

# Le temps dans les documents

**Muriel Jourdan**

Chercheur à l'Inria Rhône-Alpes

**Nabil Layaida**

Chercheur à l'Inria Rhône-Alpes

**Cécile Roisin**

Maître de conférences à l'Université Pierre-Mendès-France de Grenoble

---

## Objectifs et plan de ce chapitre

Dans ce chapitre, nous présentons les concepts et les besoins liés à l'introduction d'une dimension temporelle dans les documents électroniques, ainsi que les solutions actuelles apportées à ce domaine : outils existants, standards, prototypes de recherche. Pour éviter un effet « catalogue », nous avons choisi une description de ce domaine selon une double approche : d'une part, selon les capacités des systèmes à répondre aux caractéristiques de ce nouveau domaine et d'autre part, selon les principes sous-jacents aux différentes approches utilisées actuellement. De plus, nos propos seront illustrés au fil du document à partir d'un même exemple qui servira de point de repère et de comparaison entre les différents systèmes décrits.

La suite de ce document est organisée en quatre grandes parties : une introduction générale sur la notion de documents multimédia et des domaines d'application concernés, la description informelle de l'exemple de référence de document multimédia, l'étude des caractéristiques et des fonctions souhaitées pour la spécification de documents multimédia et enfin la présentation de différents formats et systèmes de documents multimédia représentatifs du domaine.

## 1 Introduction

### Notions de document multimédia et de scénario temporel

Depuis de nombreuses années, les documents électroniques ont fait l'objet d'études qui ont conduit à l'identification de caractéristiques attachées aux documents et classées selon différentes dimensions [1], [2], [9]. Le résultat majeur de ces travaux est la définition de standards comme XML (eXtended Markup Language) [3] qui ont permis de représenter la dimension logique des documents indépendamment de leur contenu et de leur dimension spatiale (forme), facilitant ainsi la portabilité des documents ainsi que leur traitement par des applications variées.

Avec l'intégration de données temporisées comme des éléments vidéo, sonores ou des objets en mouvement sur l'écran, les documents électroniques ont maintenant une *dimension temporelle* en plus des dimensions spatiale, logique et hypertexte. Cette évolution est le résultat des nouvelles capacités technologiques des ordinateurs (CPU, mémoire, périphériques multimédia, supports cdrom, disques ...), des réseaux (débits, protocoles) et des techniques de traitement numérique de l'information non textuelle : sons, images statiques et dynamiques (codage, compression/décompression).

Ainsi, les documents que nous qualifions dans la suite de *documents multimédia* sont non seulement caractérisés par des contenus de nature diverse : *statique* comme les textes, les graphiques, les tableaux et les images, ou *dynamique* comme le son, les vidéos et les animations, mais aussi par l'organisation temporelle de leurs composants. Dans ce chapitre, les unités d'information considérées comme atomiques sont appelées *objets multimédia* et la description de l'enchaînement des objets dans le temps est appelée *scénario temporel*.

### **Des modèles de spécification aux environnements auteur**

Nous présentons comment spécifier les scénarios temporels des documents multimédia et non les modes de représentation des objets média de base (codage) ni à leur mode de création comme la création des vidéos, la saisie des sons, etc. Par conséquent, les environnements auteur que l'on considère doivent être vus comme des compositeurs d'objets déjà créés. Nous verrons comment ces environnements dépendent des modèles de spécification sous-jacents.

### **Domaines d'application**

Le multimédia prend une importance de plus en plus grande dans notre société, notamment par le fait que la communication se fait de plus en plus en utilisant des supports qui intègrent du son, de la vidéo, du texte, etc. Les cdroms, les sites web, les bornes interactives ou la télévision interactive ne sont plus maintenant considérées comme des technologies futuristes mais entrent dans le quotidien des entreprises et du grand public.

Parmi les domaines qui utilisent la technologie du multimédia et plus précisément la notion de document multimédia, l'enseignement assisté par ordinateur (EAO) est actuellement le plus important [15]. En effet, l'EAO peut tirer parti des caractéristiques de différents média pour réaliser des supports pédagogiques attractifs grâce aux images, aux animations et au son, et qui soient de plus interactifs et adaptables aux élèves grâce aux fonctions de navigation hypermédia.

D'autres domaines s'intéressent au multimédia, notamment la médecine. Les données issues des plateaux techniques d'imagerie médicale, comme les images par rayons X, par résonance magnétique ou par échographie, peuvent être exploitées sous forme numérique et intégrées à des données textuelles (par exemple les informations relatives au patient), et sonores ( les commentaires du médecin), pour former de véritables documents multimédia médicaux qui pourront être consultés à distance par les médecins.

### **Nouveaux besoins pour les documents multimédia**

L'introduction d'une nouvelle dimension dans les documents, la dimension temporelle, introduit nécessairement de nouveaux besoins d'expression : la durée des objets, leur placement temporel et leur synchronisation qui constituent le scénario temporel. Ces informations temporelles doivent s'intégrer dans l'ensemble des informations attachées aux documents. Ainsi, la définition d'un *format de représentation* des documents multimédia doit permettre d'exprimer les caractéristiques relatives à chacune des différentes dimensions de ces documents : dimension logique c'est-à-dire l'organisation des informations en composants hiérarchiques, dimension spatiale, dimension hypermédia et dimension temporelle. Par exemple, pour spécifier le déplacement d'un objet sur l'écran, il est nécessaire d'exprimer une information spatiale pour la trajectoire, par exemple les positions initiale et finale

correspondant au déplacement, ainsi qu'une information temporelle comme la durée du déplacement et sa synchronisation avec d'autres objets.

Dans les systèmes traditionnels de traitement de documents orientés texte, le mode de communication entre l'utilisateur et l'application est caractérisé par :

- la présentation à l'écran d'une information principalement spatiale et statique,
- et la lecture active de l'utilisateur qui peut naviguer à travers l'information affichée.

Cette répartition des rôles « actif-passif » est inversée dans les systèmes de communication de l'audio et de la vidéo : l'information est de nature dynamique tandis que le lecteur/auditeur est passif. Dans le contexte des documents multimédia interactifs, ces deux rôles doivent être combinés dans le système de présentation puisque l'information manipulée est à la fois de nature spatiale et temporelle. De plus, le système doit permettre certaines formes d'interaction sur le document présenté, comme la navigation intra- ou extra-documents par le biais d'hyperliens.

La nature dynamique des objets manipulés, tels que la vidéo et l'audio, ainsi que la définition de leur ordonnancement temporel (le scénario) rendent plus complexe la réalisation d'outils auteur. Le principe statique du Wysiwyg : What You See Is What You Get, dans lequel l'information présentée à tout instant du processus d'édition correspond à l'information finale, ne peut s'appliquer à l'édition du scénario temporel des documents multimédia. Il n'est en effet pas possible de spécifier un comportement dynamique, par exemple un enchaînement entre deux vidéos, et d'en percevoir de façon immédiate et instantanée le résultat. C'est pourquoi nous distinguons deux étapes dans le processus de conception des documents multimédia : l'*étape de spécification* (ou édition) et l'*étape de présentation* (ou exécution). Un des thèmes de recherche dans le domaine des outils d'édition multimédia est la conception d'environnements qui rendent le plus souple possible le passage d'une étape à l'autre.

Pour aller plus loin dans cette analyse des besoins, nous classons les fonctions nécessaires pour la spécification des documents multimédia selon deux catégories (que nous développerons en section [3](#)) :

1. Les possibilités d'expression du langage de spécification des scénarios temporels.
2. Les fonctions de base qu'un environnement de composition multimédia doit fournir aux auteurs.

### **État de l'art dans le domaine de la spécification des documents multimédia**

Les travaux actuels sur les documents multimédia portent surtout sur la *spécification des scénarios temporels* puisque c'est la caractéristique nouvelle à prendre en compte. On peut classer les techniques proposées pour la spécification des documents multimédia selon deux approches :

1. Les techniques opérationnelles ou impératives : axe de temps absolu (Director [\[10\]](#), HyTime [\[16\]](#)), programmation par scripts (Lingo dans Director, MHEG [\[17\]](#)), spécification par structures d'exécution comme les arbres (CMIFed [\[13\]](#) et SMIL [\[18\]](#), le langage proposé par le W3C) ou les réseaux de Pétri (HTSPN [\[11\]](#)).
2. Les techniques déclaratives par relations temporelles, Firefly [\[4\]](#), Isis [\[12\]](#), Madeus [\[8\]](#), dans lesquelles l'auteur déclare les placements temporels souhaités sans donner

toutes les informations temporelles attachées aux objets. Plus précisément, ces techniques s'appuient sur l'algèbre d'Allen [19] que nous décrivons dans la section [3.1.2](#) ci-dessous.

Ces différentes approches seront développées dans ce chapitre, mais il est dès à présent intéressant de noter que tous les outils commerciaux actuels font appel à des approches impératives à base de scripts, se rapprochant plus d'outils de programmation que de véritables outils d'édition.

## 2 Un exemple de document multimédia

Nous décrivons ici un exemple de document multimédia comprenant différents types de média ordonnés de façon suffisamment complexe pour illustrer un nombre important de problèmes posés par l'édition multimédia. De plus, cet exemple sera repris et (partiellement) spécifié à l'aide des différents systèmes et langages présentés en section [4](#).

Appelons "J'Com" une agence de communication qui répond à un appel d'offre du comité d'organisation de la coupe du monde de football 2002 sur la création d'une mascotte. Pour présenter son offre de manière attractive, elle conçoit sa proposition sous la forme d'un document multimédia. Dans ce qui suit, nous donnons une spécification informelle du scénario de ce document, en identifiant (mots en italique) les composants qui seront repris dans les scénarios formels.

Le scénario est composé de deux parties: (1) une présentation de l'agence (appelée *Agence*) suivie (2) d'une présentation de la mascotte (appelée *Mascotte*). Par défaut, ces deux parties sont présentées de manière séquentielle mais des liens hypermédia (*H1* et *H2*) permettent au lecteur d'aller de l'une à l'autre comme il le souhaite.

La présentation de l'agence doit durer environ 3 minutes. Elle est composée principalement d'une séquence de trois objets : une bande son (*Passé*) qui présente l'historique de l'agence, suivie par l'affichage du nom de l'agence (*Nom*), lui-même suivi par une succession d'images présentant les meilleures réalisations de l'agence (*PressBook*). La bande son dure approximativement 1 minute, l'objet texte *Nom* 30 secondes environ et le *PressBook* 1 minute 1/2 maximum. Se rajoute à cette séquence un film de 1 minute environ (*Visite*) qui précise la situation géographique de l'agence et conduit le lecteur à travers ses bureaux. Ce film doit démarrer de telle façon que le nom de l'agence (*Nom*) apparaisse à l'écran en même temps que l'image de l'entrée du bâtiment de l'agence (approximativement 20 secondes après le début du film). Enfin, durant la présentation du *PressBook*, le Jingle (*Jingle*) de la société est joué.

La présentation de la mascotte débute par une animation de synthèse de la proposition de l'agence (*Proposition*). Cette animation se termine par une dernière image fixe de la mascotte. Celle-ci (*Image*) reste affichée à l'écran durant 30 secondes. Elle est alors accompagnée d'une bulle (*Bulle*) qui se place à droite de la bouche de la mascotte et sur laquelle est marquée son nom. De plus, dès le début de l'animation, un message sonore (*Message*) invite le lecteur à appuyer sur un bouton (*Bouton*) pour connaître plus vite le nom de la mascotte. S'il choisit cette solution, la bulle apparaît tout en haut de l'écran, elle descend alors progressivement jusqu'à se trouver à sa position finale toujours au moment où la dernière image de la mascotte apparaît à l'écran.

La figure 1 présente un diagramme de temps correspondant à une des exécutions possibles de ce document. Dans cette exécution, le lecteur a cliqué au bout de 140 secondes sur l'hyperlien *H1* lui permettant de passer directement à la présentation de l'agence. 40 secondes plus tard, il a appuyé sur le bouton *Bouton* pour voir apparaître le nom de la mascotte.

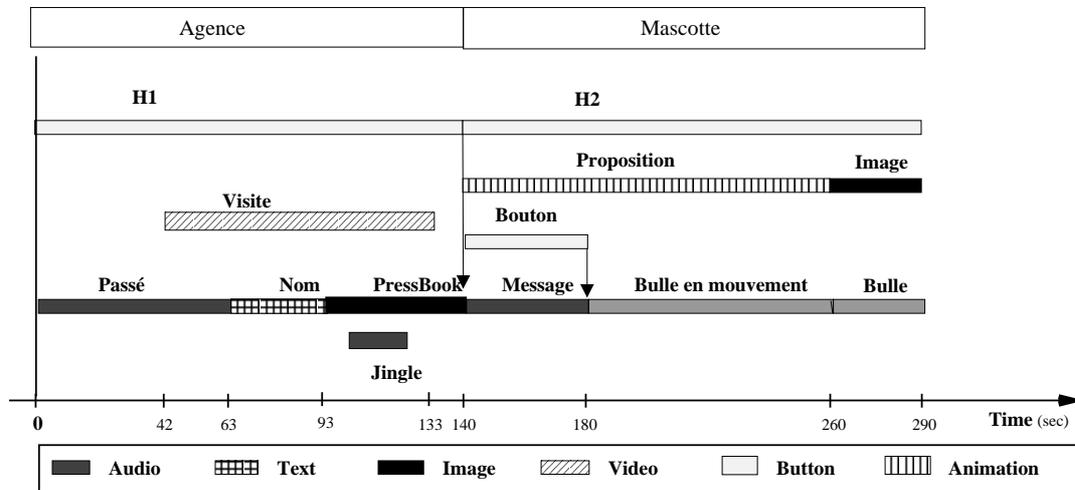


Figure 1 - Exemple d'exécution du scénario de la proposition de mascotte

### 3 Caractéristiques des applications de documents multimédia

La variété des travaux dans le domaine multimédia reflète l'étendue de la palette des besoins qui doivent être couverts par les systèmes multimédia et plus particulièrement par les systèmes auteur de documents multimédia. Actuellement, il est clair que les fonctions nécessaires à ce domaine ne sont que partiellement remplies par les applications existantes : il suffit de constater la complexité du processus de création des titres multimédia : succession de sept étapes, chacune faisant appel à des intervenants spécialistes de domaines très différents les uns des autres (scénaristes, graphistes, photographe, informaticiens, ...).

La suite de cette section a pour objet d'identifier les caractéristiques à prendre en compte dans les systèmes multimédia en les organisant en deux catégories : le pouvoir d'expression et les fonctions nécessaires pour les auteurs.

#### 3.1 Possibilités d'expression dans les scénarios temporels

Le pouvoir d'expression d'un système correspond à la capacité de ce système à traiter la plus grande gamme possible de scénarios que les auteurs peuvent vouloir spécifier. Cette caractéristique est bien évidemment difficile à évaluer dans la mesure où elle dépend fortement de l'expérience des utilisateurs. Nous pouvons cependant identifier trois classes de critères.

##### 3.1.1 Caractéristiques des objets multimédia

Les documents multimédia sont non seulement caractérisés par des objets de base de nature variée mais aussi par tout un ensemble de comportements qui peuvent ou non leur être associés. C'est ce que nous analysons ci-dessous.

## Nature des objets de base

Le qualificatif de multimédia se rencontre dès lors qu'une application traite des documents qui ne sont plus seulement textuels (plusieurs média de base sont utilisés). C'est ainsi que des documents comportant uniquement du texte et des images sont appelés multimédia. À l'autre extrême de cette échelle, on trouve des documents caractérisés par une dynamique beaucoup plus importante grâce à l'intégration d'informations textuelles, d'images mais aussi du son, de la vidéo, des images animées, des éléments contrôlés par des programmes externes appelés aussi *plug-ins*, etc.

Ces différents objets sont classés en deux catégories de part leur mode de présentation dans le temps :

- *Les objets discrets* : leur contenu est délivré de façon instantanée, comme le texte et les images statiques.
- *Les objets continus* : leur contenu est délivré de façon progressive comme les vidéos, les sons ou les animations.

De même, les objets multimédia peuvent être caractérisés par leur mode de perception :

- *Les objets visibles*, c'est-à-dire que l'on peut afficher : texte, images, vidéo, animations. Pour les objets continus de cette classe, il est nécessaire de pouvoir distinguer les instants d'affichage/désaffichage de ceux correspondant à l'exécution proprement dite, par exemple, différencier l'instant d'affichage de la première image d'une animation de l'instant du début de son déroulement.
- *Les objets audibles* comme les sons.

Pour chaque catégorie d'objets, différents formats de codage existent (Ascii pour les textes, jpeg, gif, tiff, png, pour les images, au, wave pour les audio, mpeg, ... pour les vidéos, etc.). Un critère important pour un environnement multimédia est sa capacité à accepter de nombreux formats et même à offrir une certaine forme d'extensibilité de façon à pouvoir prendre en compte de nouveaux formats.

## Style et activation

À ces différentes catégories d'objets, le système doit permettre d'y attacher des informations de présentation appelées *règles de style*. Le jeu de règles dépend de chaque type de média, comme la fonte pour les textes ou le volume sonore pour les sons. Ces paramètres de présentation doivent pouvoir dépendre du temps de façon à réaliser des *effets de style* lors de la présentation d'un objet. Par exemple, le déplacement d'un texte selon une trajectoire, la modification graduelle de la couleur, du volume du son, etc. Dans notre exemple, le déplacement de la bulle sur l'écran est un effet de style.

De même, l'auteur doit pouvoir créer une structure hypermédia (voir section [3.1.4](#)) en rendant activable n'importe quel type d'objet. Un objet activable, appelé aussi *bouton*, est un objet sur lequel une action prédéfinie dans le document, par exemple l'activation d'un lien hypermédia, est déclenchée lorsqu'il est activé. Là encore, cette propriété d'activation doit pouvoir être définie pour un intervalle de temps durant lequel le lecteur aura la possibilité d'activer le lien.

### 3.1.2 Composition temporelle

Sur le plan du pouvoir d'expression, un langage temporel d'un outil multimédia doit permettre la spécification de schémas de synchronisation arbitrairement complexes. Une évaluation de l'expressivité peut se ramener donc à une mesure du nombre de scénarios exprimables.

On verra en section 4 que différents modes de composition temporelle ont été expérimentés, depuis la spécification par placement absolu, l'utilisation de langages de programmation ou d'opérateurs de composition jusqu'à l'utilisation d'algèbres de relations.

Différents travaux théoriques, qui proviennent notamment du domaine de la planification en robotique dans lequel le problème de la synchronisation temporelle est cruciale, ont permis de formaliser l'information temporelle des objets ainsi que des relations entre eux. Ainsi, tout objet multimédia peut être manipulé à travers trois informations temporelles principales (cf. figure 2) :

- Son instant de début.
- Sa durée de présentation.
- Son instant de fin.

L'une de ces trois informations est redondante. En effet, il est possible de calculer l'une en fonction des deux autres. Mais le choix des informations retenues fait partie du langage et donc du modèle, car la mise en relation des éléments se fait à partir de ces informations. Il existe ainsi deux façons de représenter le déroulement d'un scénario : à travers les changements qui surviennent, comme « *à la fin de la bande son Passé, le nom de l'agence Nom commence à s'afficher* », ou au contraire en reliant globalement les activités entre elles comme « *le film vidéo Visite de l'Agence est présenté pendant la bande son Passé* ». Ceci débouche sur deux types de représentation :

- Une représentation fondée sur les *instants*, en particulier les instants de début et de fin des objets.
- Une représentation fondée sur les *intervalles* correspondant à la durée des objets.

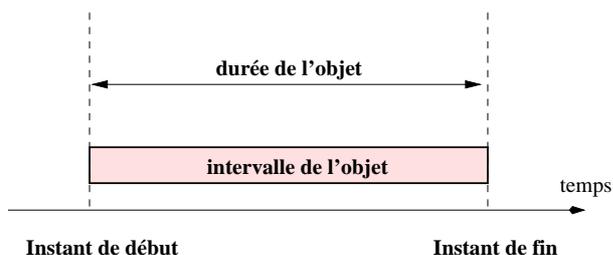


Figure 2 - Information temporelle attachée aux objets

Dans les relations à base d'instants, les unités temporelles considérées dans les relations sont les instants de début et de fin des objets. Étant donnés deux instants dans un scénario, trois relations peuvent exister entre eux. Un instant peut en précéder ( $<$ ), lui succéder ( $>$ ) ou lui être égal, c'est-à-dire coïncider temporellement ( $=$ ).

Dans les relations à base d'intervalles, les relations possibles entre deux éléments multimédia se réduisent à toutes les combinaisons de placement possibles de deux intervalles sur une droite orientée. Le modèle le plus général, proposé par Allen [19], dresse la liste exhaustive de toutes ces relations. La liste des combinaisons possibles entre éléments multimédia

comporte ainsi 13 relations consistant en 7 relations de base (cf. figure 3) et leurs relations inverses, l'égalité étant elle-même son inverse. Ces treize relations se répartissent en deux classes : celle des relations de séquentialité notée *Seq* dans le tableau et celle des relations introduisant le parallélisme de présentation, notée *Par*. Dans l'avant-dernière colonne du tableau, nous présentons la traduction de chaque relation sous la forme d'une suite de relations à base d'instant. Les variables  $x$  et  $y$  représentent des intervalles et les notations  $x^-$  et  $x^+$  correspondent respectivement aux instants de début et de fin de l'intervalle  $x$ .

Relation	Symbole	Inverse	Relation à base d'instant équivalente	Classe
<b>x avant y</b>	b	bi	$x^- < x^+ < y^- < y^+$	Seq
<b>y suit x</b>	m	mi	$x^- < x^+ = y^- < y^+$	Seq
<b>x recouvre y</b>	o	oi	$x^- < y^- < x^+ < y^+$	Par
<b>y termine x</b>	f	fi	$x^- < y^- < x^+ = y^+$	Par
<b>y pendant x</b>	d	di	$x^- < y^- < y^+ < x^+$	Par
<b>x démarre y</b>	s	si	$x^- = y^- < x^+ < y^+$	Par
<b>x égale y</b>	e	ei	$x^- = y^- < x^+ = y^+$	Par

Figure 3 - Les relations d'Allen

Outre ces treize relations de placement d'intervalles les uns par rapport aux autres, d'autres opérateurs de composition sont nécessaires, notamment pour exprimer la *causalité*, comme lorsque l'occurrence d'un instant de fin d'un élément provoque l'arrêt d'un autre, même si ce dernier n'a pas restitué tout son contenu à l'utilisateur. C'est le cas par exemple du message sonore *Message* et du bouton *Bouton* dans la partie *Macotte* du scénario.

Ainsi, la composition parallèle peut exprimer différentes sémantiques comme :

- La composition parallèle avec égalité de durée sans perte d'information des objets composés, c'est la sémantique de la relation *égale* d'Allen.
- La composition parallèle avec interruption, c'est-à-dire avec une perte éventuelle d'information d'un des composés :
  - soit lorsque le premier provoque la terminaison du plus long ; cet opérateur causal est appelé *parmin*,
  - soit lorsque l'un des composés désigné se termine quelque soit l'état de l'autre (il est donc éventuellement coupé) ; cet opérateur est appelé *parmaster*.

Un des moyens d'évaluer le pouvoir de composition d'un langage est de voir quelles relations parmi les treize identifiées ci-dessus et quelles relations de causalité peuvent être exprimées dans le langage.

### 3.1.3 Contrôle de la restitution de l'information

Il est important que l'auteur puisse exprimer le fait que certains objets continus doivent entièrement restituer leur contenu lors de la présentation : ils ne seront pas coupés avant leur fin pour respecter certaines synchronisations temporelles ou spatio-temporelles. C'est le cas par exemple de la synchronisation spatio-temporelle entre la dernière image de l'animation et

du déplacement de la bulle de notre exemple : l'auteur souhaite que ces deux instants soient simultanés et qu'en plus la bulle soit positionnée à un endroit précis par rapport à l'image.

Pour cela, le système de présentation doit implanter les deux classes d'opérateurs de composition parallèle avec égalité de durée (*égale* et *parmin*) définis dans la section précédente. En fait, dans les systèmes actuels, c'est uniquement la sémantique du *parmin* qui est le plus souvent implantée car elle plus facile à mettre en oeuvre. En effet, la première suppose que l'on sache contrôler la durée des objets de manière à assurer les contraintes de terminaison de ce type. Or certains objets sont incontrôlables.

On parle d'*objet incontrôlable* lorsque la durée d'un objet est dépendante des conditions de présentation. C'est le cas par exemple, d'une vidéo dont la durée de présentation sera influencée par la charge de la machine sur laquelle s'effectue la présentation, ou de manière plus évidente d'un bouton d'interaction. Dans notre exemple, les trois objets activables (les deux hyperliens *H1* et *H2* ainsi que le *Bouton*) sont incontrôlables. C'est aussi le cas des objets sonores et vidéo, alors que l'animation de synthèse peut être considérée comme un objet contrôlable si elle est donnée comme une succession d'images au format gif dont la durée peut être contrôlée.

Dès que dans un scénario apparaissent des objets incontrôlables et des synchronisations temporelles de type *égale* qui portent sur des instants de fin situés après ces objets, il devient nécessaire (et c'est là toute la difficulté) de passer d'un calcul statique (avant le lancement de la présentation) des durées des objets, à un calcul dynamique (pendant la présentation) de celles-ci. Par exemple, pour respecter la synchronisation spatio-temporelle, la durée associée au déplacement de la bulle doit être calculée au moment précis où le lecteur appuie sur le bouton d'interaction et non avant le lancement de la présentation.

#### **3.1.4 Types de navigation**

Les lecteurs cherchent à accéder aux informations d'un document imprimé de façons multiples : outre la lecture linéaire donnée par le parcours page après page du document, on peut citer l'accès depuis une table des matières, depuis des renvois comme les notes, l'accès au hasard, le parcours de la fin au début, etc. Dans le cadre d'une lecture électronique, cette fonction reste autant nécessaire et les environnements de navigation hypertexte ont démontré clairement l'intérêt du support électronique pour cette fonction.

Dans le cas des documents multimédia, la nature dynamique de l'information rend encore plus nécessaire et complexe la navigation : outre les hyperliens « classiques », étendus pour prendre en compte la dimension temporelle, des fonctions de parcours dans le temps sont aussi nécessaires, de façon similaire aux boutons d'accélération, pause, etc. des lecteurs de disques ou de cassettes audio ou vidéo. Nous distinguons donc :

- *La navigation indépendante du document* : cette forme de navigation est fournie au travers de boutons de contrôle du temps (TAC : Temporal Access Control) au niveau du système de présentation, comme les boutons de pause, reprise, accélération en avant ou en arrière, saut d'une scène à l'autre, etc. Ces fonctions de navigation étant indépendantes du document présenté, elles n'apparaissent pas dans la spécification du scénario du document.
- *La navigation dépendante du document* : cette forme de navigation est conçue par l'auteur du document de façon à aider les lecteurs dans leur parcours du document,

comme dans notre exemple, le bouton qui permet de voir plus rapidement le nom de la mascotte, elle fait donc partie de la spécification du document. Cette navigation est définie par des *objets activables* (ou boutons d'interaction) que nous analysons ci-dessous.

Les interactions que les auteurs peuvent spécifier dans les documents multimédia peuvent être classées dans trois catégories en fonction de la sémantique qui leur est associée (la figure 4 illustre ces trois cas) :

- *Les interactions globales* (figure 4.b) : l'activation d'une interaction globale a pour conséquence de stopper tous les objets en cours de présentation à cet instant (c'est le cas des liens de navigation classiques comme les hyperliens entre les deux parties de notre exemple) pour démarrer la présentation d'un autre document (lien externe) ou d'une scène du document courant (lien intra-document).

Un besoin associé à ce mode de navigation est la possibilité de revenir en arrière comme dans les navigateurs HTML. Cependant, la dimension temporelle des documents multimédia introduit une plus grande difficulté pour la mise en oeuvre de cette fonction puisqu'il faut revenir à l'instant de l'activation du lien, c'est-à-dire, être capable de continuer l'exécution de tous les objets à partir de l'instant où ils avaient été stoppés avant l'activation du lien.

- *Les interactions locales* (figure 4.c) : l'activation d'une interaction locale a pour conséquence de ne stopper qu'une partie des objets en cours de présentation à cet instant. C'est le cas du bouton de la partie *Mascotte* qui stoppe le message audio mais n'agit pas sur le reste du scénario de telle sorte que l'animation puisse continuer son exécution. Ce type d'interaction est plus difficile à gérer car certaines synchronisations sont difficiles à assurer en présence d'interactions locales, comme la synchronisation de la bulle avec l'animation.
- *Les interactions d'inclusion* (figure 4.d) : l'activation d'une interaction d'inclusion a pour conséquence d'ajouter un sous-scénario à partir de l'instant courant de la présentation, par exemple, un texte explicatif et un commentaire à une image affichée. Les objets du sous-scénario ne peuvent cependant pas être synchronisés avec les autres objets du document.

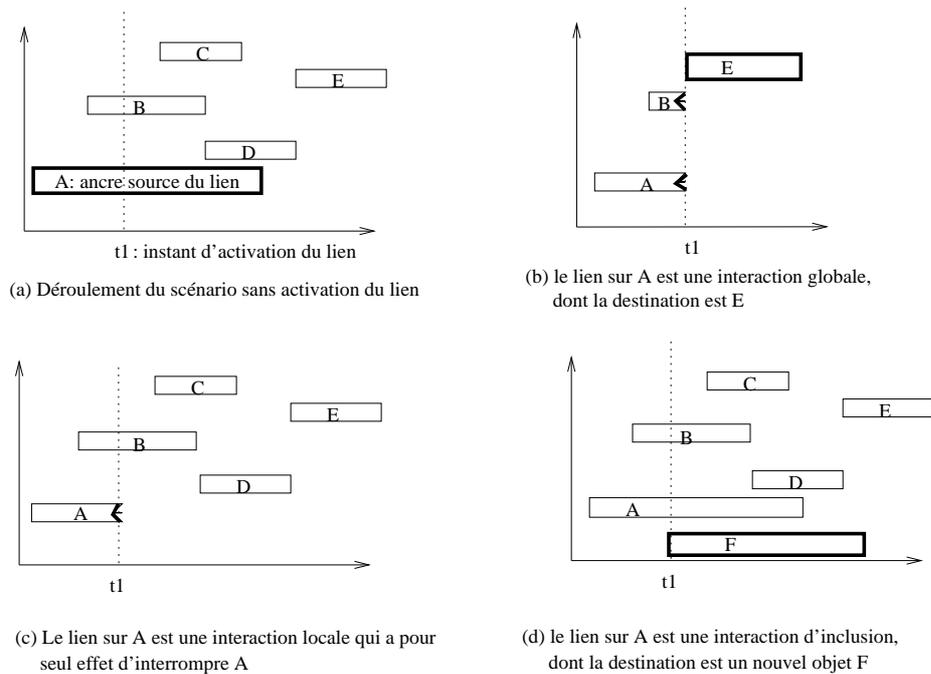


Figure 4 - Les trois types d'interactions dépendantes du document

### 3.2 Fonctions nécessaires pour les auteurs

Le processus de définition du scénario temporel d'un document multimédia est complexe car l'objet à construire est un objet dynamique et dont les comportements varient d'une présentation à une autre du fait des réactions aux interactions du lecteur. L'environnement utilisé par l'auteur doit l'aider à maîtriser cette complexité. Plus précisément, nous identifions les critères suivants (qui seront analysés dans la partie 4 sur les langages et outils) :

- Le langage de spécification doit être utilisable sans compétence en programmation informatique. Imposer à un auteur de documents multimédia de connaître les notions de variable et d'instruction limite considérablement l'accès à l'écriture de ce type de documents.
- Le langage doit permettre le passage simple et souple de la représentation mentale que se fait l'auteur du document à un véritable scénario formalisé dans ce langage. L'auteur qui conçoit un document s'en fait sa propre représentation mentale qu'il peut exprimer sous forme d'une description informelle, comme notre exemple en section 2.

On retrouve dans les descriptions informelles principalement trois types d'information : absolu, par exemple « l'image X sera présentée 20 secondes après le début du document », relatif « l'image X sera présentée après la vidéo » et imprécis « le document doit durer environ 1 minute ». Le codage de cette représentation dans les termes de l'environnement doit être le plus direct possible et accepter ces trois types d'information sans en imposer une à l'auteur.

- Le langage de spécification doit être modulaire pour permettre à l'auteur de réduire la complexité de son document et aussi de réutiliser des parties déjà spécifiées.
- L'environnement auteur doit permettre de modifier facilement les documents en cours de conception. La construction d'un document (multimédia ou non) est un processus

itératif dans lequel le document courant subit de nombreuses modifications. Pour cela, l'auteur doit pouvoir :

- Effectuer des modifications locales sans avoir à reconsidérer globalement le document. Par exemple, l'auteur doit pouvoir modifier la durée d'un objet sans pour autant mettre en cause toutes les synchronisations temporelles qu'il a pu auparavant spécifier.
- Passer aisément du mode édition vers le mode présentation (et vice-versa) pour d'une part aller voir le résultat de ces modifications (mode édition vers mode présentation) et d'autre part pour retrouver facilement les informations qu'il souhaite modifier (mode présentation vers mode édition).
- L'environnement auteur doit offrir un support visuel permettant de percevoir le scénario en cours de spécification. L'auteur a besoin d'avoir une vue globale du scénario qu'il est en train de construire. La phase de présentation, bien qu'indispensable, n'est pas une réponse suffisante à ce besoin car elle ne montre à l'auteur qu'une succession d'instantanés qui s'enchaînent les uns après les autres mais ne lui permet pas de percevoir le document dans sa globalité. De plus, elle n'est le reflet que d'une des présentations possibles du document alors que celui peut en avoir plusieurs à cause en particulier des objets non contrôlables.

D'autres besoins commencent à émerger avec l'utilisation grandissante de l'information multimédia dans des domaines d'application aussi variés que l'éducation, la médecine ou le tourisme. Pour limiter les coûts de réalisation en factorisant au mieux les tâches de conception de documents multimédia, deux directions sont à considérer, au niveau spécifique, c'est-à-dire lors de la création d'un même document, ou au niveau générique à travers la notion de classe de documents :

- Il faut pouvoir concevoir facilement des documents comportant plusieurs grilles de lecture. Il arrive souvent qu'un document multimédia puisse être présenté différemment selon le contexte de lecture pour s'adapter par exemple aux compétences des lecteurs, à leur disponibilité, etc. Ces différentes grilles de lecture ne doivent pas être conçues comme autant de documents différents reliés par des liens hypertexte, car elles partagent un nombre important de données comme les objets ou des parties de scénario temporel.
- Il faut pouvoir spécifier des modèles génériques de documents multimédia. Des standards comme SGML ou plus récemment XML ont permis aux auteurs de documents classiques de capitaliser leur travail lors de l'écriture d'une classe homogène de documents (par exemple des documents techniques) grâce à la notion de modèle de documents établi une fois pour toutes. Ils ont aussi amélioré les capacités de traitements automatiques des documents (conversion de formats, extraction d'information, etc.). De manière équivalente, la définition de modèles de documents multimédia aura un impact important sur la façon dont sont conçus puis traités ces documents. Des exemples de classes de documents multimédia sont aisément identifiables dans les différents domaines dans lesquels le multimédia aura un impact : modèles pour des guides touristiques, des présentations d'entreprises, etc.

## **4 Étude de différents langages et systèmes multimédia**

Cette étude est illustrée à travers l'exemple de la section [2](#) et situe les différentes approches selon les critères de la section [3](#). L'objectif est de comprendre les principes sous-jacents aux outils et langages existants en les classant dans deux types d'approches :

- Les approches opérationnelles : trois familles d'outils ont été identifiées, les outils qui s'appuient sur un axe temporel absolu, les langages de programmation et les représentations par structures de graphes (arbre ou réseau de Pétri).
- Les approches à base de contraintes.

Ce regroupement en deux familles nous permet de factoriser l'analyse de la situation de chaque langage/système particulier par rapport aux différents critères précédemment présentés.

## 4.1 Les approches opérationnelles

### 4.1.1 Axe temporel absolu

Les environnements auteur fondés sur les axes de temps absolu, ou *timeline*, sont les plus répandus dans les systèmes commerciaux comme Flash et Director de Macromedia [10] ou MAestro [5]. De même, le standard HyTime [16] est un langage qui s'appuie aussi sur la métaphore de la ligne de temps.

#### Environnements auteur fondés sur les timelines

Ces systèmes s'appuient sur une métaphore graphique d'un axe pour représenter la dimension temporelle d'un document multimédia (le flot de données formé par la succession d'unités de présentation). Les interfaces à base de timelines (voir l'axe temporel de Director sur la figure 5) représentent les activités de présentation concurrentes au moyen de plusieurs axes parallèles dont chacun est réservé à un média particulier : sur la figure, l'objet *Passé* est sur le canal sonore, l'objet *Nom* (texte) est sur le canal 1, l'objet *PressBook* (suite d'images) est sur le canal 2 et l'objet *Visite* (vidéo) est sur le canal 3. Un axe commun (en haut de la figure 5), partagé par tous les autres axes, est utilisé comme référence pour représenter le temps physique : l'unité est exprimée en nombre de trames (unité logique) à laquelle on peut attacher une durée (ici, chaque trame est affectée d'une même durée de 5 secondes). Enfin le bouton d'interaction *HI* est placé sur le canal 5.

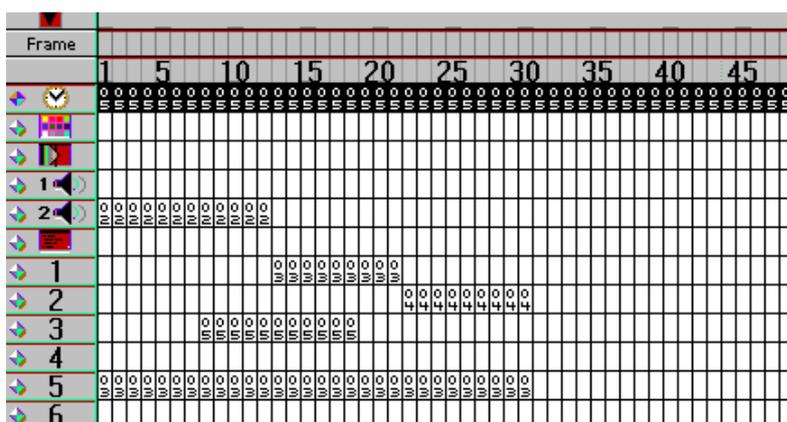


Figure 5 - Vue timeline du scénario de présentation de l'Agence avec Director

Les éditeurs à base de timelines répondent bien aux deux premiers besoins pour les auteurs identifiés en section 3.2 : en effet, l'interface utilisateur est accessible aux non informaticiens et rend compte de façon intuitive du placement temporel des différents événements de

présentation. Par contre, les inconvénients sont multiples et liés aux effets de bord que cette même interface engendre dans les différents traitements d'édition, à savoir :

- La modification de la durée ou de la date de début ou de fin d'un élément nécessite souvent le repositionnement à la main des autres éléments. En effet, les dépendances temporelles entre éléments ne sont pas maintenues sous une forme relative, tout y est rapporté au temps physique. Une opération quelconque sur le document est donc susceptible d'engendrer des changements en chaîne qui doivent être effectués à la main un par un, ce qui rend la tâche d'édition fastidieuse et source de beaucoup d'erreurs.
- À cause de la datation explicite des différents événements de présentation comme les débuts et fins d'éléments, il est impossible de représenter des éléments dont la valeur de durée est imprévisible comme les interactions de l'utilisateur.
- Comme conséquence de la place prédominante accordée au temps au niveau de l'interface d'édition, la représentation logique du document est aplatie voire rendue inexistante. L'organisation modulaire du document qui encourage la réutilisation de ses parties est donc très peu marquée.
- Les possibilités de navigation sont limitées : en effet, la plupart des systèmes timeline offrent la possibilité de naviguer entre des parties de documents en introduisant des liens vers des dates explicites de la présentation (pointeur vers l'instant  $i$  de la présentation). Alors que ces opérations semblent relativement intuitives, le résultat de l'activation d'un lien pointant vers un instant  $i$  est difficile à percevoir, notamment par le fait que les objets supposés être déjà activés à cet instant ne sont pas lancés par le système au moment du saut. Le comportement obtenu ne correspond donc plus vraiment au scénario spécifié dans le timeline.

### Spécification temporelle avec HyTime

HyTime [16] est un langage standard, extension de SGML, proposé pour : (1) la structuration des documents multimédia et hypermédia, (2) la représentation des liens hypertexte, et (3) la spécification de l'ordonnancement des événements spatiaux et temporels ainsi que de leur synchronisation. Le mode de spécification qu'il offre, à base de types d'**espaces finis de coordonnées** (FCS : *Finite Coordinate Space*) le classe donc dans la famille « axe de temps absolu ». En effet, les scénarios sont décrits en plaçant des événements à des dates précises du temps absolu et/ou à des positions géométriques sur l'écran. Chaque type d'objet multimédia (texte, audio ou vidéo) est associé à un FCS particulier. Par exemple un objet vidéo peut avoir quatre axes de coordonnées dans son FCS tels que chacun de ces axes ait une unité de mesure différente (temps, position en X et en Y sur l'écran et numéro d'image dans une séquence). Tout point d'un FCS correspondra donc à un événement potentiel de la présentation d'un document. De ce fait, la spécification d'un événement se fera par un ensemble de points associés à différents FCS. L'ensemble de ces points détermine le lieu géométrique et temporel d'occurrence de l'événement.

HyTime étend les capacités de désignation de SGML en permettant de définir des pointeurs associés à des données dispersées (contenues dans divers documents ou disséminées à travers un réseau) qui contiennent, de manière transparente, des informations relatives à leur localisation. HyTime peut supporter trois types d'adressage : par nom, par position dans un espace de coordonnées (position temporelle ou spatiale sur écran) et par référence sémantique (lien hypermédia).

Il existe un nombre très faible de réalisations et d'outils pour le standard HyTime, probablement à cause de sa complexité et de la difficulté que les développeurs affrontent pour le lire et le comprendre. Néanmoins, quelques travaux ont été effectués à partir de HyTime comme par exemple les spécifications SMDL (*Standard Music Description Language*) qui est un langage conforme au standard HyTime pour représenter les informations musicales accompagnées par d'autres informations textuelles et graphiques. Les axes absolus (FCS) de HyTime conviennent parfaitement pour l'organisation linéaire de la musique sur l'axe du temps. Un autre exemple de l'utilisation de HyTime est la réalisation d'une structure d'accueil SGML/HyTime répondant aux besoins de stockage et d'exploitation de la documentation technique d'avions à l'Aérospatiale.

L'un des principaux avantages du standard HyTime est le fait qu'il propose une modélisation générale d'un document hypermédia capable de supporter la représentation d'un grand nombre de configurations de documents. Du point de vue de l'expression des contraintes temporelles, il permet de modéliser tous les scénarios de présentation possibles si les objets média ont des comportements temporels déterministes. En revanche, le standard HyTime ne supporte pas les objets média incontrôlables à cause de l'utilisation des axes de temps absolus qui minimisent la capacité d'adaptation aux conséquences d'indéterminisme.

La trop grande complexité du standard et son incapacité à prendre en compte l'indéterminisme des scénarios temporels sont les deux raisons qui expliquent le nombre très faible d'applications basées sur Hytime. Cependant, les concepts mis en place dans le standard en ce qui concerne la spécification des liens hypertextes sont réutilisés dans d'autres standards comme XLink développé par le consortium World Wide Web.

#### 4.1.2 Langages de scripts

Les outils à base de timeline, du fait de leur pouvoir d'expression limité, sont le plus souvent complétés par des langages de programmation (appelés scripts). C'est le cas de Director dont le langage de script associé s'appelle Lingo [10]. L'environnement IconAuthor [6] permet de spécifier également un flot d'exécution au travers d'une interface à base d'icônes. De même, le standard MHEG [17] s'appuie sur le paradigme des langages de programmation orientés objets.

Il est clair qu'avec un langage de programmation, il est possible d'exprimer des scénarios aussi complexes que l'on veut : le scénario est spécifié sous forme opératoire par un ensemble d'instructions « événement & conditions -> actions ». La première partie de notre exemple est décrite de cette façon en pseudo-code dans la figure 6.

---

```
Lancer audio "Passé.au";  
  
Lancer bouton "H1";  
  
Attendre (40);  
  
Lancer vidéo "Visite.mpeg";  
  
Quand end(Passé)
```

```
Afficher texte "Nom.txt";  
  
Attendre (30);  
  
Effacer texte "Nom.txt";  
  
Afficher image "PressBook.pic";  
  
Attendre (25);  
  
Lancer audio "Jingle.au";  
  
Attendre (65);  
  
Effacer image "PressBook.pic";  
  
Quand activé(H1) Aller-à "Mascott";
```

---

Figure 6 - *Script de la partie de la présentation de l'Agence*

Le point faible de ces approches vient de leurs capacités limitées à répondre aux besoins des auteurs (cf. section [3.2](#)) : les compétences en programmation requises pour les utiliser restreignent leur emploi aux seuls informaticiens ; il est de plus très difficile de percevoir l'enchaînement temporel des objets par la simple lecture d'un script car la composition temporelle est dispersée dans le script ; enfin, comme pour tout langage de programmation, la mise à jour d'un scénario est une tâche souvent délicate du fait des erreurs qu'il est facile d'y introduire.

### **Le standard MHEG**

Le but du groupe MHEG [\[17\]](#) (*Multimedia and Hypermedia Information Coding Experts Group*) est la définition et la standardisation d'un format portable de représentation multimédia pour faciliter l'échange d'objets entre sites. Ceci a abouti à un modèle objet constitué d'une hiérarchie de classes *MHEG object* comprenant les classes *Content* (qui permettent de décrire le contenu d'objets média), *Script* (pour définir des comportements sous forme de programmes), *Link* et *Composite*, ces deux derniers étant décrits ci-dessous.

Les *MHEG links* ne sont pas les liens hypermédia (hyperliens) conventionnels mais permettent de spécifier les relations spatiales, temporelles et conditionnelles entre les objets média ainsi que les actions effectuées par ou sur ces objets. Un *MHEG link* consiste en une condition (*LinkCondition*) et un effet (*LinkEffect*) qui est une liste d'actions à exécuter lorsque la *LinkCondition* est vérifiée. Les *MHEG links* avec les deux primitives temporelles binaires : *serial* (qui définit une exécution en séquence entre des objets média) et *parallel* (indiquant une exécution simultanée des objets média) sont utilisés pour spécifier un scénario multimédia. En ce sens, le modèle d'exécution de MHEG est fondé sur la programmation par événements.

Les *objets composites* sont utilisés comme un container pour un groupe de *MHEG objects* formant ainsi un objet complexe. Ils permettent aussi la synchronisation entre des parties d'une présentation multimédia à l'aide de primitives de composition spatiale et temporelle.

MHEG est à l'origine de plusieurs prototypes d'application multimédia comme les projets GLUE, GLASS et MAJA de DeTeBerkom. Ainsi, le projet MAJA consiste en une *Java Applet* dont le rôle est de présenter des applications MHEG en reliant, au moyen de la technologie de communication CORBA, un client de présentation MHEG à un moteur MHEG sur un site serveur. Le principe de fonctionnement est le suivant. GLASS a pour objectif de réaliser un système qui peut offrir différents services comme la vidéo à la demande ou bien la TV interactive. La présentation au sein de ce système est fondée sur la norme MHEG, comme par exemple la spécification de la scène figure [7](#) : la scène se compose d'une image du fond avec plusieurs objets de type texte formant les options d'un menu. À chaque texte, est associé un objet MHEG Link qui décrit son comportement vis à vis de l'interaction d'utilisateur.

```
(Composite:InfoScene

<autres attributs ici...>

group-items:

(bitmap: BgndInfo

content-hook: #bitmapHook

original-box-size: (320 240)

original-position: (0 0)

content-data: referenced-content: "InfoBgnd"

)

(text:

content-hook: #textHook

original-box-size: (280 20)

original-position: (40 50)

content-data: included-content: "1. STAR ..."

)

links:

(link: Link1

event-source: InfoScene

event-type: #UserInput

link-effect: action: transition-to: InfoStar

)

<suite de la spécification ... >
```

)

Figure 7 - Exemple de description d'une scène en MHEG

Comme pour les langages de scripts classiques, les principes de spécification de MHEG lui permettent d'offrir un fort pouvoir d'expression (variété des objets de base traités, composition temporelle, navigation, etc.). Les besoins pour les auteurs sont cependant un peu mieux couverts : les possibilités de structuration et de modularité sont fournies par l'approche objet et la notion d'objets composites.

### 4.1.3 Structures de graphes

Dans l'approche fondée sur les graphes, le document est représenté sous la forme du diagramme de son flot de contrôle. Ce diagramme décrit l'interaction entre les éléments multimédia du document à l'exécution.

L'édition de documents peut ainsi s'appuyer sur une description plus riche et plus facile à utiliser de la synchronisation. Il est aussi possible de représenter beaucoup plus de combinaisons de scénarios que dans le cas des *timelines*. Les graphes fournissent également un support plus adapté pour les opérations d'édition comme la suppression ou l'insertion d'éléments car la disposition des éléments est maintenue sous forme explicite comme dans l'interface de *Firefly* [4].

Il existe deux variantes de ces diagrammes : les graphes plats et les graphes hiérarchisés.

#### Graphes plats

Les graphes plats décrivent la synchronisation entre les instants de début et de fin de l'ensemble des éléments d'un document. Souvent, la synchronisation est décrite au moyen des trois opérateurs d'instant « *avant* », « *après* » et « *en même temps* » comme vus section 3.1.2. Les graphes sont utilisés de deux façons : dans *Firefly* [4], leur utilisation se limite à un cadre informel de synchronisation motivé par des besoins d'interface graphique, la vérification étant réalisée par une algorithmique de gestion de contraintes. Dans *HTSPN* [11], cette utilisation correspond à un véritable support langage pour des modèles temporels formels plus élaborés, ici des réseaux de Pétri.

Bien que le principe en soit simple, l'approche des graphes plats présente un ensemble d'inconvénients :

- La description temporelle d'un document sous forme de graphe plat a tendance à devenir rapidement complexe et très peu lisible dès que sa taille croît. Cette complexité provient d'une représentation événementielle de bas niveau du scénario. En effet, chaque élément fait intervenir plusieurs événements temporels qui correspondent à son début, sa fin et même parfois à des événements internes (numéro d'image vidéo) qui sont mis en relation avec d'autres éléments.
- La représentation du document est très peu structurée ce qui limite la réutilisation de ses parties. Pour les graphes formels, ces inconvénients sont compensés par des mécanismes d'édition plus élaborés permettant, par exemple, la détection des incohérences et la résolution automatique de la synchronisation à partir de spécifications de plus haut niveau, comme dans *HTSPN*.

## Graphes hiérarchisés

L'approche basée sur une structure de graphes hiérarchisés consiste à exploiter l'organisation logique du document pour décrire et mettre en oeuvre sa synchronisation temporelle. Le grand avantage de cette approche est la possibilité d'organiser le document en modules indépendants sur lesquels on peut appliquer des primitives globales de synchronisation. Ces primitives s'appliquent sur des intervalles et sont la mise en parallèle ou en séquence des éléments appartenant à une même entité logique. C'est l'approche adoptée dans *CMIFed* (*CWI Multimedia Interchange Format Editor*) [13] qui constitue l'un des outils les plus représentatifs de ce type d'approche.

Les inconvénients cités pour les graphes plats sont compensés dans les graphes hiérarchisés grâce à des mécanismes d'édition plus élaborés permettant d'encapsuler des parties de document et de les utiliser comme des objets de base.

### SMIL

Récemment, un nouveau standard de documents multimédia, appelé *SMIL* (*Synchronized Multimedia Integration Language*) [18] a été proposé au sein du consortium World Wide Web. Ce standard propose un format de documents qui s'appuie sur une structure d'arbre d'opérateurs temporels.

La syntaxe de *SMIL* est basée sur l'utilisation d'un marquage descriptif. Une spécification *SMIL* est organisée autour de deux parties :

- la partie spatiale, `<layout>` de l'en-tête `<head>`, permet de définir un ensemble de régions sur l'écran qui contiendront l'image graphique des différents objets multimédia affichables.
- La partie temporelle `<body>`, permet de spécifier le scénario du document multimédia. Elle est composée de balises qui représentent les opérateurs temporels comme `<seq>` et `<par>` ainsi que les objets multimedia de base comme `<audio>`, `<video>`, `<text>` et `<img>`. De plus l'opérateur choix `<switch>` permet de sélectionner, moyennant des critères donnés comme la langue, le débit du réseau, etc., un élément parmi un ensemble d'alternatives possibles.

Les objets de base sont accessibles à travers des *urls* et leur durée est soit spécifiée au moyen de l'attribut *dur*, soit lue dans le fichier source du média, soit déterminée par leur mode de composition temporelle avec d'autres objets.

Dans la figure [8](#) nous décrivons la partie *Agence* de notre exemple au format *SMIL*.

```
<smil>
```

```
<head>
```

```
  <layout type="text/smil-basic">
```

```
    <region id="titre" left="4%" top="4%" width="47%" height="22%"/>
```

```
    <region id="image" left="52%" top="5%" width="45%" height="42%"/>
```

```

    <region id="bouton" left="45%" top="95%" width="15%" height="20%"/>

    <region id="video" left="5%" top="28%" width="46%" height="34%"/>

</layout>

</head>

<body>

    <par id="Agence" endsync="last">

        <!-- 3 éléments en parallèle : la séquence Seq_Passé, la vidéo Visite et
le bouton de navigation H1-->

            <seq id="Seq_Passé"> <!-- séquence de Passé, Nom et PressBook -->

                <audio id="Passé" dur="60.0 s" src="http://www.inria.fr/passe.au"/>

                    <text id="Nom" region="titre" dur="30.0 s"
src="http://www.inria.fr/texte.html"/>

                <par>

                    <audio id="Jingle" begin="25s"
src="http://www.inria.fr/jingle.au"/

                </par>

            </seq>

            <video id="Visite" region="video"begin="40s"
src="http://www.inria.fr/visite.mpg"/>

            <a id = "H1" href="#Mascotte" show="replace">

            </a>

        </par>

        <par id="Mascotte">

            <!-- la partie mascotte -->

            .....

        </par>

```

</body>

</smil>

Figure 8 - *Le scénario de l'Agence en SMIL*

Le format SMIL permet l'expression de relations causales comme le *parmin* par le biais d'attributs de synchronisation sur les objets (attributs *begin* et *end*) et sur l'opérateur <par> (attribut *endsync*). De plus, les relations de causalité qui affectent globalement les objets peuvent s'exprimer avec l'élément ancre <a> puisque sa sémantique correspond à la spécification d'une navigation globale. C'est le cas par exemple du lien *HI* défini sous forme d'ancre : <a id = "HI" href="#Mascotte" show="replace">. Cependant, il n'est pas possible de spécifier des interactions locales comme dans le cas de la partie *Mascotte* où l'élément activable *Bouton* n'a qu'un effet local sur l'élément *Message*.

## 4.2 Les approches à base de contraintes

### 4.2.1 Principes

Dans les approches déclaratives avec contraintes, l'auteur spécifie ce qu'il souhaite obtenir comme placement temporel (par exemple, qu'une vidéo soit présentée après un texte et en même temps qu'un commentaire sonore) sans avoir à spécifier toutes les informations temporelles attachées à ces objets (instants de début et de fin, durée). Un scénario est donc défini par un ensemble de relations temporelles (avant, pendant, ...) entre les objets. Par exemple, la partie présentant l'agence dans notre exemple peut se décrire par les contraintes données dans la figure 10.

À partir de cette liste de contraintes, les systèmes auteur basés sur ces approches ont pour tâche d'une part de s'assurer de la cohérence du document : l'existence d'au moins une solution (pour chaque objet, une date de début et une durée) qui respecte toutes les contraintes ; d'autre part de produire statiquement et/ou dynamiquement une (des) solution(s) de placement temporel à partir des spécifications : par analogie aux systèmes d'édition conventionnels, on parle de *formatage temporel*. La figure 9 résume ces différentes étapes. Le document formaté peut alors être présenté à l'auteur/ lecteur dans une *vue de présentation* sur laquelle ce dernier peut agir selon les fonctions d'interaction qui lui sont proposées.

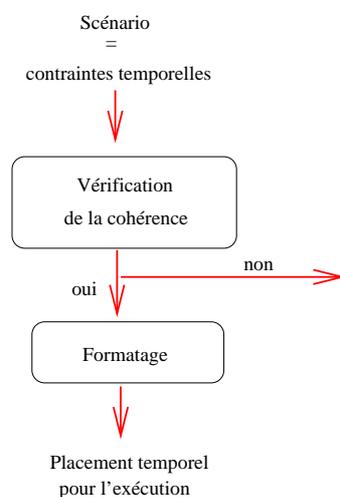


Figure 9 - Les deux étapes principales d'un environnement à base de contraintes



Grâce à cette phase de formatage, les environnements auteur utilisant des contraintes rendent la modification d'un document plus simple. En effet, une modification locale comme le changement de la durée d'un objet, l'ajout ou le retrait d'une relation ou d'un objet sera automatiquement répercutée sur le reste du document, tout en conservant les synchronisations temporelles auparavant exprimées. Par exemple, si l'auteur décide d'ajouter au scénario qui décrit l'*Agence* une musique après la vidéo de telle sorte que cette musique se termine en même temps que le *PressBook*, il doit simplement ajouter deux relations : *Musique suit Visite* et *Musique termine PressBook*. De même, il peut changer la durée de l'historique de l'agence *Passé* sans avoir rien d'autre à modifier. Étant donnée la nature très itérative du processus de création d'un document, cette facilité avec laquelle l'auteur peut modifier son document constitue un avantage certain.

Le second avantage des approches à base de contraintes se situe au niveau du système de présentation d'un document et de sa capacité à respecter les synchronisations temporelles exprimées par l'auteur. En effet, en utilisant une technique de formatage dynamique, c'est-à-dire qui calcule la solution au fur et à mesure de la présentation, il est possible de tenir compte des retards dans la présentation de certains objets. Ces retards peuvent avoir pour origine une limitation de puissance de la machine ou une transmission réseau défaillante.

Les deux avantages cités ci-dessus reposent sur l'existence d'un module de vérification de la cohérence et d'un formateur temporel performants. La difficulté majeure dans les deux cas est la prise en compte des objets incontrôlables.

Il existe actuellement plusieurs environnements auteur basés sur les contraintes, on peut citer Isis [12], Tiempo [14] et Madeus. Nous allons développer dans la suite de cette section l'environnement Madeus de manière à illustrer les avantages inhérents à ces approches.

#### 4.2.2 Le système Madeus

L'objectif du projet Madeus [9], [8] est d'offrir une réponse globale au problème de l'édition et de la présentation de documents multimédia. Nous nous intéressons ici uniquement à la façon dont un auteur spécifie le scénario temporel des documents multimédia.

*Un document Madeus possède une organisation logique décrite sous forme d'une hiérarchie de composants, les feuilles étant des objets de base (texte, image, vidéo, son). Par exemple, le document exemple est composé de 2 scènes *Agence* et *Mascotte*, chaque scène étant elle-même constituée des éléments qu'elle contient (voir partie gauche de la figure 10, seul le contenu de la partie *Mascotte* est développée). A chaque niveau de la hiérarchie, l'auteur peut spécifier des relations temporelles entre tous les objets (de base ou composites) qui partagent le même père. Ceci constitue le scénario temporel du document. Les relations dont l'auteur dispose sont données par les opérateurs de l'algèbre d'Allen, augmentées de l'opérateur *parmin* qui permet d'exprimer des comportements d'interruption. La partie droite de la figure 10 donne le scénario temporel de notre exemple.*

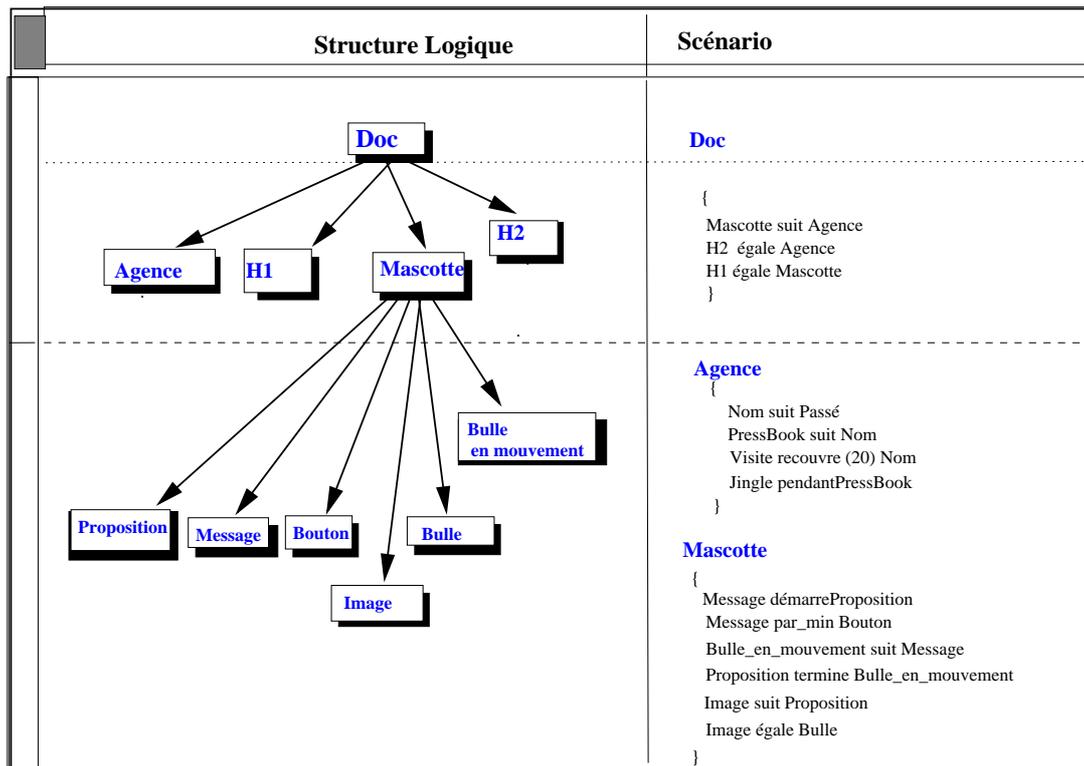


Figure 10 - Scénario temporel de la scène 1

Pour compléter ce scénario et obtenir le comportement du document exemple, il faut rajouter à cette spécification les hyperliens qui permettent de passer d'une partie du document à l'autre. Ceci se fait en associant aux objets *H1* et *H2* un attribut nommé *href* (par analogie avec le format HTML) qui spécifie l'ancre de destination de l'hyperlien, ici respectivement *Mascotte* et *Agence*.

Pour aider l'auteur dans sa tâche de spécification du scénario temporel, Madeus permet de visualiser le scénario à travers une « vue scénario ». Cette vue est inspirée du timeline classique utilisé dans les approches temps absolu, mais elle est adaptée ici à l'utilisation des contraintes (voir la figure [11](#) qui donne la vue associée de la partie *Agence* de l'exemple) :

- Chaque contrainte spécifiée par l'auteur est visualisée : un ressort (resp. deux ressorts) lie deux objets reliés par une relation *avant* (resp. *pendant*), un trait vertical lie les instants de début de deux objets mis en relation par un *démarre*, etc.
- L'auteur peut interagir avec la vue en déplaçant un objet sélectionné sur l'axe temporel : seuls les déplacements autorisés par les contraintes sont possibles ; de plus, les positions des autres objets sont réajustées en temps réel. Cette possibilité de manipulation directe de la vue permet à l'auteur de parcourir l'espace des solutions cohérentes avec son scénario.

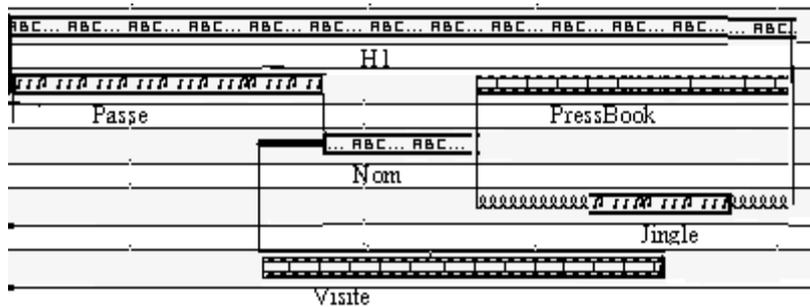


Figure 11 - *Vue scénario de la présentation de l'Agence*

L'environnement d'édition d'un document dans Madeus est fondé sur une extension du principe "Wysiwyg" utilisé par la majorité des éditeurs de texte. Celui-ci ne peut s'appliquer tel quel puisque un document multimédia est par nature dynamique et qu'il n'est du coup plus possible de fusionner la phase d'édition de la phase de présentation. Cependant, comme nous l'avons déjà mentionné il est important pour l'auteur de limiter autant que possible la distance entre ces deux phases. Pour cela, Madeus propose à l'auteur d'utiliser la fenêtre de présentation de son document comme une vue sur laquelle il est possible, après avoir stoppé la présentation du document, de sélectionner des objets pour leur associer une relation temporelle (ou spatiale). Cependant, cette approche n'est pas toujours satisfaisante et d'autres supports d'édition doivent être proposés à l'auteur. Le domaine de l'interface homme-machine pour la spécification des documents multimédia est encore largement ouvert.

## 5 Conclusion

De cette étude de différents outils, langages et environnements de spécification de documents multimédia, il ressort que le domaine est encore à ses balbutiements (lenteur dans l'émergence de standards effectivement acceptés par la profession). Ceci est non seulement dû à la jeunesse du domaine, mais aussi à la complexité des besoins à couvrir.

D'une façon un peu schématique, nous avons pu montrer que les approches de nature opérationnelle permettaient d'offrir un fort pouvoir d'expression tandis que les approches à base de contraintes ont des potentialités plus grandes sur le plan de la mise en oeuvre d'environnements conviviaux pour la création de documents multimédia. En effet, elles offrent un mode d'expression très intuitif et facilitent grandement les opérations de modification d'un document existant. Parmi les approches opérationnelles, les langages déclaratifs à base de structure de graphes comme SMIL offrent un bon compromis entre le pouvoir d'expression, l'indépendance vis-à-vis des plates-formes et des applications et enfin la facilité d'interprétation. Il est aussi à remarquer que, du fait de leur mode de représentation proche de l'exécution, les approches opérationnelles permettent de réaliser des systèmes de présentation plus facilement qu'avec les approches à base de contraintes. C'est ainsi qu'au lieu d'opposer ces approches, il est tout à fait envisageable d'exploiter les atouts de chacune d'elles :

- Utiliser un environnement auteur à base de contraintes pour créer des documents multimédia, puis lorsque leur forme est jugée suffisamment stable, les sauver dans un format comme SMIL ou MHEG.
- Utiliser des navigateurs hypermédia pour jouer ces documents au format SMIL ou MHEG.

La création de titres multimédia reste encore une activité longue et fastidieuse. Les professionnels du domaine considèrent que le temps de conception et de réalisation des documents multimédia se mesure à l'échelle de plusieurs mois et implique des spécialistes différents ayant chacun leurs outils indépendants. Pour baisser les coûts de ces produits, il est nécessaire que les environnements auteur intègrent ces différentes étapes de développement de documents multimédia et soient directement accessibles aux créateurs eux-mêmes. Les environnements actuels sont encore loin d'offrir de tels services, mais la dynamique apportée par l'arrivée de standards comme SMIL (et surtout d'outils les implantant) devrait rapidement y remédier.

## References

### Documents structurés

- [1] ANDRE (J.), FURUTA (R.), QUINT (V.), *Structured documents*, Cambridge University Press, Cambridge, 1989.
- [2] STERN Y., "Les quatre dimensions des documents électroniques", *Document numérique*, vol. 1, num. 1, pp. 55-60, 1997.
- [3] W3C Recommendation, *Extensible Markup Language (XML)*, <http://www.w3.org/TR/1998/REC-xml-1998021>, 10 février 1998.

### Environnement auteur et applications multimédia

- [4] BUCHANAM (C.), ZELLWEGER (P.T.), "Specifying Temporal Behavior in Hypermedia Documents", *Proc. of the ACM Conf. on Hypertext*, pp. 262-271, décembre 1992.
- [5] DRAPEAU (G. D.), "Synchronization in the MAestro Multimedia Authoring Environment", *Proceedings of the First ACM Conference on Multimedia*, pp. 331-340, ACM Press, Anaheim, Californie, août 1993.
- [6] ICONAUTHOR 6.0, *User's guide*, Aim Tech, 1997 [En ligne] : <http://www.aimtech.com/products/iconauthor>.
- [7] JOURDAN (M.), LAYAÏDA (N.), ROISIN (C.), SABRY-ISMAIL (L.), TARDIF (L.), "Madeus, an Authoring Environment for Interactive Multimedia Documents", *6th ACM Multimedia'98*, Bristol, 12-16 septembre 1998.
- [8] JOURDAN (M.), ROISIN (C.), TARDIF (L.), "Édition et Visualisation Interactive de Documents Multimedia", *Electronic Publishing'98*, St Malo, France, avril 98.
- [9] LAYAÏDA (N.), *Madeus: un système d'édition et de présentation de documents structurés multimédia*, Thèse de doctorat, Université Joseph Fourier, juin 1997.
- [10] MACROMEDIA, *Flash et Director*, {En ligne : <http://www.macromedia.com>, 1998.

- [11] 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.
- [12] KIM (M. Y.), SONG (J.), ``Multimedia Documents with Elastic Time'', *Proc. of the 3rd ACM Conf. on Multimedia*, pp. 143-154, San Francisco, novembre 1995.
- [13] VAN ROSSUM (G.), JANSEN (J.), MULLENDER (K.) and BULTERMAN (D.), ``CMIFed : a presentation Environment for Portable Hypermedia Documents'', *Proc. of the ACM Multimedia Conf.*, California, 1993.
- [14] WIRAG (S.), ``Modeling of Adaptable Multimedia Documents'', *Interactive Distributed Multimedia Systems and Telecommunication Services; 4th International Workshop, IDMS'97*, pp. 420-429, Darmstadt, septembre 1997.
- [15] STEMLER (L.K.), ``Educational Characteristics of Multimedia: a Literature Review'', *Journal of Educational Multimedia and Hypermedia*, vol. 6, num. 3/4, pp. 229-359, 1997.

### **Standards multimédia**

- [16] ISO/IEC JTC1/SC18/WG8 N1920, *Information Technology: Hypermedia/Time-based Structuring Language (HyTime), Second edition*, ISO/IEC, août 1997. [En Ligne] : <http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html>.
- [17] MEYER-BOUDNIK (T.) et EFFELSBERG (W.), ``MHEG Explained'', *IEEE Multimedia Magazine*, vol. 2, num. 1, pp. 26-38, 1995.
- [18] W3C Proposed Recommendation, *SMIL: Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*, <http://www.w3.org/TR/PR-smil>, 9 avril 1998.

### **Raisonnement temporel**

- [19] ALLEN (J. F.), ``Maintening Knowledge about Temporal intervals'', *CACM*, vol. 26, num. 11, pp. 832-843, 1983.