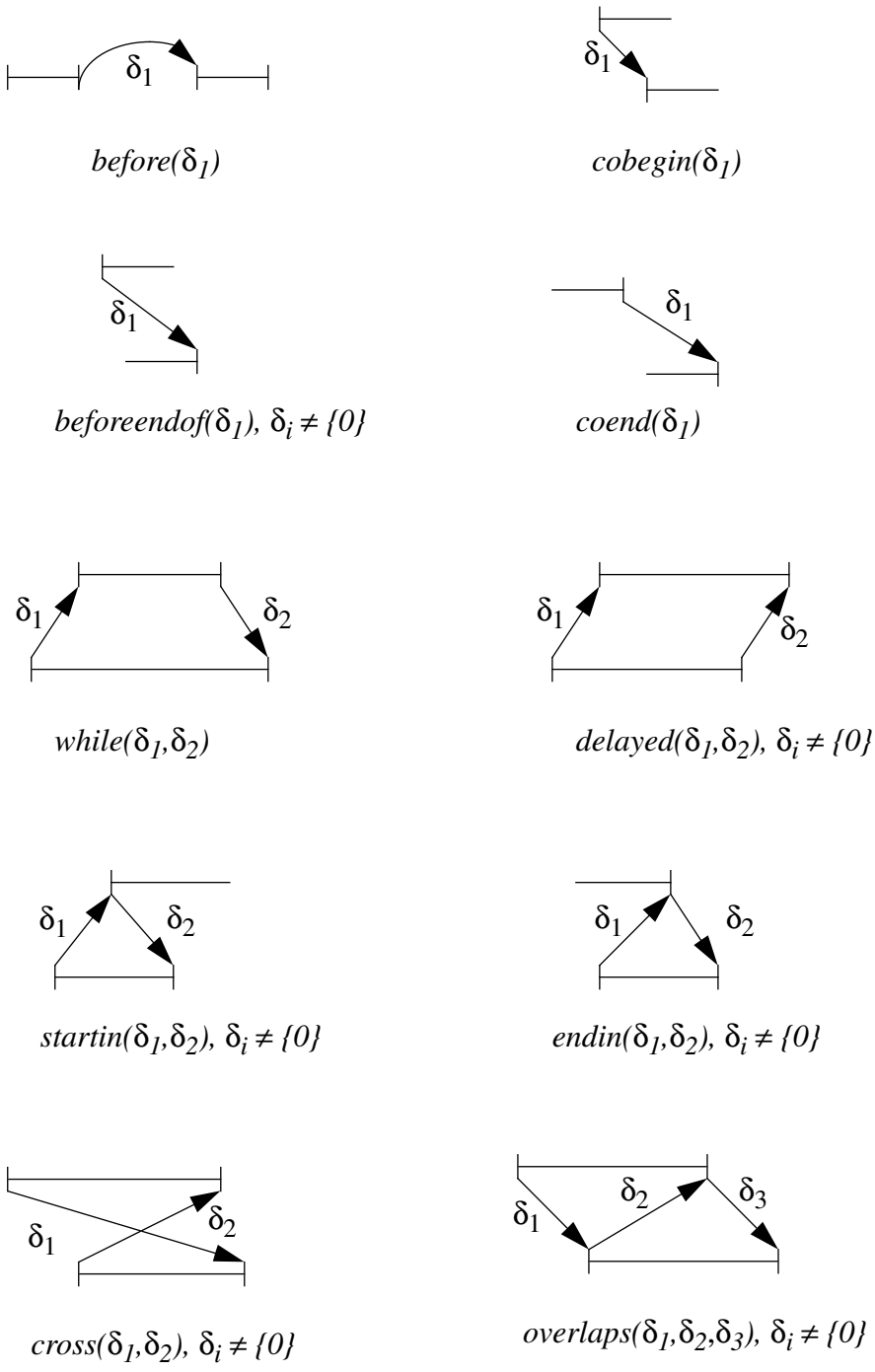


Figure 10: Basic IR patterns and their generic operators

rameter δ_1 of the operators. In the given example, we specify *x before(0) y* for 'x meets y' and *x before(+) y* for 'x < y'. As we introduced in 5.1, the delay parameter may be any subset of $\Re^+_0$. We use the notation '0' if the delay is zero, '+' if the delay has a positive value, and '*' if the delay is positive or zero.

To avoid having several specification methods for the same IR, we require $\delta_1 \neq \{0\}$ for some of operators in Figure 10. Then, the 10 operators are a complete set to specify any of the 29 IRs

generated by <, =, > and ?. An interval operator specification for each of the 29 IRs is given in Table 5.

The construction of the interval operators yields different types of operators taking 1, 2 or 3 delay parameters. The 1-parameter operators are *before*, *cobegin*, *beforeendof* and *coend*. Operators with 2 parameters are *while*, *delayed*, *startin*, *endin* and *cross*. Finally, *overlaps* is an operator that takes 3 parameters.

A delay or a duration parameter is fixed if only one value is admitted, e.g. a full length video that cannot be interrupted has a fixed duration of 90 min = [90 min, 90 min]. When specifying an interval relation, one has to specify the duration of the two presentation actions and up to 3 delay parameters. Hence, specifying 3 fixed values for the delay or the duration totally determines the final presentation sequence. Therefore, at most 3 fixed delay or duration parameters are allowed to avoid overconstraint specifications. E.g., if we specify the interval relation for two fixed length presentation actions, we can only use an interval operator taking 1 parameter. In the case of one fixed length action, we can use only 1- or 2-parameter operators. Only in the case that both actions have a variable length, we are allowed to use all operators. To elaborate the restriction, look at the following example. A user specifies fading from one video to a subsequent video.

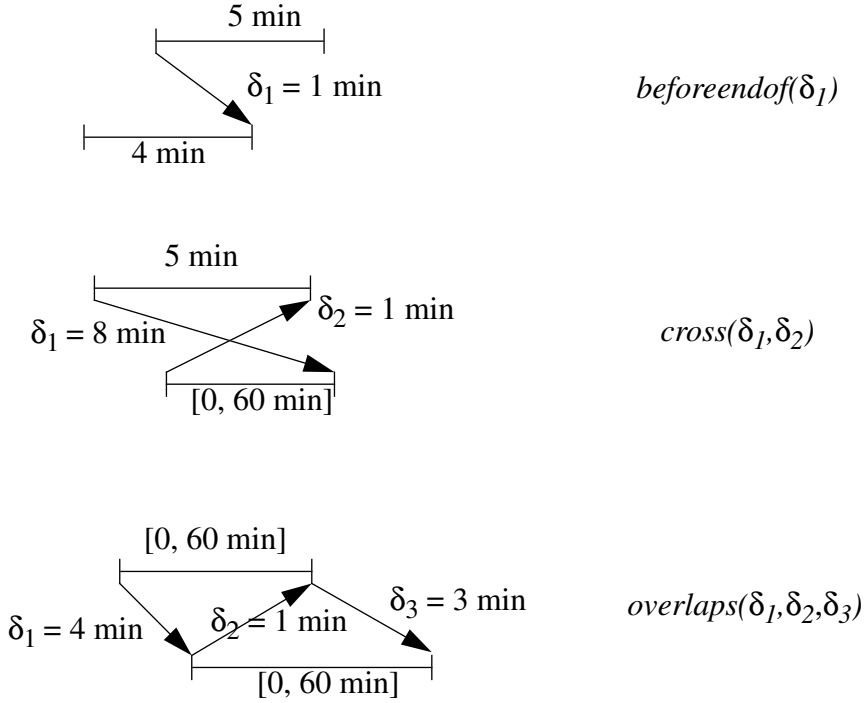


Figure 11: Specifying fading

If we have 2 videos and want to display the full natural length of both videos which have a specific fixed duration, we might specify *beforeendof*(δ_1) where δ_1 describes the time span during which both videos are displayed. The presentation planner will find a consistent scenario in any case. However, the videos do not overlap if the duration of one of the videos is shorter than the time span δ_1.

If the length of the 2 videos are variable, e.g. we need only parts of the videos for composing a video clip sequence, we might use the *overlap(δ_1, δ_2, δ_3)* operator to specify fading. Then δ_1 represents the time during which the first video is displayed but not the second, δ_2 is the time during which the both videos are active and δ_3 describes the postspan of the second video.

In case, one video has a fixed duration and the other is variable, the *cross(δ_1, δ_2)* operator is used. δ_1 indicates the total duration of the presentation and δ_2 the overlapping time.

The 10 interval operators are a complete set to represent the 29 relations generated by <, =, > and ?. But this does not imply that all operators are needed to define a complete temporal model for a multimedia environment. Sometimes, only a selection of the operators is necessary. E.g., if the duration of all media items is preknown and fixed, the temporal model may be restricted to the operators taking at most 1 delay parameter. Note that the requirements 'preknown and fixed duration' are very strict and prohibit any kind of interaction or flexibility. With the emerging interactive multimedia systems, it is expected that a larger subset of the interval operators is needed because interactive media items introduce a huge number of unpredictable durations.

5.4 Examples

We will look at three multimedia presentation scenarios to show the differences between the time-line, the point relation net and the interval operators.

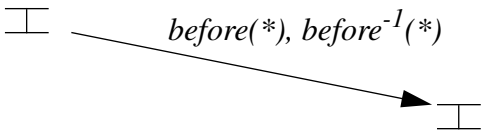


Figure 12: Expressing 'not parallel'

The first scenario shows how to specify the relation 'not parallel'. This requires to represent a disjunction of two operators (Figure 12). Note that 'not parallel' cannot be expressed by any of the temporal models in Section 4 because disjunctions cannot be specified in those models. In the interval operator system, we specify *'before(*), before^{-1}(*)'* meaning either *'x is before y'* or *'y is before x'*. Further, disjunctions of interval operators can be represented graphically (Figure 12). In point-based models, the graphical notation of this problem class is not equally transparent or not possible at all.

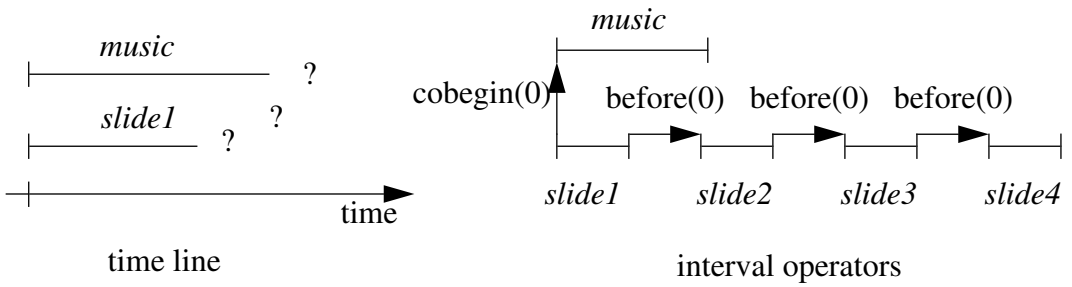


Figure 13: Slide show scenario

The second scenario starts with a simultaneous presentation of a slide and some background music. Then, the user can terminate the slide interactively and continue with the next slide. Also,

the user might stop the background music any time. Using interval operators, this scenario is specified easily (Figure 13).

In the time-line model, this scenario is not representable because the end-points of the slides and the music is determined interactively. This means that the end-points are not known ahead of time. However, we need the end-point of the previous slide to specify the beginning of the next slide. We would have to pick a point on the time-line although we do not know when this point in time will be. This specification problem of the time-line model is caused by its lack of flexibility, i.e. the time-line requires a total specification of all temporal relations between media items not admitting any indeterminism. Consequently, the time-line model is not appropriate for partial specifications or interactive media environments.

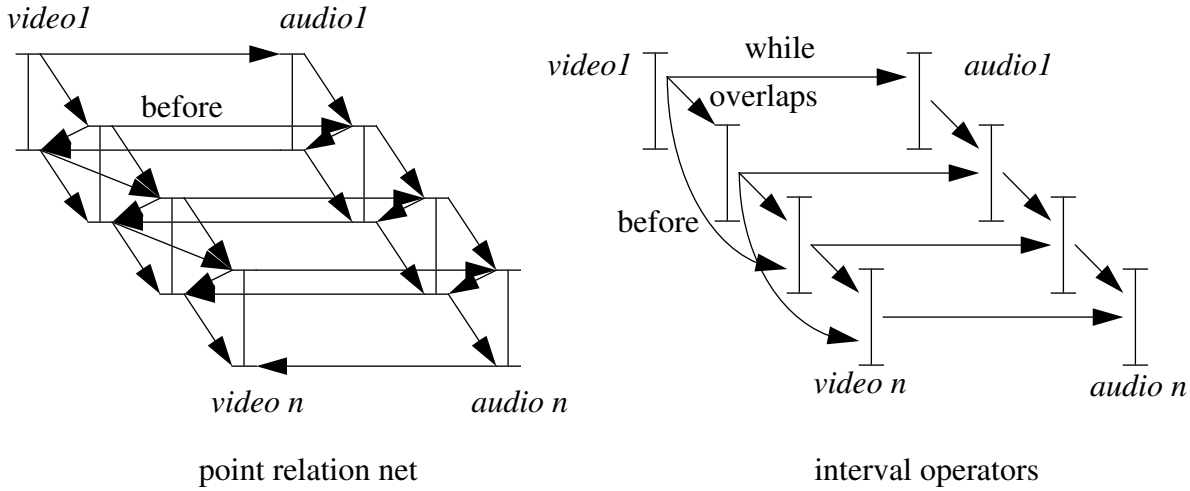


Figure 14: Video clip scenario

The third scenario is a video clip sequence. A short video-audio clip is followed by a subsequent video-audio clip, and the transition between the video-audios is done by fading. Moreover, not more than two videos should be active at the same time, e.g. the system does not allow fading between three videos at the same time. The specification of this complex scenario is done quickly and fairly intuitively by interval operators (Figure 13).

Using the point net representation, this scenario becomes quite complex because we need a huge number of point relations. Additionally, this scenario is hard to represent in a graphical notation. Point nets use only very basic relations resulting in a huge number of relations that have to be specified in complex scenarios. Interval operators have the advantage that they provide richer relations which allow the specification of complex presentations with a few powerful statements. Interval operators are more similar to natural languages which also use rich temporal relations such as ‘while’, ‘during’ and ‘overlapping’. For complex scenarios, the interval operators are more appropriate because, first, the operators represent high-level temporal relations, and secondly, the interval-based framework is more powerful.

6. Conclusion

In multimedia systems, synchronization is an important issue composed of the subtasks of representing temporal information and satisfying temporal constraints during the execution. This paper examined the representation of time for multimedia. After introducing the two temporal frameworks, point-based and interval-based, we showed that the point relations <, =, > and ? are needed in a multimedia environment. Analogically, the important relations in interval-based frameworks are the 29 IRs generated by the four PRs <, =, > and ?. Then, we determined the expressive power of existing approaches of time modelling in multimedia. It is observed that none of these models exceeds the relations set that is expressible in a point-based framework.

Learning from the shortcomings of the existing approaches, a set of interval-based operators were developed. Obviously, the interval operators represent *high-level expressions* of temporal relations. Since they were derived from the relevant 29 IRs, the interval operators cover the most essential set of interval relations. The proposed set can also be constructed from the PR set ≤, <, =, >, ≥, ?. Then, 82 IRs are representable by a single interval operator. Further, the interval operators are able to represent all 2^{13}-1 indefinite satisfiable IRs as disjunctions of operators as it was shown for the relation 'not parallel' in Figure 12. The expressive power of the interval operators cover the entire interval relation space which includes the expressive power of the point-based framework. Therefore, a huge number of temporal relations are representable by interval operators, i.e. the interval operators provide a *fine-grained model* of temporal relations. Moreover, the interval operators guarantee a high-level of *flexibility* because they were developed respecting the ?-relation which is responsible for the degree of flexibility. Finally, the interval-based framework reduces the number of possible inconsistencies. Looking at Figure 1, there are $3^4 = 81$ possibilities to specify a relation between two intervals using the PRs <, =, > and ?. But it is proved that only 29 of those represent a consistent scenario. The interval operators were developed such that the 62 inconsistent scenarios representable in point-based models cannot be specified by the interval operators. So, the interval-based operators significantly *simplify consistency checking*. This is important because extensive consistency checking may substantially affect the performance of a multimedia system. This is crucial as those systems are subject to real-time constraints. Since the interval operators provide a high level of flexibility, modelling interaction can be added easily. Studies of integrating an enhanced interaction model are in progress.

Annex A

Table 5 summarizes the 29 interval relations that are generated by the point relations <, =, > and ?. Each interval relation is represented as a conjunction of point relation (first column) or as a disjunction of basic interval relations (second column) or as an interval operator (last column).

point notation	interval notation	operator notation
<no relation>	>, di, oi, mi, si, fi, =, f, s, o, d, m, <	<no operator>
Bx<By	di, fi, m, o, <	cobegin(+)
Bx=By	si, =, s	cobegin(0)
Bx>By	>, oi, mi, f, d	$cobegin^{-1}(+)$
Bx<Ey	di, oi, si, fi, =, f, s, m, o, d, <	beforeendof(+)
Bx=Ey	mi	$before^{-1}(0)$
Bx>Ey	>	$before^{-1}(+)$
Ex<By	<	before(+)
Ex=By	m	before(0)
Ex>By	>, di, oi, mi, si, fi, =, f, s, o, d	$beforeendof^{-1}(+)$
Ex<Ey	s, m, o, d, <	coend(+)
Ex=Ey	fi, =, f	coend(0)
Ex>Ey	>, di, oi, mi, si	$coend^{-1}(+)$
Bx<Ey Bx>By	oi, f, d	$startin^{-1}(+,+)$
Ex>By Bx<Ey	di, oi, si, fi, =, f, s, o, d	cross(+,+)
Ex>By Bx<By	di, fi, o	startin(+,+)
Ex<Ey Ex>By	s, o, d	$endin^{-1}(+,+)$
Ex<Ey Bx>By	d	while(+,+)
Ex<Ey Bx=By	s	while(0,+)
Ex<Ey Bx<By	m, o, <	$delayed^{-1}(+,+)$
Ex=Ey Bx>By	f	while(+,0)
Ex=Ey Bx=By	=	while(0,0)
Ex=Ey Bx<By	fi	$while^{-1}(+,0)$
Ex>Ey Bx<Ey	di, oi, si	endin(+,+)
Ex>Ey Bx>By	>, oi, mi	delayed(+,+)
Ex>Ey Bx=By	si	$while^{-1}(0,+)$
Ex>Ey Bx<By	di	$while^{-1}(+,+)$
Ex>Ey Bx<Ey Bx>By	oi	$overlaps^{-1}(+,+,+)$
Ex<Ey Ex>By Bx<By	o	overlaps(+,+,+)

Table 5: The 29 IRs generated by the conjunctions of the basic PRs

References

[Alle83] J. F. Allen. Maintaining Knowledge about Temporal Intervals. *Comm. ACM*, 26(11):832–843, 11 1983.

[Appl91] Computer Inc. Apple. *QuickTime Developer's Guide*. Developer technical Publications, 1991.

[BDF+92] Ingo Barth, Gabriel Dermler, Franz Fabian, Kurt Rothermel, Frank Sembach, Robert Erfle, and Johannes Rueckert. Multimedia Document Handling - A Survey of Concepts and Methods. Technical report, IPVR - IBM, 9 1992.

[BDH+93] Ingo Barth, Gabriel Dermler, Tobias Helbig, Kurt Rothermel, Frank Sembach, and Thomas Wahl. CINEMA: a Configurable INtEgrated Multimedia Architecture. In *GI/ITG-Arbeitstreffen*, 2 1993.

[BHL91] G. Blakowski, J. Huebel, and U. Langrehr. Tools for Specifying and Executing Synchronised Multimedia Presentations. *2nd. Intl. Workshop on Network and Operating System Support for Digital Audio and Video, Heidelberg*, 11 1991.

[Bruc72] B.C. Bruce. A Model for Temporal Reference and its Applicants in a Question Answering Program. *Artificial Intelligence*, 3:1–12, 1972.

[BuZe92] M. Cecelia Buchanan and Polle T. Zellweger. Scheduling Multimedia Documents Using Temporal Constraints. *Proc. of the 3rd International Workshop on Network and Operating System Support for Digital Audio and Video, San Diego*, pages 223–235, 11 1992.

[BuZe93] M. Cecilia Buchanan and Polle T. Zellweger. Automatic Temporal Layout Mechanisms. In *1st ACM Intl. Conference on Multimedia*, pages 341 – 350, 8 1993.

[CaHa74] R. H. Campbell and A. N. Habermann. The Specification of Process Synchronisation by Path Expressions. In G. Goos and J. Hartmanis, editors, *Lecture notes in Computer Science No. 16, Operating Systems*, pages 89–102. Springer Verlag, 1974.

[Drap93] George D. Drapeau. Synchronization in the MAEstro Multimedia Authoring Environment. In *1st ACM Intl. Conference on Multimedia*, pages 331 – 340, 8 1993.

[Gibb91] Simon Gibbs. Composite Multimedia and Active Objects. *Proc. OOPSLA'91*, pages 97–112, 1991.

[Hoar78] C.A.R Hoare. Communicating Sequential Processes. *Communications of the ACM*, 21(8), 8 1978.

[Hoar85] C.A.R Hoare. *Communicating Sequential Processes*. Engelwood Cliffs NJ: Prentice-Hall, 1985.

[Hoep91] P. Hoepner. Synchronisation der Praesentation von Multimediaobjekten-Modell und Beispiele. In J. Encarnacao, editor, *Informatik-Fachberichte 293, Telekommunikation und multimediale Anwendungen der Informatik*, pages 455–464. GI, Springer-Verlag, 10 1991.

[HyTi92] HyTime. Information technology - Hypermedia/Time-based Structuring Language (HyTime). *ISO/IEC DIS 10744*, 8 1992.

[KeLo91] Somnuk Keretho and Rasiah Loganantharaj. Qualitative and Quantitative Time Interval Constraint Networks. *Proc. of ACM, San Antonio*, 1991.

[KrCa92] Francis Kretz and Francoise Calaitis. Standardizing Hypermedia Information Objects. *IEEE Communications Magazine*, 5 1992.

[LiGh90] T. D. C. Little and A. Ghafoor. Synchronisation and Storage Models for Multimedia Objects. *IEEE Journal on Selected Areas in Communications*, 8(3):413–427, 3 1990.

[LiKo92] T. D. C. Little and F. Koa. An Intermedia Skew Control System for Multimedia Data Presentation. In *3rd Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 121 – 132, 11 1992.

[Mark91] Brian D. Markey. Emerging Hypermedia Standards - Hypermedia marketplace Prepares for HyTime and MHEG. In *USENIX 91*. DEC Multimedia Engineering, 6 1991.

[MHEG92] ISO/IEC/WD MHEG. Information Technology - Coded Representation of Multimedia and Hypermedia Information Objects. Working Draft 5, ISO/IEC, 3 1992.

[Rich89] Michael M. Richter. *Prinzipien der Kuenstlichen Intelligenz*. B.G Teubner Stuttgart, 1989.

[RoDe92] K. Rothermel and G. Dermler. Synchronization in Joint-Viewing Environments. In *3rd Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video*, 1992.

[Stei90] Ralf Steinmetz. Synchronisation Properties in Multimedia Systems. *IEEE Journal on Selected Areas in Communications*, 8(3):401–412, 4 1990.

[Stei92] Ralf Steinmetz. Multimedia Synchronisation Techniques: Experiences Based on Different System Structures. In *Multimedia Communications*, Monterey, USA CA, 4 1992. 4th IEEE COMSOC International Workshop.

[vBee92] Peter van Beek. Reasoning about qualitative temporal information. *Artificial Intelligence*, 8(956):297 – 326, 12 1992.

[ViKa86] M. Vilain and H.A. Kautz. Constraint propagation algorithms for temporal reasoning. In *AAAI-86 Philadelphia, PA*, pages 132 – 144, 1986.