Chapter XX

A survey on authoring techniques for temporal scenarios of multimedia documents

Muriel Jourdan, Nabil Layaïda and Cécile Roisin
OPERA project, INRIA Rhône-Alpes
655 avenue de l'Europe,
38330 Montbonnot, France.
e-mail: [Muriel.Jourdan, Nabil.Layaida, Cecile.Roisin]@inrialpes.fr
http://opera.inrialpes.fr/OPERA/

Abstract: In this chapter, we present a survey on authoring techniques for the creation of temporal scenarios for multimedia documents. We classify existing techniques by confronting them against two kinds of requirements: expressive power and authoring capabilities. These two sets of requirements lead to grouping multimedia authoring systems into two classes: operational ones and constraint-based ones. Both approaches provide synchronization facilities allowing the construction of temporal scenarios. But, the in-depth analysis achieved in this chapter shows that they do not fulfill the same requirements. This analysis is illustrated by a document example that is specified using each presented system.

1. Introduction

In traditional text oriented document systems, the communication mode is characterized by the spatial nature of information layout and the eye's ability to actively browse parts of the display. The reader is active while the rendering itself is passive. This active-passive role is reversed in audio-video communications: information flows actively to a passive listener or viewer. Interactive multimedia documents combine both of these roles as they contain both spatial and temporal types of information. In addition, they allow the reader to interact with the document presentation. For example, hypermedia links can be used to navigate inside the same document and/or between different documents.

Multimedia documents combine in time and space different types of elements like video, audio, still-picture, text, synthesized image, ... Compared to classical documents, multimedia documents are characterized by inherent temporal dimension. Basic media objects, like video, have intrinsic duration. Furthermore, they can be temporally organized by the author which adds to the document a temporal structure called the **temporal scenario**.

Due to this temporal dimension, building an authoring tool is a challenging task because **Wysiwyg** paradigm, used for classical documents, is not relevant anymore: it is not possible to specify a dynamic behavior and to see immediately its result. Edition and presentation operations are carried out at different times and by different users: the first ones being performed by the authors, the second ones by the readers. So we must distinguish between the specification phase (or editing phase) of the temporal scenario and its presentation phase (or execution phase). An authoring system for multimedia documents must handle these two phases, because it is essential for an author to easily skip from the editing phase to the execution phase in order to gradually test and improve the document presentation. Within the past decade, numerous research papers (Cmifed [26], Firefly [3], HTSPN [23], HPAS [29], Interval Expressions [13], Isis [24], Madeus [9]), have presented various ways of specifying temporal scenarios, focusing on a particular understanding of temporal synchronization. The purpose of this paper is to classify and to discuss the relevance of these proposals. Compared with other surveys on multimedia synchronization [2], [7], [15], [19], [21] and [28], this study is characterized by the twofold step:

- We distinguish between two kinds of requirements: expressive power and authoring capabilities.

- We classify the various ways of specifying temporal scenarios into two broad classes: operational and constraint-based approaches.

This allows having a clearer framework for classifying and comparing existing authoring environments from the temporal dimension point of view.

Moreover, we clarify what becomes often confusing when dealing with temporal scenario specifications: the various semantics that are associated with the Allen's operators [1]. It is an important point, since it is very common in multimedia authoring that expressive power is considered as equivalent to the ability to express all the Allen's operators. In this chapter, we give more precise definitions of expressive power requirements for multimedia specifications. As a result, this allows to have a more rigorous basis for comparing the different approaches.

The chapter is organized in four sections. In the first one, we present a working example containing interesting challenges as far as both expressiveness and authoring capabilities needs are concerned. In the second section, we define and detail these two kinds of requirements. In the third section we define the two classes of approaches (operational and constraint based), we present some well-known authoring environments for each class and check if they fulfill the above requirements. The last section concludes this overview. In particular, like in [7], a table summarizes the results for the environments covered in this study.

2. An example of multimedia document

Before getting in the main purpose of the paper, we present a multimedia document example. In this example, we try to cover some relevant cases that can be required when composing multimedia documents (see section 3). Its specification is (partially) given through the different systems that we analyze in section 4.

"BestCom" is a communication company that answers a call of the International Football Organization for the design of a mascot. In order to provide an attractive and a complete response, BestCom has created a multimedia document to be presented to its client.

The scenario is organized in two parts: (1) a presentation of the company (called Company) and (2) a presentation of the mascot proposal (called Mascot). These two parts of the scenario can be read sequentially but hypermedia links allows the reader to jump at any time from one part of the document to the other.

The Company part should globally last about 3 minutes. It is composed of a sequence of three objects: an audio clip(History) which gives the history of the company, followed by a textual message displaying the name of the company on the screen (Name) and ends with a graphic listing its main achievements (PressBook). History audio lasts for approximately 1 minute, Name and PressBook must be each one displayed during 45 seconds at least. In addition to this first specification, the author added a 2 minutes movie that gives an overview of the company together with its geographical localization (called Geography). This movie must be started so that the mapping of the Name on the screen is synchronized with the period of the Geography movie when the company building appears (approximately 20 seconds from its beginning).

The Mascot part is mainly composed of a virtual animation of the proposal (Animated Proposal). This animation ends with a last picture of the mascot (Proposal). This last frame remains displayed on the screen during 30 seconds together with a balloon (Balloon) on the right of the mascot mouth, which contains its name. In addition, in order to see the mascot name faster, the reader is asked by an audio message (Message) to click on a button (Button) during the presentation of the Animated Proposal. When the reader clicks on the button, the audio message stops and the balloon appears at the top of the screen and moves until it reaches its final position (near the mouth of the mascot) exactly when the animation ends. Figure 1gives a possible execution of this document in which the reader of the document has interacted twice. At second 140, after the beginning of the document, he jumped to the Mascot part (cropping the presentation of the Company Press-book) and, at the 180th second, he activated the button to see the mascot Name faster. As a consequence, the name appears on a balloon at the top on the screen and moves during such a period of time that it stops exactly when the animated proposal ends.

We will show in section 4 how some parts of this example could be specified in various approaches (see Figure 5 to Figure 11).

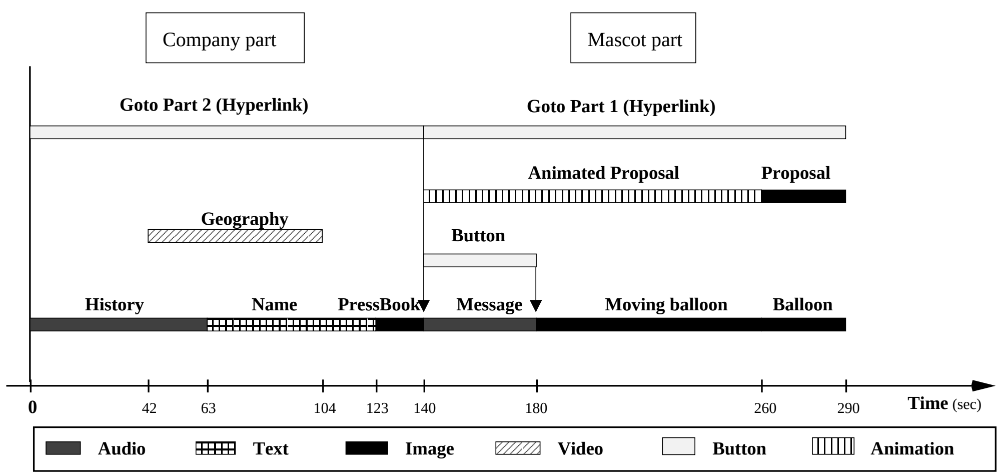


Figure 1. A possible execution for the Mascot presentation

3. Multimedia authoring requirements

The variety of multimedia approaches reflects the large number of requirements that have to be covered by a multimedia authoring system. But these needs are only partially fulfilled by existing applications. In order to give a structured and readable analysis, we only focus on authoring requirements: we group them in two main classes (expressive power and authoring capabilities).

3.1. Expressive power

The expressive power of an authoring system is somehow related to the ability of the system to cover a broad range of temporal scenarios required by the author. This criterion is hard to measure since defining an acceptable level of expressive power is strongly dependent on author practice and experience. Furthermore, we still have limited knowledge about the authoring process of time-based documents. So far, expressive power has been mainly considered from an informal point of view. In this chapter, we try to have a more formal approach.

- We consider a document presented by an authoring system as a state machine (see Figure 2) characterized by:
- A set of objects O.
- A set of inputs IN (clock tics and events on the objects).
- A set of outputs OUT (start, stop, ... on the objects).
- An execution loop defined as: getting some inputs, producing some outputs and updating some state variables.

This state machine can be modeled by a Mealy automata where each state is a node and each arc is labeled by an input vector and an output vector that reflect the state transitions representing the document behavior.

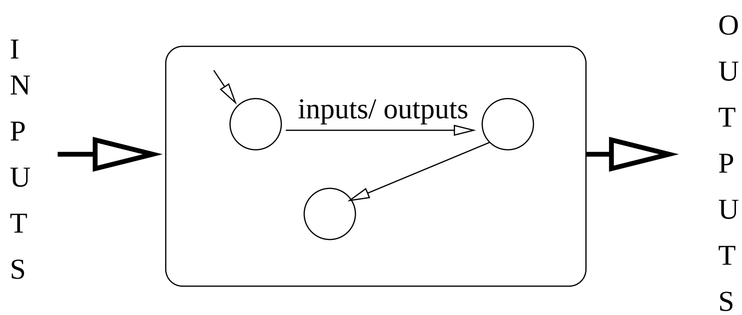


Figure 2. State machine of a document

With this description, the expressive power of a specification language can be evaluated by its ability to describe any state machine from a given set of objects and their related events. In the remaining part of this chapter, we will refine this model and use it to obtain a comparison criteria.

We have classified authoring requirements into three sets: (a) the needs arising from the intrinsic nature of the objects composing multimedia documents, (b) those arising from their composition and finally (c) those related to hypermedia navigation. Our analysis is independent from the underlying method used by the author to build these documents (programming language, constraint operators, tree structure, etc.).

(a) Multimedia objects

Authoring environments must provide the author with:

- **A wide variety of basic objects (issue 1):** Objects like text, video, audio, still pictures, virtual animations, programs or applets are good examples of what must be supported. These objects differ in nature, as they can be discrete: their content is delivered instantaneously such as text and still picture; or continuous: their content is delivered progressively such as video or audio.

  Therefore, the set O of objects is composed of DO (Discrete Objects) and CO (Continuous Objects).

A presentation system must allow to distinguish between mapping (and unmapping) of objects and execution or playtime of continuous objects. For example, a video can be mapped onto the screen (the first image is displayed) but its execution can be run at different instant. Therefore, presenting objects to a reader requires the following outputs of the state machine:

- map(o), unmap(o) for displayed objects,
- start(co), stop(co), pause(co) and resume(co) for continuous objects,

- **The possibility to control the delivery of continuous objects (issue 2):** The author must be able to express that the content of a continuous object is to be completely delivered (i.e. it will not be interrupted or cropped) even if its end has to be synchronized with other time points. For instance, in our example, the virtual animation (Animated Proposal) has to be delivered entirely, otherwise the spatial synchronization required between the Balloon and the Mascot picture cannot be met.

- **Interactivity capabilities (issue 3):** The author must be able to make any kind of objects activable (called buttons) during a given period of time with which the reader can interact during the presentation of the document.

  Therefore, objects are set (unset) activable by the two following outputs of the state machine: activable(o) and disactivable(o). A user activation on such an object o is defined by the click(o) input.

- **Temporal style definitions (issue 4):** motion effects or audio with variation of volume are examples of temporal styles. They can be associated at any temporal interval with any discrete or continuous object o by the two outputs of the state machine: styleon(o, s) and styleoff(o, s).

- **A support for unpredictable objects (issue 5):** Java applets or some videos are examples of such kind of objects. They are either continuous objects or objects transformed into buttons.

  - Continuous objects are unpredictable if their duration is not known beforehand or statically [3], because their effective duration at presentation time can be affected by external factors (resource limitations like machine load for instance). Therefore, it is impossible to assign them a static duration if their content must be completely delivered.

  - Buttons are by nature always unpredictable since their effective duration is defined by the interaction of the reader during the presentation phase (and may change at every execution).

The distinction between predictable an unpredictable objects does not only depend on the nature of the objects (video, audio) but also on their internal coding and the semantics of their content. For example, to recover a network delay, the duration of a video coded in MPEG can be controlled by dropping some of its frames [18]. Similarly the duration of a synthetic audio should be easier to control than a recorded speech. Thus, some objects can become predictable in authoring systems with powerful object access, manipulation and rendering techniques, while they can be considered as unpredictable in less advanced ones. For this reason we give the following definition:

*An object is **predictable** in a particular environment if and only if the presentation system guarantees its duration without semantic loss.*

Therefore, the set CO of continuous objects contains the set UCO of unpredictable continuous objects and the state machine can receive another kind of inputs: end(uco) which is the event related to the normal termination of an UCO object.

In our example, the three sensible areas (the two hyperlinks and the Button) are unpredictable. Moreover, audio and video are considered as unpredictable objects while the Animated Proposal is considered as predictable since it is a synthesized animation (in animated gif).

(b) Temporal Composition (issue 6)

As far as expressive power is concerned, temporal composition aims at expressing any arbitrary ordering between temporal intervals corresponding to the different objects. In terms of the state machine model, this is equivalent to the ability to describe any significant state machines for a given set of objects and their related events as identified above. A state machine is said to be not significant when it contains some executions (sequences of transitions from an initial state) which are inconsistent. For instance, a sequence in which an object is stopped before it begins running is irrelevant. Taking into account the different requirements listed above, the input and output sets of the state machine model of a document can be more precisely defined:

- IN = {tic, click(o), end(uco)}, see issues 3 and 5.
- OUT = {map(o), unmap(o), start(co), stop(co), pause(co), resume(co), styleon(o,s), styleoff(o,s), activable(o), disactivable(o)}, see issues 1, 3 and 4.

Let O, IN, OUT be the set of objects, inputs and outputs; MS(O, IN, OUT) be the set of significant state machines that can be defined upon O, IN and OUT by a specification language S. The comparison of the expressive power between two specification languages S1 and S2 can be done by comparing the sets of objects (O1, O2), of inputs (IN1, IN2), of outputs (OUT1, OUT2) and the state machines (MS1(O1, IN1, OUT1), MS2(O2, IN2, OUT2)) of the languages.

(c) Interactions (issue 7)

Two kinds of interactions can be distinguished:

- Temporal Access Control (TAC) such as pause, resume, fast-forward, etc ... which provide the reader with a way to control the document rendering. TAC does not depend on the document and therefore, it does not appear in the specification of the temporal scenario.

- Document interactions through buttons (see issue 3) whose semantics are specified in the document specification.

Document interactions can be classified in two classes according to their associated semantics. If the interaction involves all the active objects of the document at the presentation time, it is named a ***global interaction****; if only a subpart of this set is concerned by the interaction, it is* ***a local interaction****. It is obvious that both kinds of interaction appear in a multimedia specification as:*

- A usual hypertext link (allowing navigation facilities) is a global interaction since it interrupts or freezes the execution of all the active objects of the document. This results in starting other objects in another part of either the same document or another document. The hypermedia links between the two parts of the example are global interactions.

- On the contrary, a local button involving only a sub-part of a scenario is a local interaction: the button of the Mascot part interrupts the audio message and starts the balloon movement while the Animated Proposal is not affected.

A global interaction can be seen as a special case of local interaction. But we prefer to consider it separate, as their management is different (local ones are much more difficult to handle for synchronization reasons) and most of the existing systems do not provide local interactions but only global ones.

3.2 Authoring capabilities

At this point, the relevant question is how long does it take for an author to design a scenario? We retained seven criteria to measure the efficiency of a given approach regarding the issue of authoring capabilities. These are:

- **Adaptability to computer illiterate people (issue 8):** The idea is to evaluate how a system can be used efficiently by a large community of authors. In particular, those having no particular skills in computer programming.

- **Straightforward design (issue 9):** The author has a temporal organization of objects in his mind which is mainly expressed in terms of relative temporal placements between objects, i.e. temporal information (duration or ending/beginning instants) is given by reference to objects[1]. An authoring system must allow the user to specify in any order the temporal relations.

  For instance, in the Company part (Figure 1), the author would first specify a sequence between three objects (History, Name and Pressbook) and then a synchronization of 20 s of shift between the beginning of Name and the beginning of the Geography Video. The ease of translating these placements in terms of a given authoring approach is the whole question.

- **Indeterministic scenario authoring capabilities (issue 10):** A scenario can specify multiple presentations of the same document due to the presence of unpredictable objects. Thus the authoring system must help the author to get a global perception of his document, since he cannot manually explore all the possible solutions. This can be done by visualization tools and by static checking techniques which can inform the author about some global properties of the scenario, as for instance the mutual exclusion of two audio.

- **Adaptability to the incremental nature of the editing process:** Building an interactive multimedia document is a cyclic "specify, test and modify" process: one never reaches the right temporal layout at the first stage. Two requirements follow this observation:

  1. **Ease of local modifications (issue 11):** It must be easy for the author to make a local change in the specification. In particular, the authoring system must undertake

[1] By contrast with an absolute placement which supposes that temporal information of an object is defined without taking into account the other objects.

the global consequences of a local change into the document specification: both from the structural and temporal point of view. This feature depends heavily on the used authoring approach and therefore it will be illustrated later for each presented method.

2. **Fast editing/presentation cycle (issue 12):** It should be fast to switch between the specification phase of a document and its presentation.

- **Abstraction capabilities (issue 13):** A multimedia authoring system must help the author to compose large documents by providing the means to abstract and reuse parts of documents.

- **Multimedia document models (issue 14):** Generic models such as SGML [12] or XML [30] of textual documents have improved documents manipulation technologies. Similarly, the ability to define classes of multimedia documents will ease the author's task and will enhance multimedia environments by providing automatic document processing,

- **Multigrids reading support (issue 15):** A multimedia document should be understandable by different categories of readers. Categories can be defined by the native language of the readers or the level of the students in a course. An authoring tool must help the author while designing such kind of documents by allowing the share of common parts which can be objects as well as temporal scenarios.

4. State of the art through two classes of approaches

In this paper, we distinguish between two classes of authoring approaches: operational and constraint-based. They have been defined depending on how close the document description is to the presentation level. Unlike other classifications, we do not consider imperative against declarative approaches that allow only separating programming or script-based paradigms from other paradigms. Instead, we have chosen to talk about constraint-based and operational approaches since they allow a better identification of common properties with respect to the requirements previously mentioned.

The two classes of approaches are defined as follows:

- Operational approaches are based on the direct specification of the state machine, which defines the temporal scenario of the document (see section 1.1). The author specifies how a scenario must be executed: based on either a script language or an operational structure (tree or Petri-nets are good examples). Therefore the presentation phase directly implements the operational semantics provided by the used structure[2].

- Constraint-based approaches set the specification outside this operational scheme (see Figure 3). They are based on constraint programming and are characterized by a formatting phase that computes starting times and durations, as required by the scenario. This formatting phase can be seen as a compilation of a declarative specification into an operational structure, which can be interpreted by the presentation phase. Thus, the

[2] Some operational approaches (like tree structures) have also this declarative property but their formalism is however closer to the operational structure than to the constraint-based one

author specifies what scenario he needs without involvement of how to get the result in terms of operational actions, in a declarative way.

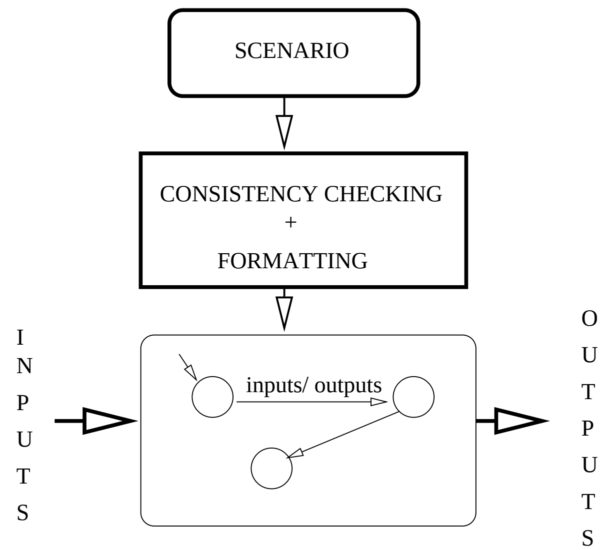


Figure 3. The state machine for the constraint-based approach

The end of the section is devoted to the presentation of main multimedia authoring systems in the light of the requirements and classification previously stated.

4.1. Operational Approaches

We present operational approaches through three classes: absolute temporal axis, script languages and graphical structures. The authoring process of these approaches is mainly characterized by the fact that authors have to usually give more temporal information as necessary.

Let's take the following example: the author wants to synchronize (both their beginning and ending) a video with a sequence of one still picture and another video. Assume that the durations of the two videos are predictable. With an operational approach (other than scripts), the author has two solutions: either he defines durations of the three objects; or he only defines the duration of the still picture and specifies that the shorter video interrupts the longer one. In the first solution, the author must assign duration values and as a result we obtain a scenario that is not easy to modify. In the second solution, the end of one video is not delivered entirely to the reader. This abnormal ending of one video may not be satisfying for the author.

4.1.1. Absolute temporal axis

The most intuitive way to specify a temporal specification is to place objects on a temporal axis. Doing so, the author gives absolute values for the beginning instant and the duration of each object. In such approaches, the corresponding state machine only has tics as inputs (see Figure 4): it does not take into account neither unpredictable objects (issue 5) nor buttons (issue 3).

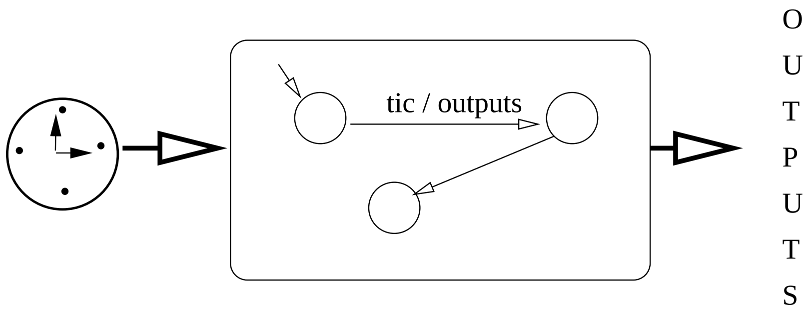


Figure 4. Input of the state machine associated with a absolute temporal axis

Authoring capabilities of such approaches are poor: issue 8 is met, thanks to the simplicity of the metaphor, while the others are far from being fulfilled. For instance, the author has to translate each relative placement into absolute ones (issue 9) preventing him from any easy modifications (issue 11).

In fact, such a paradigm is always used together with another kind of approach. For instance, Macromedia Director [20] uses both an absolute temporal axis and a script language.

4.1.2. Script languages

The most widespread approaches for specifying multimedia documents are based on the programming paradigm Lingo [20], IconAuthor [11], MHEG [22]. It is obvious that these approaches are the most expressive as they are capable of implementing scenarios with any arbitrary complexity. In MHEG for example, a temporal behavior can be associated to any composite object by means of a “link object” composed of a set of “event & conditions -> actions” statements. Thus, as far as the expressive power requirements are concerned, these approaches are very satisfactory (issues 1 to 7), even if most existing languages do not fit all of them. As an example, the Company part is given in Figure 5 in terms of the inputs (end(o)) and the outputs (start, map, unmap) of the state machine model previously defined.

```
Start audio "History.au";
Wait (40);
Start video "Geography.mpeg" ;
When end(History)
    Map text "Name.doc";
Wait (45);
Unmap text "Name.doc" ;
Map picture "PressBook.pic" ;
Wait (45);
Unmap picture "PressBook" ;
```

Figure 5. The Company part scenario in script language

The weak side of these approaches is their poor authoring capabilities. Except for the abstraction and multi-grids abilities provided with the object-oriented design and macro facilities in MHEG (issues 13 and 15), they cannot perfectly meet the authoring requirements previously listed. First of all, authors are assumed to have programming skills (issue 8). Moreover, it is difficult to achieve a temporal placement by using a script language and to modify an existing script in order to get a new temporal organization (issues 9 and 11).

In some languages like MHEG, the temporal composition is spread among the objects behavior programs, preventing the author from getting a global view of his scenario.

4.1.3. Graphical Structures

In order to provide higher level specification interfaces and to give a better perception of the structure of the temporal composition, some tools propose to use graphical structures such as trees and Petri-nets in order to specify the organization of the document.

As far as expressive power is concerned, the common problem of these approaches is that some scenarios cannot be expressed due to the necessity of handling variables to dynamically compute some durations (combination of issues 5 and 6). This is the case with the Mascot part of our example: we need a variable to compute the duration of the balloon movement (or its speed), which depends on the activation instant of the button, in order to respect the termination constraint (Balloon near the Mascot mouth). Indeed, the specification is not met if the speed of that movement is statically computed: the resulting execution would interrupt either the balloon movement or the animation (depending on which one terminates the first).

The other common limit is the restriction of issue 11 and more precisely about the partial support of automatic adjustment of durations during the editing process. Indeed, it is the author’s responsibility and not the authoring tool to compute object durations values. These values have to be recomputed by the author when the scenario is changed. This situation will be detailed below for each approach.

Tree structures

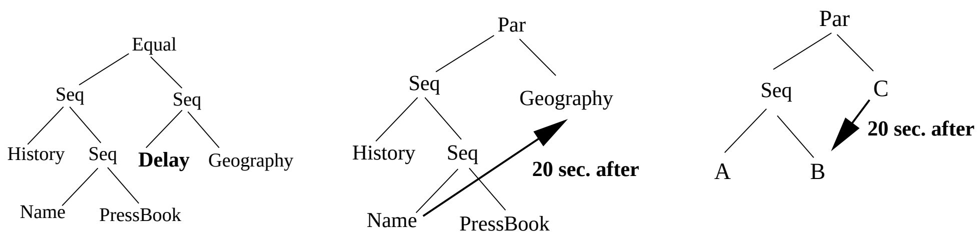
Tree structures are well known in the area of structured documents (like in SGML standard [12]) to express hierarchical decomposition and they have numerous algorithms associated to them. In the multimedia document context, the tree structure may be used to represent temporal composition: each node is associated with a temporal operator and each leaf represents a basic object.

The set of temporal operators depends on the system: CMIFed [26] proposes both sequential and parallel operators. The semantics of the parallel operator is to start simultaneously its operands without any constraint on the operand termination; SRT [15] has the sequential and the equal operators (the operands start simultaneously and must have the same duration). Let us note that in SRT a static checking phase is used to verify that the operands of an equal operator have the same duration. As a consequence, it is not possible to integrate in the SRT formalism neither unpredictable objects nor local interactions issues. Interval Expressions [13] and SMIL [31] have a more advanced set of operators to express interruptions like Par_min. This last operator is defined as follows: A Par_min B expresses that the two operands start together and the shortest one stops the other.

As far as authoring requirements are concerned, one important limitation of these approaches involves the straightforward and structural edition criterion (issue 9) which is illustrated by the Company part of the working example. The only way to get a tree-structure of this scenario is to fix the delay between the beginning of History and that of Geography, see Figure 6(a). Thus, the temporal information between Geography and Name is lost and modifying the duration of History implies reconsidering the specification (issue 11).

In order to cope with this problem, CMIFed introduces the notion of Synchronization Arcs, which breaks the tree structure by allowing additional temporal information between any pair of nodes of the tree. In our example, a synchronization Arc is set between Geography and Name which expresses the delay (20 seconds) between their beginnings, see Figure 6(b). The semantics of such synchronization arcs is unclear, as far as there is a potential conflict with the tree structure. For instance, considering the specification given in Figure 6(c): if A is longer than 20 seconds, what is the resulting scenario execution? Moreover, it is possible to generate deadlocks by using such arcs, but nothing is said about their detection.

Synchronization Arcs are also used to express local and global interactions (issue 7). These two kinds of interactions have been integrated into a uniform model: the Amsterdam Hypertext Model (AHM) [6]. As far as multigrids reading support (issue 15) is concerned, the channel view of the CMIFed environment provides the author with a way to share temporal scenarios among multiple versions of documents.



(a) The company Part with SRT **(b) The Company part with CMIfed** **(c) Arc synchronization**

Figure 6. Company scenario with SRT and CMIFed

Petri-net Structure

Petri-nets are well known in the area of parallel computing to express temporal synchronization and to perform static checking of properties such as safety and liveness. In the context of multimedia authoring, it has been used both in OCPN [19] and HTSPN [23] to model temporal scenario. Objects are modeled by places, temporal information is either associated with places (duration in OCPN) or with arcs (validity interval in HTSPN) and transitions are labeled with temporal operators (sequentiality and equality in OCPN and a richer set of possibilities, with Par_min for instance, in HTSPN). HTSPN provides the author with a way of abstracting some behaviors by a hierarchical organization through abstract places representing sub-networks (issue 13). They also provide the reader with global and local interactions (issue 7).

Petri-nets are more appropriate than tree approaches to capture the temporal structure of the scenarios (issue 9): the Company part of our example can be expressed in OCPN as shown in Figure 7. The synchronization between Name and Geography is not lost (Delay2 in Figure 7). Nevertheless, if the author changes the duration of History from 60 s to 30 s, other durations must be manually updated (Delay1 in the figure).

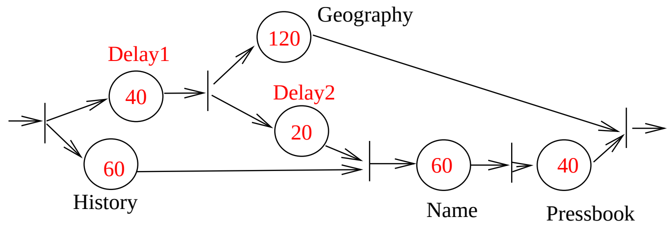


Figure 7. Company scenario using Timed Petri-Nets

In fact, OCPN is close to SRT: the equal operator implies a static checking of operands duration and both systems do not handle unpredictable objects (issue 5).

In addition to the previous remark about the need of explicit delay definitions, Petri-nets miss author's skills and straightforward design requirements (issues 8 and 9): translating a scenario into a graph structure of places and arcs is definitively not an end-user activity despite the graphical nature of Petri-nets.

4.2. Constraint-based Approaches

The starting point of all these approaches is the constraint theory. The idea is that the author declares a set of relations between either instants (=, >, <) or intervals (thirteen relations of Allen's algebra [1]). Each multimedia object can also be associated with a minimum, a maximum and an optimal duration, either chosen by the author or automatically fixed by the authoring system depending on the nature of the object (text, image. video, etc.). This is the Elastic time model first introduced in TemporalGlue [5] and Firefly [3]. For instance, the Company part of our example can be described by the Allen's relations given in Figure 8. The DelayGlob object is a fictive object (without content) which fixes the global duration of the Company part.

Set of objects with their possible durations

History	[55,65]
Name	[45, 80]
Pressbook	[45,80]
Delay2	[20,20]
Geography	[115,125]
DelayGlob	[170,190]

Set of constraints

History meets Name
Name meets Pressbook
Delay2 starts Geography
Delay2 meets Name
History starts DelayGlob
Pressbook finishes DelayGlob

Figure 8. Company part described with Allen's relations

The main point of constraint-based approaches is that the aim of the temporal formatter (see Figure 3) is firstly to check the temporal consistency of this set (there are no contradictory requests) and secondly to compute one solution, possibly an optimal one, which satisfies all the relations (see Figure 3). Algorithms used for these two phases are issued from either the temporal constraint satisfaction area [4] or linear programming [8].

These approaches are very interesting as far as authoring capabilities are concerned. Indeed, it is an intuitive way of expression (issue 8) which can easily capture the temporal information of multimedia documents (issue 6) by using the thirteen relations of Allen’s algebra. Modifications of the scenario are made easier since the author has not to reconsider the whole structure of his document but only to add or to delete a relation. For instance, if the author decides to complete the Company part by adding a Music after the Geography video in such a way that the Music ends simultaneously with the end of the PressBook, he just has to add the two following relations: Geography meets Music; Music finishes PressBook. Moreover, when he modifies an object, he has not to change the rest of the specification (issue 11). In the same part of the example, the author can replace the History object by a shorter one without modifying any other objects duration. This feature is very interesting to cope with the incremental nature of the editing process and also when the author wants to reuse his document in another context, as for instance to translate it into another language. The counter part of these advantages is the necessity to provide a formatting phase that produces an executable form of the scenario. One challenge is to make such systems with good time performances. Another one is to handle local interactions and unpredictable objects (issues 5 and 7). Therefore, temporal consistency checking phase must take into account uncertainty and thus, the formatting phase becomes partially dynamic.

More precisely, it should be possible to express any relative placement between intervals [1] together with interruption operators (issue 6). In order to take into account issue 2 (controlling the delivery of continuous objects), it is necessary to distinguish two cases of the equal relation: either the termination of one of the objects causes the termination or cropping of the other one, or the duration of the two objects are constrained to start and end at the same time, provided that their content is not cropped.

As a matter of example, let's take two objects A and B that are respectively a video and an audio. Consider the two following specifications:

- A and B start together and the shortest interrupts the longest one.
- A and B start together and have the same duration.

The difference between the two scenarios is that in the first case, the content of one object is not fully delivered to the reader of the document, while the second case expresses that both objects deliver their whole content. We distinguish these two temporal compositions as Par_min for the first specification and equal for the second one. A large number of misunderstandings in the field of temporal synchronization comes from the unclear distinction between these two behaviors.

In the rest of this section, we present the main features of three systems belonging to this category: Isis, Firefly and Madeus. Let us note that another more recent work, namely HPAS [29], proposes also a constraint-based multimedia editing system which is close to these ones.

4.2.1. Isis

In Isis [16], the set of relations between objects contains the four basic relations of the Allen's Algebra (meets, equals, starts, finishes). The other ones can be built from this set by introducing appropriate delays. All the objects durations are considered as predictable ones (issue 5 is not taken into account). A great effort has been put in order to compute the optimal solution taking into account fairness considerations (fair time dispatching among objects). Algorithms used for this computation are issued from linear programming.

In order to get better time performances, Isis designers have studied how to benefit from incremental methods [25]. This brings them to use temporal constraint networks (Dechter's algorithm [4]) and to adapt the shortest path algorithm of Dijkstra. One of their other challenges is to help the author when a temporal inconsistency is found in a set of relations (issue 10).

We are not aware of Isis experiments in introducing unpredictable duration in their framework. Moreover they do not provide the author with a Par_min relation. As a consequence, it is not possible to describe the Mascot part of our example. Isis provides the author with an interactive graphical interface, which uses a graphical syntax equivalent to the set of relations. Figure 9 shows the view associated with the Company part. This graphical view of the temporal scenario presents flexibility of objects by means of a spring metaphor (issues 8 and 10).

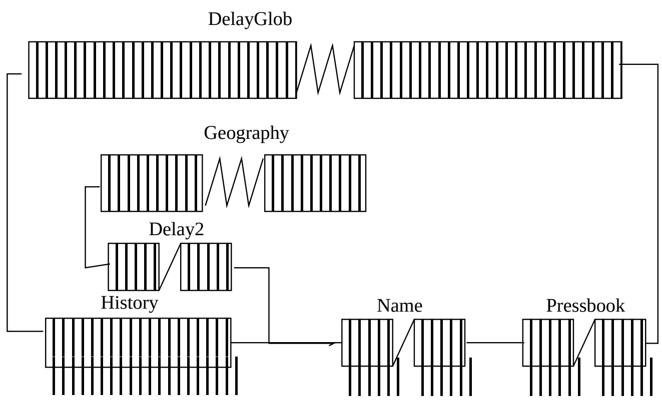


Figure 9. Company scenario with Isis

In order to offer abstraction functions (issue 13), they provide the user with a structuring capabilities in terms of a timed Petri-net in which each place is a component described by a set of relations and each arc models a global user-interaction [24]. From the authors' point of view, the use of Petri-nets seems to be in contradiction with the authoring advantages (mainly issues 8 and 9) gained by the constraint-based approach.

4.2.2. Firefly

In Firefly [3], temporal composition is expressed in terms of relations between instants (beginning and ending of objects and user interactions). The equal relation between instants is not oriented. So, if the author wants to start and to end two objects simultaneously, he cannot decide whether the ending synchronization is due to an interruption or not (distinction between equal and Par_min). In fact, the two semantics are supported but the Firefly scheduler makes this choice: unpredictable objects imply a Par_min composition.

This is the first work in the constraint-based area that has considered the unpredictable nature of some multimedia objects (like user interactions) (issue 5). In order to provide a static formatting process handling this kind of objects, the Firefly scheduler partitions the temporal scenario at compile-time by grouping connected components (i. e. instants related by a temporal relation or a predictable duration). Figure 10 shows the Firefly description of the Mascot scenario: there are two sets of connected components and two unpredictable objects (Button and Message) which appear with a dotted line.

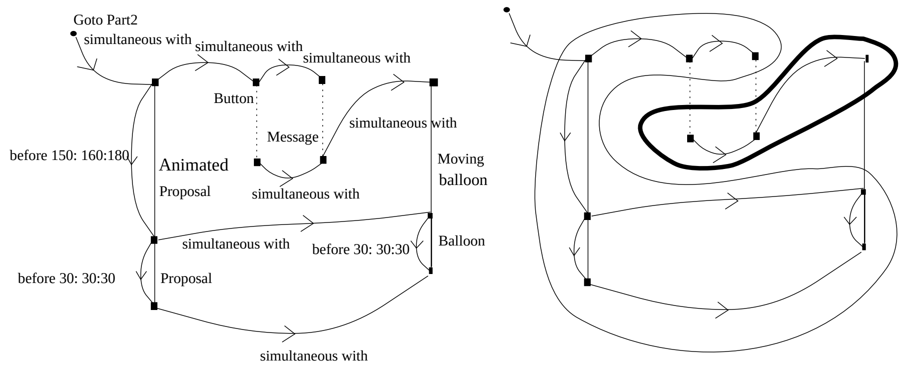


Figure 10. Mascot scenario with Firefly

The simplex algorithm is used to independently find the optimal solution of each partition. An event-driven scheduler dynamically handles the integration of the partitions. Unfortunately, the time performances of this batch process is not good-enough [3] in a interactive and incremental context (issue 12). This is the most important drawback of the Firefly approach.

Another weak point of Firefly is that it does not provide the user with any abstraction capabilities (issue 13).

4.2.3 Madeus

The set of temporal relations in Madeus is composed of the quantified Allen operators together with two additional interruption operators: Par_min and Par_master (only the first operand called master can interrupt the other one). Object duration can be predictable or not (issue 5). Abstraction capabilities (issue 13) are provided by hierarchical decomposition of the scenario which gives a framework to set temporal relations: two objects can be related if and only if they have the same parent in the tree structure. This located placement of the relations usually fits well with author's needs and does not show the disadvantage of the approaches which set a temporal operator on each node of the tree structure (see section 3.1.3). However, it may induce the same drawback of tree structures when the author wants to apply some modifications that break down the current structure of the document (issue 11).

The hierarchical structure of our example is very simple (left part of Figure 11). The root is composed of the two parts (Company and Mascot), the two hyperlinks (GotoPart1 and GotoPart2) and the delay DelayGlob. The relations associated with the root (resp. Company part and Mascot part) are given in the right part of Figure 11.

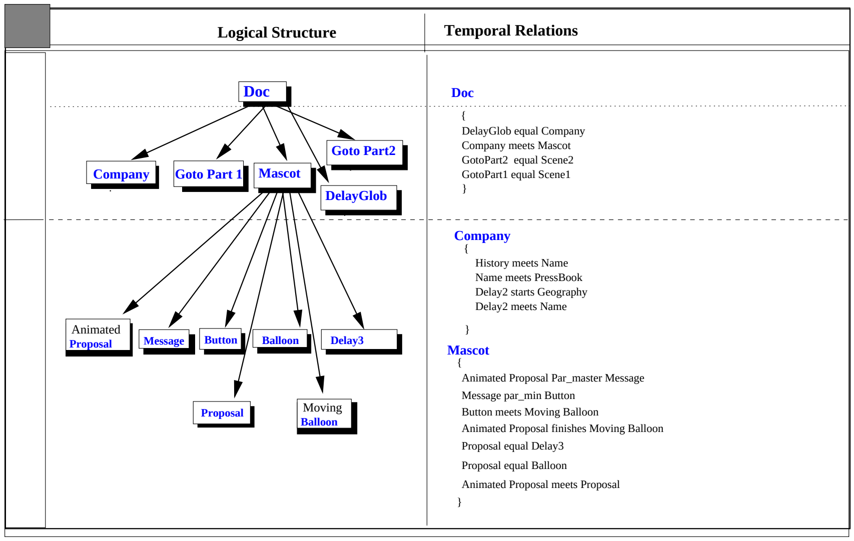


Figure 11. The complete example with Madeus

Madeus uses temporal constraint networks algorithms to incrementally detect temporal inconsistencies (issues 11 and 12). Difficulties arise when integrating unpredictable durations and interruption operators in such algorithms [17]. As far as we know, the only theoretical work which deals with the integration of interruption in the temporal constraint networks context is [27].

Madeus is a real running application [9] which partially implements the formatting phase, providing the user with easy and rapid switches between the specification phase and the presentation phase (issue 12). Currently, the editing phase is mainly based on an integrated textual editor, although it is possible to modify the temporal scenario by direct manipulation on objects and by the use of a palette of temporal operators (issue 8). However the design of a valuable interface for a constraint-based authoring system is more complex since, as stated in issue 10, the difficulty is to provide the author with a global perception of the set of solutions [10]. Algorithms to manage unpredictable objects are being studied and currently they bring partial results.

5. Synthesis

In this chapter, we have analyzed how different approaches for the specification of multimedia documents can meet the requirements for expressive power and authoring capabilities.

This work is summarized in the table of Figure 12 where we tried to compare the different systems according to the different issues (1 to 15) in spite of the difficulty of this exercise. The fifteen issues, identified throughout this work, are recalled below:

- A wide variety of basic objects and a rich set of operations on them (issue 1)
- The possibility to control the delivery of continuous objects (issue 2)
- Interactivity capabilities attached to objects (issue 3)
- Temporal style definitions (issue 4)

- A support for unpredictable objects (issue 5)
- Temporal Composition (issue 6)
- Interactions (issue 7)
- Adaptability to computer illiterate people (issue 8)
- Straightforward design (issue 9)
- Indeterministic scenario authoring capabilities (issue 10)
- Ease of local modifications (issue 11)
- Fast editing/presentation cycle (issue 12)
- Abstraction capabilities (issue 13)
- Multimedia document models (issue 14)
- Multigrids reading support (issue 15)

We use a three levels notation for that table:"-" the issue is not addressed, "+" the issue is a major concern of the system and "+/-" the issue is partially supported. When we have no information, we use a "?".

Issue / Authoring tool	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CMIFed	+ / -	-	+ / -	-	+	+ / -	+	+ / -	-	-	-	+	+	-	+ / -
ISIS	+ / -	-	?	-	-	+ / -	+ / -	+	+	-	+	+	+	-	-
Firefly	+ / -	-	?	-	+ / -	+ / -	+ / -	+	+	-	+	-	-	-	-
Madeus	+ / -	+ / -	+	+	+ / -	+ / -	+ / -	+	+	+ / -	+	+	+	-	+ / -
MHEG	+ / -	-	+	-	+	+ / -	+								
SMIL	+ / -	-	+	-	+ / -	-	+ / -								

Figure 12. Comparative table

Let us note that for MHEG and SMIL languages, we have not filled the authoring issues since currently there is no associated authoring tool for them.

We can draw the following main remarks from this table:

- Each system/language partially fits the first issue but none of them is able to provide fine temporal operations on objects: no distinction between mapping (respectively unmapping) and starting (respectively stopping) of objects and no pause and resume actions.
- Temporal style definition issue (issue 4) has not been fairly taken care of, although it is an essential feature to build attractive multimedia documents (see for example the numerous programmed animations on the Web).
- It is difficult to compare the temporal composition feature (issue 6) of the studied systems because this issue is clearly related to the inputs and the outputs taken into account by the system (see issues 1 to 5). As we have stated in 2.2, even if Allen's relations can be expressed by all of them, they are not equivalent when considering their ability to control the delivery of continuous objects.

- Except in CMIFed, the interaction issue (7) is weakly addressed, mainly because local interactions raise difficulties in a constraint-based context.
- Constraint-based approaches are better adapted for satisfying author needs (issues 9 and 11) thanks to the relative approach they allow.
- Document modeling has not yet been considered for multimedia documents (issue 14) but it is clear that it will become a main issue when such documents become widely used.

Finally, we want to focus on the two following conclusions.

- Constraint-based approaches seem to be more adapted for building powerful authoring tools and they can offer equivalent or higher expressive power capabilities than operational techniques: the author has not to give the duration of all the objects involved in his document (as formulated in the description of our example in section 2). The durations are computed by a temporal formatter, removing the burden of this task from the author and allowing him to obtain reusable scenarios. However, this formatting has to be time-efficient and must provide the solutions desired by the author. Firefly chooses a linear programming technique to perform its formatting phase while incremental considerations motivate the use of constraint networks techniques in Isis and Madeus. Unpredictable objects, partially handled by Firefly and Madeus, arise time performance difficulties in the formatting phase.
- Constraint-based approaches can also provide a better support for the presentation phase in a distributed environment: in such context, the presentation system must deal with network delays that can affect the timing of other objects in the presentation leading to an out of synchronization situation. A global supervision of the timing during the presentation scheduling is therefore required in order to ensure that the author's specifications are met. During the presentation, temporal supervision can be seen as the process of adjusting the scenario in order to meet the timing constraints. In a constraint-based system, this adjustment can be dynamically achieved by the temporal formatter while it is not possible in an operational system in which some durations are statically fixed.

The multimedia authoring domain is still in its infancy but let's bet that it will expand considerably very soon. New standards such as SMIL [31] should give a new boost to this domain because users are eager to get new multimedia authoring tools. Taking into account the distribution of multimedia objects will become the great challenge in the years to come.

BIBLIOGRAPHIE

[1] Allen J.F. "Maintaining Knowledge about Temporal Intervals". Communications of the ACM, vol. Vol. 26, num. No. 11, pp. 832-843, November 1983.

[2] Blakowski G., Steinmetz R.A. Media Synchronisation Survey: Reference Model, Specification, and Case Studies. ``IEEE Journal Of Selected Ares In Communications'', vol. 14, num. 1, pp.5-34, January 1996.

[3] Buchanan M.C., Zellweger P.T. Automatic Temporal Layout Mechanisms. ``Proceedings of the First ACM International Conference on Multimedia'', pp. 341-350, Anaheim, California, August 1993.

[4] Dechter R., Meiri I., Pearl J. Temporal Constraint networks. ``Artificial Intelligence'', vol. 49, pp. 61-95, 1991.

[5] Hamakawa R., Reikimoto J. “Object Composition and Playback Models for Handling Multimedia Data”. Proceedings of the First ACM International Conference on Multimedia, pp. 273-281, ACM Press, Anaheim, California, August 1993.

[6] Hardman L., Bulterman D.C.A., van Rossum G. “The Amsterdam Hypermedia Model: Adding Time and Context to the Dexter Model”. CACM", vol. 37, num. 2, pp. 50-62, February 1994.

[7] Erfle R. “Specification of temporal constraints in multimedia documents using HyTime”. Electronic Publishing, vol. 6, num. 4, pp. 397-411, décembre 1993.

[8] Hillier FS., Lieberman G.J.. ``Operations research'', vol., Holden-Day, San Francisco, 1974.

[9] Jourdan M., Layaïda N., Sabry-Ismail L. “Time Representation and Management in Madeus : an authoring environment for Multimedia documents”. in Proceedings of Multimedia Computing and Networking 1997 ", SPIE 3020, San-Jose, February 1997.

[10] Jourdan M., Roisin C., Tardif L. “User Interface of a new generation of authoring environment of multimedia documents”. Proceeding of the Third ERCIM Workshop on User Interfaces for All, Strasbourg (France), ftp://ftp.inrialpes.fr/pub/opera/publications/uiall.ps.gz 3- 4 November 1997.

[11] IconAuthor 6.0, User’s Guide, Aim Tech.

[12] International Standard ISO 8879. ``Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML)'', International Standard Organization, 1986.

[13] Keramane C., Duda A.. Interval Expressions. ``IEEE Internatiol Conference Multimedia Computing Systems'', USA, November, 1996.

[14] Koegel, Buford J. F. ``Multimedia Systems'', vol. ACM Press, Addison Wesley, , 1994.

[15] Kim W., Kenchammana-Hosekote D. et Srivasta J. Synchronization Relation Tree : A model for Temporal Synchronization in Multimedia Presentation. ``Technical Report TR92-42, Dept. of Computer Science, Univ. of Minnesota, 1992 ''.

[16] Kim M. Y., Song J. Multimedia Documents with Elastic Time. ``Proceedings of the Third ACM International Conference on Multimedia'', ACM Press, ed., pp. 143-154, Polle Zellweger, San Francisco, California, 5-9 November 1995.

[17] Layaïda N., Sabry-Ismail L. Maintaining Temporal Consistency of Multimedia Documents Using Constraint Networks. ``Multimedia Computing and Networking 1996, M. Freeman, P. Jardetzky, H. M. Vin, ed., pp. 124-135, SPIE 2667'', San-José, USA, February 1996.

[18] Legall D.. MPEG : A Video Compression Standard for Multimedia Applications. ``Communications of the ACM'', vol. Vol. 34, num. No. 4, pp. 45-68, April 1991.

[19] Little T.D.C., Ghafoor A. Synchronization and Storage Models for Multimedia Objects. ``IEEE Journal on Selected Areas in Communications", vol. Vol. 8, num. 3, pp. 413-426, April 1990.

[20] MACROMEDIA DIRECTOR. ``User's Guide", Macromedia Inc., 1995.

[21] Perez-Luque M. J., Little T. D. C. A Temporal Reference Framework for Multimedia Synchronisation. ``IEEE Journal on Selected Areas in Communications", vol. 14, num. 1, pp. 36-51, January 1996.

[22] Price R.. MHEG: An Introduction to the Future International Standard for Hypermedia Object Interchange. ``Proceedings of the First ACM Conference on Multimedia", pp. 121-128, ACM Press, Anaheim, Californie, August 1993.

[23] Sénac P., Diaz M., Léger A., de Saqui-Sannes P.. Modeling Logical and Temporal Synchronization in Hypermedia Systems. ``IEEE Journal of Selected Areas on Communications", vol. 14, num. 1, pp. 84-103, 1996.

[24] Song J., Doganata Y, Kim M. and Tantawi A.. Modeling Timed User-Interactions in Multimedia Documents. ``Proceedings of the IEEE International Conference on Multimedia Computing and Systemsn", USA, November, 1996.

[25] Song J., Kim M., Ramalingam G., Miller R., Interactive Authoring Multimedia Documents. ``Visual Language", 1996.

[26] Van Rossum G., Jansen J. , Mullender K. and Bulterman D. CMIFed : a presentation Environment for Portable Hypermedia Documents. ``Proceedings of the ACM Multimedia Conference", California (USA) 1993.

[27] Vidal T and H. Fargier. Contingent duration in temporal CSP : from consistency to controllabilities. ``4th Int. Workshop on Temporal Representation and Reasoning (TIME97)", Daytona Beach (Florida), USA, May 10 -11 97.

[28] Wahl T., Rothermel K. Representing Time in Multimedia Systems. ``IEEE Proceedings of the International Conference on Multimedia Computing and Systems", pp. 538-543, IEEE Computer Society Press, Boston, Massachusetts, 14-19 Mai 1994.

[29] Yu J. and Xiang Y. Hypermedia Presentation and Authoring System. ``Sixth International World Wide Web Conference", Hyper Proceedings, URL: http://www6.nttlabs.com/HyperNews/get/PAPER91.html, California, USA, April 1997.

[30] W3C. ``Working draft specification of XML (Extensible Markup Language)", W3C: http://www.w3.org/TR/WD-xml-lang, August 1997.

[31] W3C. ``Working draft specification of SMIL (Synchronized Multimedia Integration Language)", W3C: http://www.w3.org/TR/WD-smil, November 1997.