

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mentouri – Constantine
Faculté des Sciences de l'Ingénieur
Département d'Informatique

N° Ordre : 116/TS / 2007
N° Série : 06/INF / 2007

Thèse pour Obtenir le Titre de Docteur en Science

**Méthodes et Outils pour la Construction
de Scènes Multimédia Distribuées**

Présentée par :
Ilhem LABED

Devant le jury :

Président : *Professeur SAHNOUN ZAIDI*

Directeur de thèse : *Professeur BOUFAIDA MAHMOUD*

Examineurs :

Professeur BENMOHAMED MOHAMED

Docteur POITRENAUD DENIS

Docteur ZIDANI ABDELMADJID

Soutenue le 08 Décembre 2007

Le multimédia recouvre de nombreuses applications dans des domaines aussi divers que l'enseignement à distance (*e-learning*), l'imagerie médicale, les jeux vidéos ou encore l'art numérique. Pendant longtemps, la plupart des systèmes n'ont intégré du multimédia que la possibilité de rendre des données temporelles, comme du son ou de la vidéo, en ajoutant simplement un module de lecture de ces données. Mais aujourd'hui ce point de vue a changé et une réelle intégration des diverses données mono (et multi) médias est possible.

Au fur et à mesure de l'émergence du domaine multimédia, les applications ont évolué en proposant des fonctionnalités et des modèles intégrant des données multimédias de manière plus structurée et cohérente. La composition des données a été la première problématique importante du multimédia "réel", s'intéressant à la définition d'objets composites à partir d'objets simples. La possibilité de répartir tout ou partie d'un contenu multimédia sur un réseau est apparue avec le *World Wide Web* qui a engendré au fil des années divers modes de distribution du contenu. La facilité avec laquelle accéder à ce contenu augmente un peu plus chaque jour.

En fait, le multimédia part du postulat que toute information est plongée de manière intrinsèque dans le temps et l'espace. Certaines données médias ne dépendent pas du temps ou en dépendent fortement, alors que d'autres n'ont qu'une représentation visuelle ou bien auditive. Mais dans tous les cas, les informations ont une vie dans le temps et l'espace.

Comme premier facteur, le temps a été de tout âge un sujet âpre à la discussion. C'est un facteur critique de notre monde car il n'est pas manipulable par l'être humain ou les machines que celui-ci crée. Il en résulte que c'est un aspect primordial des systèmes multimédias. La spécification temporelle du comportement d'un système ou d'un document multimédia repose sur la garantie qu'elle sera exécutée de manière fidèle selon le "sens" convenu pour cet outil. Cet état de fait a poussé les gens du domaine à proposer aux auteurs de nombreux langages, outils portables et abstraits, afin de définir des paradigmes de création de *scénarios*, ç à d, des relations temporelles entre éléments du document multimédia.

Le second facteur d'importance est celui de l'information spatiale qui est mieux cernée que celle du temps, car sa maîtrise nous est plus naturelle et a fait l'objet de nombreux travaux étalés sur de longues années. C'est aussi le facteur qui véhicule le plus d'informations et est le plus sujet à l'interprétation du spectateur humain (le lecteur), une des raisons pour lesquelles il est peu abordé dans le multimédia et reste l'apanage du domaine du média visuel. La plus grande partie des outils multimédias n'offrent qu'une structuration simple de l'information spatiale, concentrant leurs efforts de gestion essentiellement sur le facteur temps. L'auteur de document multimédia ou l'utilisateur de système multimédia doit donc souvent faire passer les informations spatiales dans les données mono médias.

Dès lors, les standards et les modèles employés pour représenter les documents classiques sont devenus inadaptés pour représenter des documents multimédias et de nouveaux modèles sont alors proposés pour la création, la modification et la présentation de documents multimédias. Cependant, la nature dynamique des éléments manipulés ainsi que la diffusion de leur ordonnancement font que les standards tels que SMIL [Smil 01] ne suffisent pas et rendent plus complexe la réalisation d'outils auteurs assurant la spécification des scénarios temporels ainsi que des modèles de document adéquat permettant la gestion du document à l'étape de conception et plus tard à celle de l'exécution.

Motivation et objectifs

Le travail de cette thèse s'intéresse à la modélisation multimédia. Notre principal objectif est de contribuer à l'édition et la présentation des documents multimédias en utilisant un système auteur qui, par une démarche d'édition proposée à l'auteur ainsi que les différents modèles de documents sur lesquels se base son architecture, permettra à la fois :

- (i) l'acquisition des scénarios relatifs aux documents multimédias à présenter ;
- (ii) la validation des contraintes qui leur sont associées ;
- (iii) ainsi que la restitution de ces documents à leur demande par l'utilisateur,

ceci tout en respectant l'idée de base : "*le confort de l'utilisateur prime*", ç à d, que l'interface d'un tel système doit être d'utilisation facile non seulement pour les professionnels de la programmation mais aussi pour les néophytes et les non initiés.

Ces motivations nous ont conduit à aborder cette thèse avec les objectifs suivants :

- 0 Définir un modèle de document multimédia, décrivant le contenu des médias ainsi que leur synchronisations temporelle et spatiale, et à partir du quel peuvent être effectuées les différentes fonctions d'édition et de présentation. Le modèle que nous proposons est basé sur une approche objet ce qui permet la modularité et la réutilisation. L'auteur peut éditer les différentes parties de la présentation indépendamment et les relier ultérieurement pour composer son scénario ce qui n'est pas pris en charge dans la plupart des standards proposés, tels que SMIL, qui ne proposent aucun moyen pour définir des modules de présentation indépendants. L'absence de cette propriété dans un système n'est pas perceptible pour de simples documents, mais dès que les documents se compliquent, un grand besoin de modularité se fait sentir.
- 0 Définir un modèle de vérification qui permet d'utiliser des descriptions du contenu de médias et de leurs relations de synchronisation afin de contrôler les incohérences temporelles lors de la phase d'édition. Ce modèle doit aussi permettre de valider les scénarios pouvant être modifiés suite à des interactions utilisateurs pouvant survenir à tout moment de l'exécution. Le modèle que nous proposons est une extension de réseaux de Petri qui proposent des bases formelles de détection d'incohérences ;
- 0 Définir un processus d'édition et de présentation qui facilite l'édition incrémentale des documents, leur validation et leur présentation. La génération automatique de présentations multimédias est une fonctionnalité de grande importance. Le système que nous proposons doit être souple et riche. Il doit prendre en compte la connaissance partielle de l'utilisateur et doit permettre la génération automatique de présentations en prenant en compte les indications de l'utilisateur qui peut ne pas maîtriser les interfaces compliquées et qui peut intervenir au moment de la *présentation* du document ;
- 0 Définir une nouvelle architecture pour un environnement auteur qui puisse offrir le moyen d'utiliser les points précédents afin de gérer et de présenter les documents résultants.

Les objectifs de cette thèse exposés ci-dessus se situent clairement dans un contexte applicatif qui tient compte, en premier lieu, des besoins des utilisateurs qu'il ambitionne de contenter.

Plan de la thèse

Après une introduction, ce mémoire est organisé en deux parties chacune d'elles constituée de trois chapitres.

Partie I : Constitue l'état de l'art à partir du quel nous introduisons le contexte des documents multimédias.

Chapitre 1 : Il dégage les notions de base et les grandes lignes qui caractérisent les documents multimédias. Cette étude aborde dans une première partie les caractéristiques d'un composant média alors que la deuxième partie présente les fonctions de base qui doivent être fournies pour une présentation multimédia afin de satisfaire les besoins identifiés et la qualité requise.

Chapitre 2 : Ce chapitre est consacré aux deux dimensions : spatiale et temporelle, caractérisant tout document multimédia. Il présente entre autres quelques modèles utilisés pour la modélisation de ces dimensions tout en mettant en évidence les apports et les points faibles de chacun d'eux.

Chapitre 3 : Nous présentons au cours de ce chapitre une étude comparative des standards (produits commerciaux et de recherche) les plus utilisés dans le domaine de la spécification et de la présentation des documents multimédias.

Partie II : Constitue le travail que nous avons réalisé afin d'aboutir à un système auteur d'édition et de gestion de documents multimédia : Lumière.

Chapitre 4 : Le modèle de document noyau de ce travail est présenté au cours de ce chapitre (avec un exemple illustratif). Ce modèle (MDO2) s'apparente à ceux proposés dans un contexte orienté objet ce qui permet, entre autres avantage de l'approche, l'apparition d'un média plusieurs fois dans un document sans grands efforts pour gérer la situation.

Chapitre 5 : Il décrit le modèle (une extension de réseaux de Petri) proposé pour la validation des scénarios multimédias. Une introduction au modèle des *réseaux de Petri*

avec arc de lecture précède la présentation de l'extension proposée afin de supporter les caractéristiques des documents multimédias. Nous exposons aussi les techniques et algorithmes qui sont associés au modèle pour la vérification et la validation des contraintes temporelles établies entre les médias constituant un document et que nous appliquons sur un exemple.

Chapitre 6 : Ce chapitre présente les éléments de bases proposés dans le cadre du système Lumière que nous proposons pour l'édition et la gestion des documents multimédias. Nous nous intéressons au cycle de vie d'un document ainsi que l'architecture du système et la mise en œuvre de ses différents composants en particulier l'interface proposée à l'utilisateur du système.

La conclusion de ce mémoire résume les points essentiels du travail réalisé et présente quelques perspectives de recherche suggérées par le bilan de ce travail.

Au sens générique, le terme "*multimédia*" se rapporte à une communication à travers plusieurs types de média. Un ensemble bien ordonné de médias, dans un document multimédia, permet une communication plus intéressante et plus dynamique. Cette dernière propriété permet à l'information d'atteindre les destinataires plus efficacement et de mieux capter l'attention des interlocuteurs. Parce qu'il a acquis une telle efficacité, le multimédia est utilisé dans un grand nombre de services de notre vie. En effet, nous pouvons trouver de l'information multimédia partout dans les sites Web, dans des outils de formation assistée par ordinateur, dans des kiosques interactifs, dans des systèmes d'accès à des manuels techniques, dans les jeux vidéo, dans l'annonce d'un nouveau produit, sur un cdrom ou un DVD de logiciel, etc.

Ce déploiement d'applications, reposant sur les documents multimédias, est rendu possible par les évolutions technologiques comme la capacité et la puissance des ordinateurs, les possibilités de communication rapides (supports rapides, protocoles de bas et de haut niveaux comme ceux de l'Internet), les périphériques adaptés au multimédia (la carte vidéo, la carte son, le cdrom, et le DVD). Les couches logicielles elles aussi évoluent pour faciliter le déploiement du multimédia, que ce soit dans les domaines du système d'exploitation (Linux, Windows NT/2000/XP), des langages de programmation (C/C++, Java, Python, etc.) et des langages de structuration de l'information (XML). Mais qu'est donc au juste le multimédia ?

Cette partie s'intéresse à la présentation du contexte dans lequel sont apparus les documents multimédias ce qui permet alors de fixer la terminologie employée dans ce travail. Elle fait aussi le tour des différents aspects de la conception et la gestion des documents multimédias. Il s'agit d'en dégager les caractéristiques générales et de présenter les méthodes et modèles couramment employés dans le domaine multimédia. Le dernier chapitre termine la première partie par la présentation de systèmes dédiés pertinents vis-à-vis des problématiques présentées.

Le domaine du multimédia est très vaste, tant dans ses applications que dans ses études théoriques comme en témoignent les travaux présentés dans [Buford 94, Lu 96, Steinmetz 95]. Il comprend et étend les problématiques de domaines aussi divers que les réseaux [Diaz 94], les bases de données [Martin 97], l'édition de documents [Hardman 93b], les interface homme-machine [Laird 02], la composition musicale [Hanappe 99] et les systèmes d'exploitation [Bulterman 91]. Il rejoint même en certains points des domaines hors le contexte informatique comme la philosophie [Tooley 95] ou l'enseignement [Quentin-Baxter 98]. Comme on peut l'imaginer alors, de nombreuses notions sont à définir afin de bien mettre en évidence les différentes caractéristiques du domaine.

1 MEDIAS

Nous présentons ici les divers médias usuels dans le domaine informatique. Après une définition générale nous les décrivons les uns après les autres tout en indiquant différents critères selon lesquels les considérer.

1.1 Définition d'un média

La notion de medium signifie originellement en latin '*milieu, centre*' mais aussi '*lieu accessible à tous, à la disposition de tous, exposé aux regards de tous*' [Gaffiot 02]. Le mot prend plus tard le sens de l'intermédiaire et de moyen de communication de la pensée [Maubourguet 91]. Il transite ensuite par le mot anglophone *mass media* qui désigne aujourd'hui l'ensemble des moyens d'information par voie papier, radiophonique et télévisée. Il est finalement raccourci en média. Cette dernière acception est généralisée en informatique en considérant un média comme un moyen de transmettre, stocker ou présenter des informations [Lecomte 00]. Au sein de ces informations, peu sont compréhensibles par un être humain à cause de leur trop bas niveau de granularité (bit, trame réseau, page physique).

Le multimédia s'intéresse aux médias véhiculant des informations conceptuellement accessibles par un utilisateur humain, c'est-à-dire à travers ses cinq sens d'observation. Ces

médias basiques sont ensuite étendus à divers types d'informations qui possèdent des caractéristiques communes. La classification classique en multimédia regroupe les types de média suivant: texte, image, son, animation et vidéo. Nous ajoutons à ceux-là le média programme qui englobe les données qui précisent des traitements d'autres données médias. Ne sont pas inclus dans cette classification les médias tactile, olfactif et gustatif, car ils sont de rare utilisation dans le monde informatique.

Les données se référant à un média sont appelées mono média. Chaque média est l'objet d'études particulières visant à faire avancer la compréhension et la maîtrise du média et de ses données. La représentation des données mono média, dans un format informatique compréhensible par une machine, est appelée codage. Des efforts considérables sont faits afin de définir des codages adéquats pour chaque média afin de faciliter certaines opérations [Lecomte 00]. En effet, les données sont inséparables de leurs traitements au sein des systèmes informatiques et des choix doivent être faits quant à ceux qui seront privilégiés.

1.2 Principaux médias

Différents types de médias sont considérés dans la littérature des documents électroniques et où l'on associe à chaque type des propriétés et un comportement spécifiques.

1.2.1 Texte

C'est le média artificiel le plus ancien au monde et celui qui a fait l'objet des premiers développements en informatique. Ses informations sont conceptuellement bien intégrées dans nos modèles courants, ce qui les rend simples à modéliser au sein des systèmes informatiques. Un texte basique est une séquence de caractères d'un alphabet. L'encodage des caractères a été un problème pendant les débuts de l'informatique personnelle, mais ne pose plus d'insurmontables problèmes aujourd'hui grâce à des standards répandus, comme l'ASCII, l'ISO-8859 et l'UNICODE. Un texte est traditionnellement découpé en mots, phrases, paragraphes, sections et chapitres au sein d'une oeuvre littéraire comme un article ou un livre. Un texte enrichi ajoute des informations de mise en forme en associant à chaque caractère des paramètres de présentation, comme la police de caractère (groupe de caractères de caractéristiques de formes identiques), la casse, le gras, l'italique, le soulignement ou bien

encore la couleur. Lorsque le texte est mis en page, chaque caractère possède un alignement relatif aux caractéristiques de la page (tailles, marges, etc.).

Le texte est un flot de caractères qui possède une direction intrinsèque. Les exercices de style comme les palindromes, les calligrammes de G. Apollinaire ou les ASCII art (textes qui inspirent des dessins) ne modifient en rien cette propriété mais jouent sur des effets de forme. Cette linéarité horizontale de sens gauche à droite en occident est différente des écritures arabes qui partent de droite à gauche et encore plus différente des écritures asiatiques, en particulier pour les kanji ou idéogrammes dont l'organisation est fortement spatiale. La figure 1.1 regroupe les quatre principaux types de textes utilisés dans les documents électroniques.

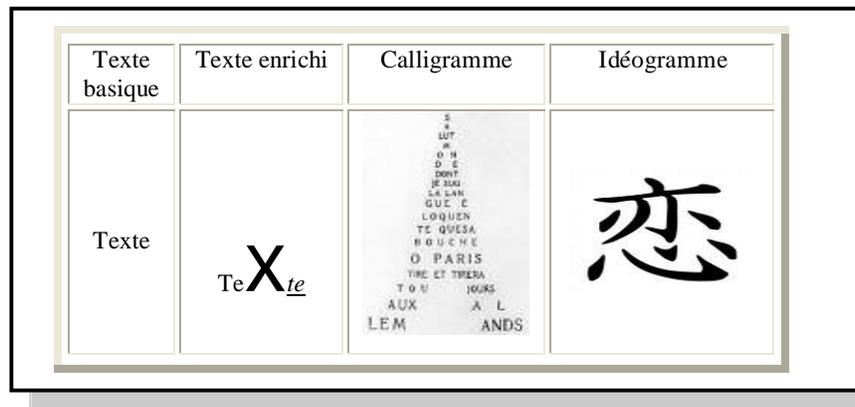


Figure 1.1 – Catégories de textes

La linéarité du texte a été d'une certaine manière brisée par l'introduction de liens hypertextes, rendu célèbre par le web et le langage HTML, permettant au lecteur de passer d'un morceau du texte à un autre en activant le lien.

Bien que moins dense que les autres médias, le texte reste encore aujourd'hui l'élément privilégié pour exprimer le sens et expliciter la signification des autres médias grâce à son intuition et sa simplicité [Flanck 02]. Sa linéarité reste toutefois une limitation qui rend nécessaire l'usage des autres médias pour créer des contenus riches et denses.

1.2.2 Image

Les images sont des entités bidimensionnelles (ou 2D) composées de points, ou pixels, et d'une taille, en nombre de pixels. Chaque pixel possède une couleur et éventuellement une transparence. Les couleurs possèdent des codages propres, comme le triplet RVB de proportions de rouge, vert et bleu ou encore un code parmi une palette de taille entre deux

(bicolore codé sur un bit) et seize millions (true color codé sur 24 bits). Cette définition fournit la représentation bas niveau du médium par le système informatique. Un être humain percevra par contre des courbes et des formes, groupes de pixels de couleurs et de positions proches, seules interprétables par lui.

Le domaine de la génération d'image, ou infographie, s'est grandement développé et a produit des outils puissants comme *Adobe Photoshop*. De nombreux formats binaires d'image ont été standardisés, comme *TIFF*, *BMP*, *JPEG*, *GIF* ou *PNG*. Ils permettent de stocker des informations sur les pixels de l'image et les échanger entre systèmes informatiques, chacun facilitant certaines opérations comme l'affichage progressif ou bien la compression. Suivant la mouvance XML, des formats textuels d'image sont apparus, comme le standard *SVG* (Scalable Vector Graphics).

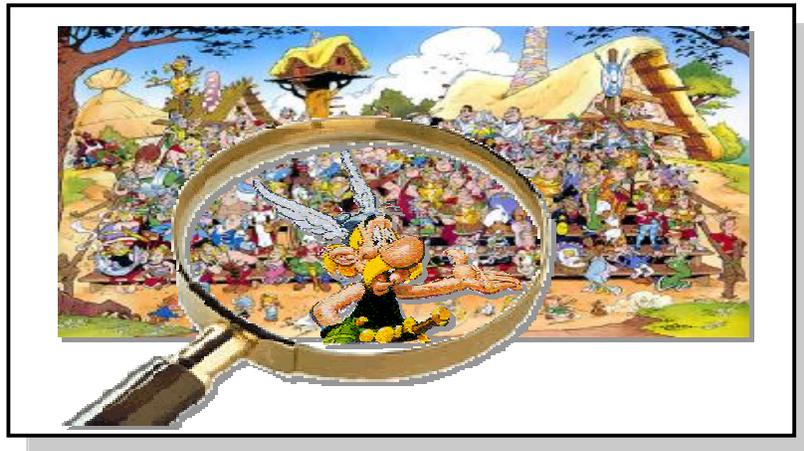


Figure. 1. 2 – Superposition de "layers" constituant une vue

Les formats d'image continuent d'évoluer en intégrant des techniques issues d'autres domaines. Par exemple les *smart graphics* [Edwards 02] peuvent découvrir et s'adapter à des conditions particulières, comme les tâches pour lesquelles elles seront utilisées ou les paramètres réseaux de transmission des images. La représentation de bas niveau d'une image ne peut avoir que deux dimensions, on parlera d'image simple, alors que celle de plus haut niveau peut en avoir plus. Une image peut aussi être une superposition de plusieurs images simples, en jouant sur la transparence et l'alignement afin de préciser leurs positions relatives comme pour la composition des trois dessins (village gaulois, Astérix et loupe) qui donne en résultat la figure 1.2. Dans cette figure nous avons l'impression d'avoir effectué un zoom sur une partie de la toile de fond de façon à agrandir le dessin du guerrier ce qui n'est qu'une

illusion créée par le processus de superposition. Cette approche est souvent utilisée afin de définir des assemblages de divers couches (layers) empilées les unes au-dessus des autres, en laissant à la charge d'un système le calcul de l'image finale qui en résulte.

Les images représentent la plupart du temps des informations en trois dimensions, ou 3D, que ce soient des images réelles (photographies) ou bien de synthèse, c'est-à-dire générées par ordinateur. Dans cette optique 3D, les images sont plutôt conçues comme des assemblages complexes de formes correspondant à des objets réels. Le domaine des images de synthèse permet de pousser très loin le détail des images créées, jusqu'à une qualité photo-réaliste extrêmement proche de la réalité, grâce à des outils de modélisation physique. Des paramètres comme l'illumination ou les forces de contact peuvent être spécifiées dans la définition d'une scène dont le rendu sera calculé par un logiciel comme le raytracer *POV* (Persistence Of Vision).

1.2.3 Son

Le son est l'un des deux principaux médias utilisés aujourd'hui, avec la vidéo. C'est un média temporel dont l'information est représentée par un signal périodique et continu. C'est ce signal qui est émis des périphériques comme les haut-parleurs et perçu par l'oreille de l'utilisateur humain. Le son étant une donnée temporelle qui peut évoluer vite, il requiert une grande quantité d'informations. Les formats sonores fournissent des méthodes de compression qui sont générales pour les représentations continues, qui offrent déjà un gain par rapport aux signaux physiques, ou spécifiques dans le cas discret, comme le modèle psycho acoustique du format *MPEG-1, Layer 3* ou aussi *MP3*. Leurs implémentations définissent des algorithmes de compression et sont des composants logiciels appelés Codecs. D'autre part, le codage relatif des informations indique les valeurs en terme d'écart par rapport aux précédentes valeurs et permet, lorsqu'il est possible, un gain additionnel. Les codages facilitent plus ou moins certains traitements des informations, parmi lesquels la décompression ou l'accès non séquentiel [Hardman 93a]. Tout comme l'image peut posséder plusieurs dimensions, le son peut constituer une dimension supplémentaire dans certains contextes, comme on le voit nettement dans certains jeux vidéo.

1.2.4 Vidéo

Le média vidéo est probablement le plus célèbre aujourd'hui dans le monde du multimédia et celui qui, pour certains, justifie de s'y investir. Cette importance provient aussi

de celle de médias comme la télévision ou le cinéma qui fournissent un point d'entrée simple dans le monde du multimédia dans notre société actuelle.

Ce média n'en reste pas moins un parmi les autres et repose sur des notions et des traitements quasi-exclusivement issus des médias animation et son, qu'il unit en une seule entité. Bien que n'étant pas atomique, car composée à partir de deux autres médias, la vidéo est un média classique que la plus grande partie des systèmes multimédias traitent. La représentation basique d'une vidéo comme une animation et un son joués indépendamment et en parallèle est relativement rare aujourd'hui. La vidéo requiert en effet une synchronisation fine qui oblige à regrouper ensemble les échantillons sonores et les trames d'animation qui se correspondent. On appelle aussi cette synchronisation couplage ou lip-synch, en référence à la synchronisation des lèvres avec les paroles leur correspondant. Les formats vidéos, comme *Windows Avi* (Audio Video Interleave) ou *Apple Quicktime mov*, sont la plupart du temps des conteneurs dans lesquels sont indiqués les codecs audio et vidéo des données médias suivi de la séquence des regroupements d'informations audio et vidéo compressées. La technique de streaming, ou diffusion en flux, permet de diffuser les vidéos sur les réseaux par paquets d'informations. Elle nécessite des formats de codage adaptatifs qui permettent de s'adapter aux variations de paramètres des réseaux.

Les applications qui utilisent cette technique de communication, comme celle de vidéo à la demande ou de vidéo conférence, sont prisées et sont toujours source d'intérêt pour les grandes sociétés [Dimitrova 99]. Les techniques de codage de l'animation par formes ou objets ont ouvert la voie à des augmentations de la vidéo, comme le système de la société Arts Video Interactive [Artsvideo] a pu le prouver. Ce format permet de définir des hyperliens entre objets, le clic de l'utilisateur sur l'un d'eux lançant l'affichage de la vidéo à l'endroit où se trouve le second objet. Les hyper-vidéos sont un type d'objet multimédia récent, à l'intersection entre le multimédia passif et l'hypertexte comme on peut en faire l'expérience sur le web [Grigoras 01]. D'autres développements et études sont en cours afin de faire évoluer les vidéos, notamment dans le cadre des portails vidéo [Dimitrova 99].

1.2.5 Programme

Ce média n'est pas encore très conventionnel en tant que média bien qu'une grande partie de l'activité informatique consiste à créer des programmes et qu'aujourd'hui des programmes reçus comme données permettent de modifier d'autres programmes. Bien qu'imperceptible

aux sens humains, ce média convoie de l'information entre systèmes informatiques, qui sont en quelque sorte les entités les percevant.

L'idée n'est pas nouvelle [Keramane 96], mais sa réalisation et son intégration au sein d'autres systèmes n'a pas encore aboutie. Le standard multimédia MHEG [Vieira 97] fut un des premiers à lancer l'idée avec la classe abstraite *Program*. Cette classe permet d'indiquer les programmes externes au système d'exécution MHEG, généralement des morceaux de codes procéduraux qui exécutent des actions spécifiques.

Ce média se trouve à la frontière entre le paradigme du document, où l'on indique des informations passives traitées par un système, et celui de l'application, où l'on précise à la fois les données et leurs traitements. Les programmes sont des données actives, dans le sens où elles ne seront résolues que par leur exécution. Par exemple, le résultat d'une requête sur une base de données ne peut être connu qu'à l'issue de son exécution par le système de gestion de la base de données [Adiba 99], la résolution de la requête fournissant l'objet multimédia que la requête définit. Cette notion n'est pas à confondre avec celle de dynamique temporelle qui indique le fait qu'une entité possède des instants de début et de fin ou une durée qui ne seront connues que lors de l'exécution, bien que ce que représentent ces données soit connu statiquement avant l'exécution.

On retrouve ce concept de programme comme donnée mono média dans le langage Java de Sun Microsystems sous la forme de sérialisation du code. Cette fonctionnalité des objets Java permet de les stocker sous une forme particulière qui permettra leur transfert vers un autre ordinateur via le réseau ou bien leur récupération lors d'une exécution ultérieure du programme.

Les objets actifs, utilisés par exemple dans le système PREMO [Herman 94], définissent un paradigme où il est possible d'indiquer des données programmes. Aujourd'hui, l'exemple typique de donnée programme se présente dans les composants logiciels appelés players qui permettent de présenter des formats spécifiques de données, l'association entre player et format étant généralement faite par le système d'exploitation. On les retrouve dans le format de données multimédias Hotmedia d'IBM [Kumar 99], ou bien des plugins des navigateurs Internet. L'idée de déclarer les programmes comme des données, permet d'abstraire certains traitements en les déclarant génériques dans une donnée programme qui sera instanciée à

l'exécution. Au delà même réside l'idée des programmes évolutifs, modifiés par des données programmes qu'ils reçoivent et vont les modifier.

Cette approche ouvre de nouvelles perspectives dans de nombreux domaines, comme dans l'infographie où les "smart graphics" [Edwards 02] permettent de créer des images s'adaptant dynamiquement aux systèmes et aux utilisateurs auxquels ils seraient présentés. Elle nécessite encore des efforts intense, tant au niveau pratique vis-à-vis de la portabilité des divers systèmes qu'au niveau théorique à travers une interface de haut niveau capable de décrire la plus grande partie des programmes et des problèmes liés au code mobile [Cardelli 00].

1.3 Typologie des médias

La diversité des médias et de leurs caractéristiques nécessite d'en dresser une carte qui permet de saisir leur organisation générale. La classification se fait au niveau des informations même ou bien de leurs représentations. De nombreux critères peuvent être pris en compte afin de les classer, du plus technique au niveau des bits au plus conceptuels au niveau de leur interprétation par l'utilisateur.

Nous donnons des critères parmi les plus utilisés dans le domaine du multimédia et qui sont des critères naturels, et non techniques, de classification des médias. La typologie en résultant est suffisante pour décrire la plupart des classes de médias utilisées aujourd'hui.

Le premier critère est la part qu'a le média dans l'interaction avec l'utilisateur, ç à d, son importance. Cette part est en fait proportionnelle à la quantité d'information véhiculée par le média. Ce critère revient donc à classer les médias selon leur taille et ce, indépendamment d'un format de représentation particulier, ce qui donne l'ordre approché suivant, du plus important au moins important: visuel, auditif, tactile [Ohba 98] avec une proportion de 70-80% pour le média visuel [Lo Presti 02].

Une autre classification des médias est la présence de leur caractéristique temporelle ou non. Les données d'un média non temporel ne dépendent pas du temps, comme pour le texte ou les images, et sont parfois appelées discrètes ou statiques. Celles d'un média temporel en dépendent intrinsèquement, c'est-à-dire que ses données ont une durée comme l'audio ou l'animation, elles sont nommées continues. Une autre définition [Owen 98] des termes

continus et discrets sépare respectivement les médias en analogiques et numériques. De plus, la représentation des médias temporels peut être discrète, si elle utilise des notations entières, ou continues, si elle utilise des fonctions continues du temps. Pour éviter toute confusion, nous parlerons de médias non temporels et temporels plutôt que discrets et continus.

Classification 1	<u>Importance 1</u> : Visuels (tous les médias sauf Audio)
	<u>Importance 2</u> : Auditif (Audio, Vidéo)
Classification 2	<u>Temporels</u> : Audio, Vidéo
	<u>Non temporels</u> : Image, Texte, Graphique
Classification 3	<u>Naturels</u> : Numérisation d'objets réels
	<u>Synthétiques</u> : objets générés ou retouchés
Classification 4	<u>Linéaires</u> : présentation depuis début
	<u>Non Linéaires</u> : Accès par traitement

Figure. 1. 3 – Classifications des Médias

Une autre utilisation du terme de numérique est faite en opposition au terme de naturel et fait référence au moyen avec lequel les données médias ont été obtenues. Lorsque ces données proviennent d'une numérisation d'un objet réel, on parle de média naturel. Lorsqu'elles sont générées par des programmes informatiques, éventuellement en retouchant des données de médias naturels, les médias sont dits numériques ou encore synthétiques.

Une autre caractéristique des données médias est leur linéarité [Ayars 01]. Les données d'un média linéaire ne peuvent être produites que depuis son début naturel et en suivant son déroulement normal. Cela signifie qu'il est impossible d'accéder à n'importe quelle partie des données d'un média linéaire sans attendre que cette partie ne soit naturellement présentée (ou bien de redémarrer la présentation de ces données depuis le début si la partie se situe dans le passé). Les données du média programme sont linéaires, alors que la plupart des autres données médias sont non linéaires, trivialement pour les médias non temporels. La non linéarité requiert parfois des traitements complexes afin d'accéder à la portion désirée du

média, comme par exemple pour une vidéo MPEG dont les trames peuvent être interdépendantes et nécessiter de recalculer des trames antérieures à celle voulue.

La notion de réversibilité est liée à celle de linéarité et indique s'il est possible de présenter les données médias selon leur ordre temporel inverse. C'est une caractéristique peu examinée et très peu de données de médias temporels possèdent aujourd'hui cette propriété.

Les classifications proposées sont résumées dans le tableau de la figure 1. 3 et dans tout cela nous dirons qu'il est rare qu'un média vive en solo. En effet, des compositions d'éléments de différentes natures dans des documents, dits multimédias, sont les plus généralement utilisées afin de présenter, stocker ou même traiter les informations manipulées dans les différents domaines.

2 DOCUMENTS MULTIMEDIAS

Parmi les notions qui se dégagent des divers systèmes multimédias, la notion de *Document MultiMédia (DMM)* occupe une place importante. Elle joue le rôle de point pivot autour duquel se rattachent diverses problématiques et sert de point de référence dans les systèmes multimédias.

2.1 Définition

Dans la littérature, plusieurs définitions ont été proposées pour l'expression "*document multimédia*" (interchangeable, pour certains auteurs, avec *présentation multimédia* et même *application multimédia* pour d'autres). La définition de référence a été donnée dans [Blakowski 96], où *un document est qualifié de multimédia s'il supporte la présentation intégrée de plusieurs médias dont au moins un est de nature temporisée.*

La diversité des définitions [Bulterman 05, Roisin 99, Sampaio 03] n'empêche en aucun cas qu'elles se complètent les unes les autres. En effet, un *document multimédia* est toujours décrit comme étant un assemblage, selon un scénario spécifié, de médias traditionnels dits statiques (texte, graphique), tout aussi bien que des médias continus (animation, audio et vidéo), des médias structurés (documents HTML, SMIL, SVG), et même des programmes

comme des applets ou scripts. Un bon scénario est celui qui se caractérise par une bonne combinaison de plusieurs éléments multimédias pouvant donner un meilleur résultat de présentation des informations.

Par exemple, une présentation qui intègre en même temps une vidéo avec une audio et des sous titres de la vidéo ; est souvent plus efficace que de présenter les trois éléments indépendants. La composition de ces éléments fournit la plupart du temps une abstraction des mécanismes de la mise en oeuvre de sa sémantique et peut donc être interprétée par une variété d'applications différentes.

2.2 Processus de production

De par l'importance des documents multimédias, des études de processus de production de documents multimédias sont largement décrites dans la littérature aussi bien dans le domaine de la recherche [Bailey 01a, Bailey 01b] que dans le domaine industriel [Tran_Thuong 03]. Les processus présentés peuvent différer dans l'allure générale, mais néanmoins on y retrouve un certain nombre de phases clés établies selon les besoins et exigences des utilisateurs [Bollard 99]. Certaines de ces phases sont complexes et développées en fonction du contexte de conception. Par exemple, si nous faisons le parallèle avec l'écriture de documents HTML, dans un contexte familial l'auteur crée empiriquement la structure de son document alors que dans le cas d'un site professionnel cette structure est préalablement formalisée [Tardif 00]. Plus précisément, ce processus de conception peut être décomposé en six phases principales :

- 1) *Apparition d'une idée pour un document multimédia* suite à un besoin utilisateur.
- 2) *Ecriture du scénario (Outlining) et du scénarimage (story-board)* qui permet d'identifier, de manière visuelle, les *composants* à faire apparaître dans le document, l'interactivité associée à chacun d'eux, ainsi que les *relations* qui pourraient exister entre eux, en bref, identifier la logique du produit. A cette étape, soit que l'auteur possède initialement l'ensemble des données médias brutes, qu'il doit éventuellement retravailler afin de pouvoir passer à la phase suivante, soit qu'il essaie d'identifier leur présence sur un support de stockage local ou à distance. Cela permet de récupérer les propriétés des composants à utiliser ultérieurement, telles que le type, la durée associée, dans le cas des médias dynamiques, etc.

Une sous étape de scénarimage est la conception de l'interface (écrans graphiques attendus). Parfois, il existe diverses solutions d'interfaces pour un écran. Des modèles de ces interfaces peuvent être dessinés de façon plus détaillée pour mieux les percevoir et les comparer.

3) *Réalisation de la maquette* : à la fin de l'écriture du scénarimage, une maquette est réalisée afin de tester un scénario et d'avoir une première idée du document dans sa forme finale. À partir de cette maquette, les choix faits précédemment peuvent être remis en cause. Il est à noter que, dans un contexte commercial, la réalisation de cette maquette est laissée aux soins des informaticiens et nécessite une description très précise du scénarimage et du scénario.

4) *Production du document* où une fois les choix faits, il ne reste plus qu'à produire le document. Cette phase se décompose en deux sous tâches : *Création* ou collecte des médias qui composeront le document, cette étape est généralement réalisée par des graphistes et *Assemblage* des différents médias, selon le scénario donné dans les étapes de conception, pour constituer le document.

5) *Stockage, indexation, et transfert du document* qui doit être enregistré de façon permanente si l'auteur désire le rejouer ou le distribuer. Selon ses besoins, l'auteur aura le choix de stocker son document multimédia dans un fichier intégrant les données ou bien de n'y faire que référence à celles-ci. Il peut ensuite indexer sa présentation multimédia afin d'utiliser les services d'une base de données et distribuer son oeuvre à travers les réseaux informatiques s'il le désire.

6) *Exécution ou restitution* du document multimédia sur son ordinateur. Par souci de clarté, nous parlerons d'exécution d'une présentation multimédia. La présentation multimédia peut être dans un format directement exécutable ou bien nécessiter un système d'exécution particulier selon son format. L'exécution peut nécessiter le transfert de données, via un réseau, et l'interrogation d'une base de données.

Au cours de cette thèse nous nous intéresserons essentiellement aux trois dernières étapes (4, 5, 6) de ce processus, c'est-à-dire à la production du document proprement dite via un environnement d'édition et présentation de documents multimédias. Cet environnement devra rendre accessible la réalisation de la maquette (étape 3) à des non informaticiens et plus particulièrement aux personnes qui écrivent le scénarimage ainsi que d'aider les auteurs à mettre en scène un scénario issu des étapes de conception et donc à intégrer les médias

élémentaires dans un document. Nous n'oublierons cependant pas, qu'au cours de cette production, des remises en cause du scénario ou du scénarimage peuvent arriver à tout moment.

2.3 Dimensions des documents multimédias

En fait, la présentation des documents électroniques à l'utilisateur a jusqu'ici été effectuée principalement en fonction de leur *structure sémantique* (hypertextuelle), de leur *structure logique* et de leur *structure spatiale*.

La structure hypertextuelle correspond à l'organisation des données dans un réseau de nœuds connectés par des liens qui définissent des relations sémantique entre les nœuds. L'ancre, notion introduite par le modèle de référence hypertexte Dexter [Halasz 90], permet de spécifier la partie de l'information qui sera source ou destination d'un lien. Ainsi quand un utilisateur clique sur une ancre source, le lien est suivi causant la présentation de la partie destination. Cette structure est connue par tous ceux qui ont surfé, ne serait ce qu'une fois, sur le net.

La spécification de l'organisation logique correspond à la décomposition d'un système d'information en composants. Ainsi, l'auteur détermine le sujet à présenter et commence à définir les différents éléments et la façon selon laquelle ils seront synchronisés.

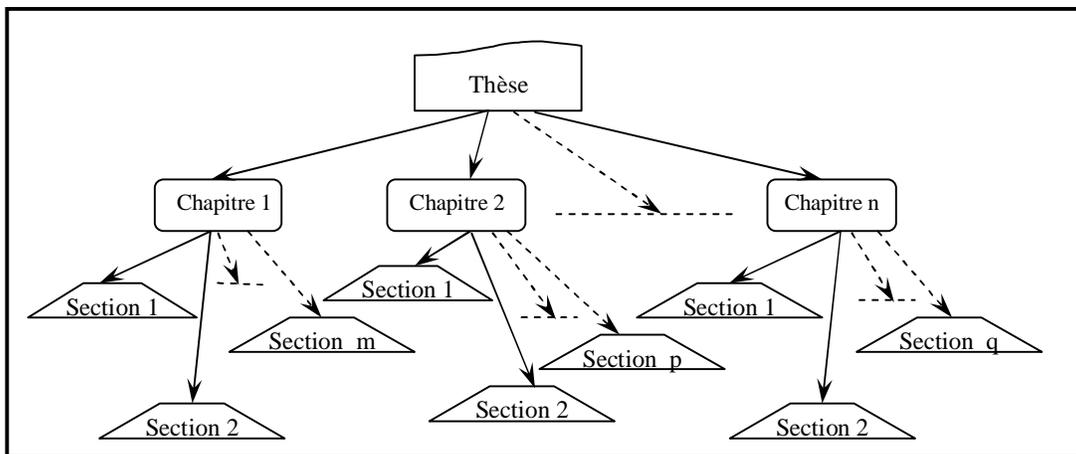


Figure 1. 4 – Organisation logique d'un document "thèse"

Par exemple, un sujet comme la construction d'une thèse peut être décomposé en n composants, représentant chacun un chapitre, en commençant par le premier chapitre. L'opération de décomposition étant récursive, chaque composant peut faire l'objet d'une nouvelle décomposition comme dans le cas des chapitres eux-mêmes décomposés en sections, etc., (voir figure 1.4). L'opération de décomposition se termine lorsque on arrive aux objets médias de base (comme un texte, une image, une vidéo, une audio, etc.). Le résultat de cette opération donne une structure logique hiérarchique pour la présentation.

La spécification de l'organisation spatiale définit, quant à elle, le placement, sur écran, des objets médias les uns par rapport les autres pendant la présentation [Carcone 97]. En réalité, un média dans un scénario occupe un espace (fenêtre), spécifié par les coordonnées du coin supérieur gauche et de la taille de cette fenêtre (hauteur, largeur). Il peut, ou non, partager cet espace avec un (ou plusieurs) média(s) selon l'espace d'intersection entre sa fenêtre et celle(s) de son (ses) voisin(s) comme présenté dans la figure suivante.

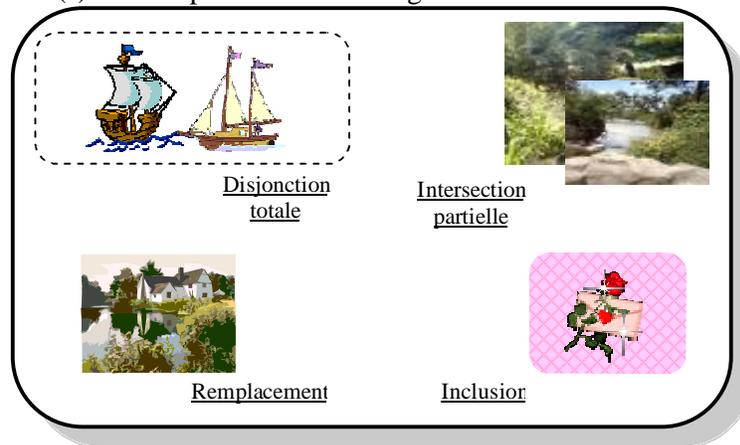


Figure 1. 5 – Formes de co-existence spatiales ente objets

Un nouveau type de structure est maintenant considéré : la *structure temporelle* qui décrit l'enchaînement des éléments dans le temps dont certains ont eux-mêmes une dimension temporelle (vidéo, audio). La spécification de l'organisation temporelle définit l'ordonnancement temporel de la présentation des différents composants afin de construire le scénario souhaité. Cette spécification est développée dans la section suivante consacrée à la synchronisation. De par la complexité de la tâche de spécification temporelle, il est nécessaire d'avoir un mécanisme efficace pour structurer les scénarios temporels. De plus, ce mécanisme doit pouvoir être supporté par un ensemble d'algorithmes pour vérifier la cohérence du scénario spécifié.

En résumé, un document multimédia peut donc être défini comme une composition de médias selon plusieurs dimensions :

a) La *structure de style* qui permet de décorer ou paramétrer la présentation des éléments de multimédia. Par exemple, la taille et la police du texte, la vitesse d'affichage des images de la vidéo, le volume de l'audio, etc. Cette dimension est en fait dépendante du type de médias utilisés (cf. section 2.2).

b) La *structure spatiale* permettant de présenter graphiquement ou mettre en page des éléments de multimédia, ç à d, définir le placement des objets médias les uns par rapport les autres sur l'écran pendant la présentation.

c) La *structure temporelle* qui est la dimension spécifique du multimédia. Elle permet de définir l'ordonnancement temporel de la présentation des différents composants afin de construire le scénario souhaité et de définir l'évolution du contenu du multimédia dans le temps, par opposition aux présentations fixes des documents statiques traditionnels. De par la complexité de la tâche de spécification temporelle, il est nécessaire d'avoir un mécanisme efficace pour structurer les scénarios temporels. De plus, ce mécanisme doit pouvoir être supporté par un ensemble d'algorithmes pour vérifier la cohérence du scénario spécifié.

L'avancée technologique au niveau de la puissance des ordinateurs, des périphériques et des réseaux a abouti à la création et la propagation d'un grand nombre d'applications multimédia, telles que le télé-enseignement, la visioconférence, la vidéo à la demande, le e-commerce, la télévision interactive, les applications de la réalité virtuelle, etc. Ces applications nécessitent la coopération des utilisateurs qui contrôlent le déroulement des scénarios associés aux documents. On parle alors de *documents multimédias interactifs*. Une nouvelle dimension s'ajoute alors aux trois précédentes :

d) L'interactivité qui donne à l'utilisateur la possibilité d'interagir avec une présentation du multimédia. Cette dimension est aussi considérée comme une dimension sémantique du multimédia qui permet d'organiser le contenu d'une présentation de manière non linéaire. L'utilisateur peut donc, en utilisant des opérations dites TAC (Temporal Access Control), suivre des liens sémantiques dans la présentation pour contrôler le scénario (afin d'avancer la présentation, de l'arrêter, de la suspendre) ou encore accéder aux parties qui l'intéressent

(naviguer grâce à la nature hypermédia du document) ou encore plus, gérer les caractéristiques des médias, à savoir le volume d'une audio, la disposition d'une image ou d'un texte, etc.

Une dernière dimension apparaît dans un souci de qualité de service afin d'offrir aux utilisateurs de meilleures performances et de répondre ainsi au mieux à leurs aspirations :

e) L'ajout d'effets d'animation permet de dynamiser la présentation d'un élément ou d'un groupe d'éléments multimédias. Elle a un rôle important dans la présentation multimédia. Sans animation, une présentation multimédia peut devenir terne même si elle est bien définie dans les autres dimensions. Aujourd'hui, quand il manque de vrais systèmes multimédias, l'animation est utilisée pour augmenter la dynamique du document et se garantir l'intérêt de l'utilisateur. On peut constater par ailleurs que de nombreux langages/outils non multimédias proposent des moyens pour introduire des effets d'animation : la qualité des présentations PowerPoint a été améliorée depuis qu'il supporte des effets d'animation ; HTML utilise des scripts et des applets pour animer des éléments de HTML, ce qui est connu sous le nom de HTML dynamique ; Flash avec son format binaire de l'animation (swf) est également utilisé sur le Web. De ce fait les présentations obtenues sont plus vivantes et plus attractives.

Une étude approfondie confirme que ces dimensions, bien que distinctes, ne sont cependant pas indépendantes les unes des autres et leur séparation est parfois problématique [Rousseau 99], puisque certaines informations se retrouvent à cheval entre deux dimensions. En effet, la structure logique d'un document peut comprendre des informations qui proviennent des structures spatiale (par exemple pour une section composée de trois paragraphes est la succession de ces trois paragraphes) et temporelle (un film est la séquence de ses scènes). Les applications multimédias prennent en compte les divers aspects spécifiés afin de déterminer les actions à entreprendre pour rendre la présentation conformément à l'intention de l'auteur. Comme on peut le voir dans le cadre des oeuvres cinématographiques, le sens de la présentation naît de l'interaction de ces différentes dimensions.

La dimension temporelle reste quand à elle celle qui fait couler beaucoup d'encre de par le fait qu'elle soit spécifique aux documents multimédias et donc ignorée jusqu'à leur propagation. En fait, la présentation d'un document multimédia se distingue, principalement, de la présentation d'un document statique de par la dimension temporelle qui apparaît de façon interne aux objets médias (comme pour l'audio et la vidéo qui possèdent des propriétés

temporelles inhérentes), ainsi que de façon externe aux objets (comme pour les relations temporelles définies explicitement par l'auteur entre des objets médias). L'état d'un objet média dans un document multimédia change au fur et à mesure que la présentation avance dans le temps. Par exemple, un objet média qui joue au présent peut s'arrêter dans le futur et vice versa. La façon selon laquelle les changements d'état doivent avoir lieu est définie par le mode de synchronisation de la présentation multimédia et est détaillée dans la section suivante.

2.4 Qualité de service et DMMs

Avec l'évolution des technologies de communication et de l'informatique, les applications multimédias distribuées sont devenues omniprésentes et la qualité de service (QoS pour Quality of Service), offerte par ces applications, devient d'un intérêt fondamental. En effet, les applications multimédias manipulent un grand nombre de ressources (en termes de mémoires, bandes passantes, etc...) et nécessitent donc de les gérer, à différents niveaux, afin d'assurer une QoS de bout à bout (ang. : end-to-end) ainsi qu'une utilisation équitable de ces ressources.

De par ce principe, présenter des données multimédias de haute qualité de service est d'une importance capitale dans la conception des DMMs. La QoS dépend de la qualité des informations incluses dans le document, telles que la qualité de l'image : noir et blanc, dégradé de gris ou de couleurs [Jin 04]. Elle dépend aussi des coûts associés à la présentation tels que le prix à payer pour obtenir une copie d'un article édité dans une revue payante. L'utilisateur peut négocier ces paramètres selon ses besoins et ses ressources. En effet, il se peut qu'à cause de la qualité du moniteur, l'utilisateur préfère visualiser une image en noir et blanc que de ne pas avoir d'image du tout ou encore au lieu de visualiser en séquentiel deux vidéos, le lecteur du document préfère, afin de payer un prix moins élevé ou encore de gagner du temps, de présenter en parallèle les séquences vidéo de la manière qu'il juge la plus adaptée à ses besoins. Dans ce cas, on dit que le critère coût/ qualité est celui qui détermine le choix des utilisateurs [Almeida 04, Labeled 99, Labeled 04a].

En résumé, la spécification de la QoS d'un DMM doit

- présenter des paramètres quantitatifs (exp. jitter, délais, largeur de bande passante) et qualitatifs (exp. stratégies de gestion de la CPU, mécanismes de gestion d'erreurs) ainsi que les règles d'adaptation [Rogge 04] ;
- être de nature déclarative, ç à d, spécifier ce qui est nécessaire et non comment le gérer ;
- être associée à un processus de compilation permettant d'établir la correspondance entre la spécification et les composants du système qui exécute l'application.

En fait, trois niveaux sont considérées lorsque l'on parle de la spécification de la QoS dans le contexte multimédia : application, utilisateur et ressource :

- (1) Niveau utilisateur : au début de l'édition d'un document multimédia, l'utilisateur spécifie, à un niveau abstrait, la qualité (qualité du son, temps d'exécution, ...) qu'il s'attend à retrouver au moment de la présentation effective.
- (2) Niveau application : le développeur ou un compilateur traduit la spécification de la QoS que l'utilisateur a informellement proposé au niveau (1) en paramètres concrets. Cette spécification ne considère aucun paramètre de système d'exploitation ou de réseau mais tient en compte les propriétés telles que la réutilisation et extensibilité de l'application.
- (3) Niveau ressource : en dernier, pour l'application à exécuter sur une plateforme de système d'exploitation et dans un réseau physique le développeur ou le compilateur ont à raffiner la spécification du niveau d'application en y ajoutant les besoins ressource, tels que la largeur de la bande passante et la mémoire à allouer.

Malgré l'importance de la gestion de la QoS, et bien que un grand chemin a été parcouru pour faciliter et améliorer les présentations multimédias, peu de travaux se sont intéressés au critères de la qualité pour les DMMs. Parmi les systèmes et modèles qui ont pris en charge cette QoS on retrouve ceux présentés dans [Al-Salqan 96], [Golaup 06], [Little 93] et [Mourlas 02]. Les limites des travaux proposés se situent essentiellement au niveau de leurs modèles de documents qui ne permettent pas (ou de façon non exhaustive) :

- (i) la prise en charge des évolutions d'un document qui peut résulter d'une interaction survenue à la présentation ;
- (ii) les calculs des "coûts" associés à l'exécution du document, excepté pour certain qui utilisent cependant des algorithmes assez compliqués.

3 CONCLUSION

Depuis longtemps, les informations sont diffusées au travers des mass médias populaires comme les journaux ou les stations de radio. La télévision (en sous entendant le cinéma) fut le premier moyen de communiquer via deux médias, le son et l'image animée, une véritable révolution qui commença après la seconde guerre mondiale. On peut dire qu'elle fut d'une certaine manière la première brique du multimédia et son succès s'est étendu à des industries parallèles, en sortant du "petit écran" comme dans le cas du cinéma numérique aujourd'hui.

L'informatique s'est alors révélée comme un outil privilégié, voire incontournable, qui a permis de fournir des techniques et d'encourager des initiatives afin de résoudre des problèmes comme celui de la diversité des formats de données manipulées ou des méthodes de diffusion. De grandes tendances existent aujourd'hui au sein de l'informatique et le multimédia, avec ses diverses facettes, est l'une des plus populaires d'entre elles et l'on remarque que le domaine est en perpétuelle évolution et bénéficie d'une importante créativité qui est à la fois une source d'intérêt pour de nombreux utilisateurs et l'origine de divergences conceptuelles.

Partant de cet état de fait, nous avons porté notre choix sur les documents multimédia –leurs conception et édition-, domaine que nous avons parcouru en décrivant des aspects tels que les types des médias manipulés, les dimensions qui caractérisent les documents multimédias produits selon des processus bien étudiés garantissant une certaine qualité de service.

Le chapitre suivant est dédié à l'un des points les plus importants dans notre contexte de travail : la synchronisation des médias qui se regroupent selon un ordre temporel ainsi que spatial pour constituer une entité unique qu'est le document multimédia.

La synchronisation est une caractéristique importante des documents multimédias qui concrétise la sémantique d'une présentation multimédia conçue par un auteur. En effet, 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 interrompus 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 d'une animation et de l'apparition du texte "FIN" ; l'auteur peut souhaiter que ces deux instants soient simultanés et qu'en plus le texte soit positionné à un endroit précis par rapport à l'image.

Deux types de synchronisation sont identifiés dans la littérature multimédia : *la synchronisation temporelle* qui concerne l'ordonnancement temporel de la présentation des différents composants afin de construire le scénario souhaité ; et *la synchronisation spatiale* relative au positionnement des médias visuels sur le moniteur ainsi que leur disposition les uns par rapport aux autres au moment de leur présentation.

1 SYNCHRONISATION TEMPORELLE

Les données médias sont plongées dans le temps, même si leur comportement n'en dépend pas toujours, et confèrent aux présentations multimédias la même propriété intrinsèque. Parmi les multiples facteurs à prendre en compte dans une présentation multimédia, le temps est le plus important car c'est une ressource critique. Contrairement aux autres dimensions, il n'est pas contrôlable par les systèmes informatiques car son écoulement est une donnée intrinsèque. Par exemple, aucun équivalent à la notion de passé n'existe dans l'espace, même si des similitudes existent [Jourdan 98, Li 96].

Il est donc nécessaire d'indiquer comment sont assemblées les données médias dans le temps et vis-à-vis des autres données de la présentation multimédia, ce que l'on recoupe sous le nom de *synchronisation temporelle*. Certains langages multimédias n'indiquent néanmoins pas

explicitement la synchronisation, qui doit alors être déduite implicitement du modèle ou, le plus souvent, de l'application de présentation.

Aujourd'hui, les présentations multimédias doivent permettre à l'auteur de préciser la synchronisation qu'il désire et d'autre part assurer que l'exécution sera conforme à ses vœux, quelle que soit l'application de présentation envisagée. Chaque outil, application ou langage multimédia définit des constructions et offre des fonctionnalités à l'auteur pour qu'il édicte la synchronisation de sa présentation. Ces concepts définissent un modèle de synchronisation, parfois de manière implicite, qui permet de fixer leur cadre. Nous nous attacherons ici à examiner tout d'abord les caractéristiques de la synchronisation dans les différents outils multimédias pour ensuite présenter les différentes approches de synchronisations proposées dans la littérature des documents multimédias.

1.1 Caractéristiques de la synchronisation

La synchronisation comporte un certain nombre d'aspects découplés que l'on peut regrouper en plusieurs familles. Deux d'entre eux sont récurrents dans les outils multimédias : les formes de synchronisation et l'indéterminisme.

1.1.1 Les formes des compositions temporelles

Une synchronisation peut être définie entre des composants d'un même objet média (elle est dite *intra-objet*) ou encore entre des objets différents (c'est la synchronisation *inter-objets*) qui peut pour sa part être jugée naturelle ou synthétique.

1.1.1.1 Synchronisation *intra-objet*

Ce type de synchronisation est inhérent à l'objet média continu qui est considéré comme une suite ordonnée d'unités de présentation ayant des relations temporelles implicites entre elles comme pour les échantillons de son et les trames d'animation [Blakowski 96]. Les informations de synchronisation sont stockées dans l'objet lui-même lors de sa capture. Le rythme, ou vitesse, de présentation de ces unités d'information est un des premiers paramètres que l'on peut définir. Ce rythme peut être constant tout le long de la donnée ou bien varier afin d'introduire certains effets sur des portions de la donnée média ou de minimiser la taille des informations stockées. En effet, des modifications peuvent être nécessaires afin d'adapter la présentation des médias aux besoins des utilisateurs. L'exemple le plus répandu est

l'attribution d'une vitesse de présentation autre que la vitesse normale pour jouer une séquence vidéo plus rapidement ou plus lentement. En effet, et par définition, une vidéo est dite perçue dans les normes (sans accélération ni ralentissement) si sa vitesse de présentation est établie à 25 images/seconde ce qui équivaut à ce que la durée de présentation de chaque composant dans cette séquence est de 40 millisecondes (voir figure 1.6). Toute modification de ce rapport entraîne une altération de la vidéo perçue par l'œil humain.

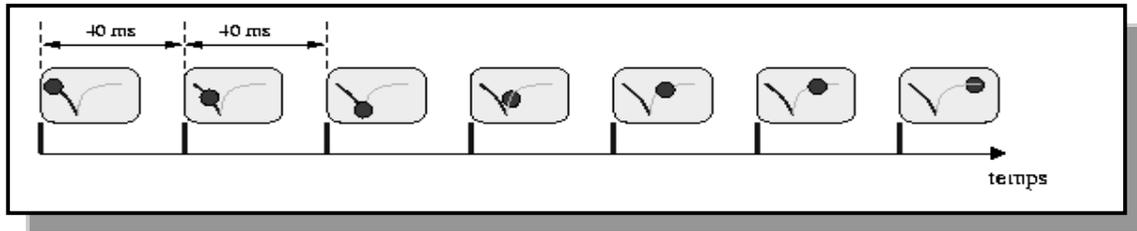


Figure 1.6 – Exemple d'une synchronisation intra- objets

Des relations plus complexes peuvent être mises en place entre les unités d'information. C'est par exemple le cas des images MPEG dont certaines peuvent être codées par rapport à d'autres comme dans l'exemple de deux images où il est spécifié que la deuxième doit être affichée progressivement selon une vitesse équivalente à la moitié du rythme de présentation de sa précédente. Ces relations peuvent s'appuyer sur les éléments logiciels qui produisent les unités d'information, comme c'est le cas du standard MPEG-4 qui offre des modèles des tampons où sont stockées les données et permet de prendre en compte les temps de transfert [Manjunath 03].

1.1.1.2 Synchronisation inter- objets

Les synchronisations inter-médias sont généralement des dépendances temporelles artificielles spécifiées explicitement par l'auteur du document. Ce sont ces liaisons qui définissent à proprement parler le contenu de haut niveau d'abstraction, ou scénario, de la présentation multimédia [Diaz 98].

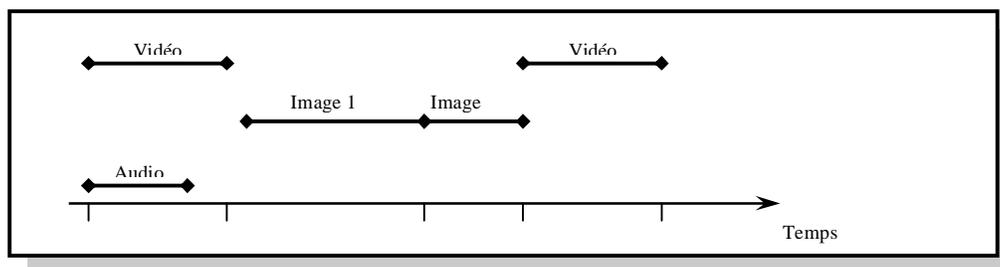


Figure 1.7 – Synchronisation synthétique

La synchronisation peut s'exprimer à divers niveaux de détails, ou granularité, à la fois sur des entités de taille plus ou moins fine et à travers des relations qui permettent de donner plus ou moins des précisions. De plus, la synchronisation inter- objets peut être synthétique (*à gros grain*) ou naturelle (*fine*). La granularité des éléments composés est le critère de classification.

La synchronisation synthétique est celle spécifiée explicitement par l'auteur entre différents objets médias afin de décrire un scénario souhaité. Cette forme est atteinte au plus haut niveau de granularité lorsque la synchronisation inter- médias n'a lieu qu'entre les points caractéristiques des données médias. Le début et la fin de la donnée média sont deux points caractéristiques à toutes les données médias (voir figure 1.7). Mais ces points caractéristiques peuvent correspondre à d'autres informations abstraites au sein des données médias, comme le début d'une scène dans un film par exemple. Ce niveau de synchronisation est le plus abstrait et permet de spécifier le comportement de la présentation multimédia d'un point de vue global.

La synchronisation dite fine lie des données médias, souvent une vidéo avec une audio, à l'échelle de leurs unités d'informations. A ce même niveau de détail, la synchronisation peut aussi admettre une précision beaucoup moins grande, par exemple dans un contexte distribué lorsque les données médias arrivent à des instants imprévisibles. Pour des valeurs importantes du délai entre deux unités d'information, la synchronisation n'est tolérée que pour des données textuelles ou graphiques. Il est par exemple possible de préciser des rôles maîtres et esclaves des diverses données afin de choisir le comportement à adopter lorsque la synchronisation ne peut être respectée [Rousseau 98]. Par exemple, si nous enregistrons une vidéo et son commentaire audio avec des vitesses respectives de 25 images/seconde et 8000 échantillons/seconde, une synchronisation naturelle est alors établie dont le paramètre est d'afficher une image vidéo tous les 320 échantillons audio. Ce type de synchronisation est appelé *synchronisation de lèvres* (*lip synchronisation*).

Il est aussi possible de ne considérer dans la synchronisation que des groupes d'unités d'information, appelée synchronisation à grains fins. Des points de synchronisation sont alors définis sur les diverses données médias et servent de points de rendez-vous lors de la restitution des données médias. La précision exigée est généralement peu importante du fait qu'une grande quantité d'unités d'information sont présentées entre deux points de synchronisation. Ce mécanisme correspond à une re-synchronisation car après un point de

synchronisation les données peuvent se désynchroniser jusqu'au prochain point de synchronisation où l'écart sera rectifié [Lo Presti 02].

1.1.2 L'indéterminisme

L'indéterminisme d'une présentation multimédia est la divergence entre la présentation et le scénario spécifié par l'utilisateur. Cette divergence peut avoir plusieurs sources, les plus importants étant l'*incertitude liée aux moments de début et/ou de fin* d'un média ainsi que les *délais générés par le système*, particulièrement dans un environnement distribué. Les diverses sources d'indéterminisme constituent à la fois un avantage, en donnant de la latitude à l'auteur du document, et un inconvénient, en obligeant le système responsable de l'exécution à gérer des informations incomplètes.

1.1.2.1 Instants de début et de fin indéterminés

La première source d'incertitude concerne les données médias. Ces données peuvent avoir un contenu dont les points limites de début et de fin ne sont pas connus avant leurs instants d'occurrence lors de la présentation du document multimédia. On parle souvent de données de durées inconnues, ou même d'objets incontrôlables, pour indiquer le fait que même si la synchronisation impose un instant de début, il est impossible d'avoir des informations sur la fin avant qu'elle ne survienne, ou l'inverse. Ces occurrences dépendent d'un facteur extérieur au système d'exécution qui peut être l'interaction avec un utilisateur. Dans ce cas, une donnée particulière encapsule généralement l'interaction, comme un bouton d'interaction pour lequel nous ne savons pas à priori à quel instant l'utilisateur va l'activer.

Cette forme d'indéterminisme se résume en fait à exprimer un délai indéterminé sur l'occurrence d'un instant [Courtiat 94]. Cet écart peut être majoré afin de limiter la source d'indéterminisme [Steinmetz 90] et empêcher qu'une attente infinie ne bloque l'exécution d'une présentation multimédia ou permettre par une vérification statique que la synchronisation ne comporte pas de problèmes [Sampaio 00b] tels que la présence d'un bouton d'interaction sans information sur le moment où l'utilisateur va appuyer sur le bouton.

1.1.2.2 Délais systèmes

Etant donné l'intérêt qui a été porté par la recherche dans les systèmes et les réseaux au multimédia, de nombreuses caractéristiques de ces systèmes ont été intégrées aux méthodes de synchronisation [Blair 97, Steinmetz 90]. Ces caractéristiques englobent toutes sortes de

délais existant à divers niveaux de la circulation d'information à travers les systèmes rentrant en jeu dans l'exécution de la présentation multimédia.

Un grand nombre de systèmes multimédias postulent le synchronisme des actions et de leurs réponses au sein du système, en considérant comme minimales les délais de réaction. Ils fournissent alors un service au mieux des ressources disponibles [Pérez-Luque 96], la puissance des ordinateurs et la rapidité des réseaux réduisant aujourd'hui ces délais à des valeurs faibles.

Les recherches dans les domaines des systèmes et des réseaux se placent plutôt sous l'hypothèse de l'asynchronisme entre actions et réponses, tentant de modéliser les écarts entre eux. Ceci permet de définir et garantir une qualité de service sur les délais d'un endroit à un autre du système. Ces garanties temps réels correspondent à des contraintes strictes dépendantes des données médias [Blair 97]. Les deux principales catégories de délais intervenant sont:

- ¶ les délais de transmission sur le réseau de communication [Santoso 93] ou bien d'accès aux unités de stockage, comme dans le cas de la vidéo à la demande [Campos 99] ;
- ¶ les délais de traitement des informations, comme ceux dus à l'usage de tampons [Koenen 01] ou liés à des procédures implantées matériellement [Chaoui 00]. C'est dans cette catégorie que s'inscrivent les cas des requêtes dont les réponses sont envoyées par des bases de données [Martin 97].

Note : L'indéterminisme introduit donc des décalages et, par conséquent, une désynchronisation entre les objets médias. Celle-ci peut dégrader la perception humaine du scénario. Des études expérimentales [Steinmetz 96] ont été effectuées sur les seuils de décalage acceptables par la perception humaine. Ces valeurs peuvent être utilisées pour mesurer la qualité de service offerte par le système de présentation [Adiba 99].

1.2 Modèles temporels pour les DMMs

Avant d'aborder les méthodes de spécification d'un scénario, nous commençons par la définition des entités de base qui le composent. En particulier, nous caractérisons le comportement temporel de chaque élément indépendamment d'un scénario donné. Cette étape nous permet de modéliser les différents éléments multimédias de façon homogène et

indépendante de leur contenu et de leur type (audio, vidéo, texte, etc.), on parle alors d'unités temporelles. Tout élément multimédia peut être manipulé à travers trois informations temporelles essentielles :

- 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 [Rogge 04].

Il existe deux façons de représenter le déroulement d'un scénario : à travers les changements qui surviennent (la terminaison de la vidéo correspond au démarrage de l'audio) ou au contraire en reliant globalement les activités entre elles (la séquence audio est présentée pendant la séquence vidéo). Ceci débouche sur deux types de représentations :

1. Une représentation fondée sur les instants. Dans ce cas, un élément multimédia, qu'il soit logique ou de base, est décrit dans un scénario par un instant de début et un instant de fin, comme dans Firefly [Buchanan 93, Buchanan 05] et Maestro [Drapeau 93].
2. Une représentation fondée sur les intervalles. Un élément multimédia est considéré comme une entité temporelle de base décrite par sa durée comme dans OCPN [Little 93] et Cmifed [Bulterman 95].

Dans les représentations multimédias fondées sur les intervalles, les unités temporelles de base peuvent être classées en trois catégories en fonctions des caractéristiques attachées à leurs durées [Layaïda 97] :

- ¶ Les intervalles discrets. Ce sont des unités dont la présentation ne dépend pas du temps, comme le texte ou les images fixes. En vue de leur intégration avec d'autres données ayant une dimension temporelle, celles-ci peuvent être affectées d'une durée de présentation explicite ou implicite dans le contexte d'un scénario donné.
- ¶ Les intervalles déterministes continus. Ils correspondent à des données dont la présentation dépend du temps et dont la valeur de durée est connue a priori. Des flots

audio et vidéo sont des exemples de telles unités temporelles. Dans certains cas comme la vidéo, la durée effective de présentation de ces données est liée à la vitesse de restitution des images qui la composent.

- ¶ Les intervalles indéterministes (discrets ou continus). Ces intervalles se distinguent par le fait qu'ils n'ont pas de durée connue a priori. Ils correspondent, par exemple, à des flots audio ou vidéo continus auxquels on accède à travers le réseau. Ce sont, en partie, ces éléments qui engendrent des scénarios indéterministes.

Les unités temporelles ne sont pas seules à influencer le mode de perception des scénarios multimédias mais elles sont associées aux relations temporelles qui permettent de décrire la façon dont les éléments multimédias doivent être combinés temporellement pour produire le scénario d'un document. Les relations temporelles servent à la fois comme entrée de la partie analyse d'un modèle et comme une mémoire des intentions de l'auteur. Cette représentation permet de définir l'état courant d'un document sur lequel s'appliquent les différentes opérations de composition.

Afin de définir un jeu de relations ou d'opérateurs temporels permettant la construction d'un document, nous étudions dans cette section trois aspects importants liés à ces relations temporelles qui peuvent intervenir pour décrire un scénario :

- Les unités temporelles mises en jeu dans les relations.
- La sémantique temporelle associée à ces relations.
- La topologie de la structure temporelle du scénario (arbre, graphe).

Étant donnée l'existence de deux modes de représentation des unités temporelles (les instants et les intervalles), il en résulte deux classes de relations temporelles : les relations temporelles fondées sur les instants et les relations fondées sur les intervalles. Dans ces deux types de relations, les scénarios sont représentés à partir d'un ensemble de relations temporelles dites primitives. Ces relations sont exclusives et permettent d'exprimer comment les unités temporelles d'un document (les éléments) se situent les unes par rapport aux autres. Dans ce qui suit nous présentons les différents modèles existants.

1.2.1 Modèles à base d'instants

Ces modèles manipulent la notion d'instant, ou point temporel, c'est-à-dire un morceau de temps qui ne peut être décomposé [Hamblin 72]. Tout instant possède une date qui indique sa position absolue dans le temps et est utilisée afin d'ordonner les instants. Les valeurs des dates peuvent être exprimées dans divers systèmes de mesure du temps, par exemple SMPTE time (Society of Motion Picture and Television Engineers) [Buford 94], qui sont tous isomorphes à l'ensemble des réels. L'ajout de relations entre les instants oblige à s'abstraire de leur date d'occurrence et l'on parle alors plutôt de points temporels.

Timeline

Le plus simple modèle à instant est le modèle timeline qui consiste à placer les instants sur un axe de temps en leur affectant une date fixe (voir figure 1.7). C'est le modèle qui forme généralement la base des applications de composition multimédia, comme toute la famille d'applications de la société Macromedia [Macromedia 01]. Il est intuitif et simple à mettre en oeuvre puisque le système d'exécution peut suivre simplement la ligne de temps pour respecter la synchronisation. D'autre part, on retrouve ce modèle dans le domaine des réseaux sous forme d'estampille [Santoso 93].

Ce modèle ne permet pas la synchronisation relative des données médias. En effet, la coïncidence d'instants peut être voulue mais ne peut être imposée car le déplacement d'un instant n'entraîne aucune modification de l'autre.

Le modèle timeline peut être étendu en un timeline arborescent [Hirzalla 95]. Les branchements entre différents timeline correspondent à des choix qui sont liés à une action de l'utilisateur et permettent de rendre une certaine forme d'interactivité par l'usage d'alternatives. Dans ce modèle, les timelines possèdent des points de début et de fin. Ces points peuvent être assignés à un point de branchement indiqué par un symbole de choix, un type de lien synchrone (déclenchement immédiat) ou asynchrone (un délai d'attente de l'action utilisateur est permis) et une limite maximale sur le nombre d'occurrences de ces points. Ce modèle conserve l'intuition du modèle timeline et l'augmente en permettant une forme d'interactivité contrôlée.

Réseaux de points temporels

Dans Firefly [Buchanan 93, Buchanan 05], un réseau de points est construit en indiquant une des relations "simultaneous with" ou "before by X", X étant l'écart qui doit exister entre deux points (voir figure 1.7). Les points peuvent être en début ou en fin de données médias ou bien être internes. L'étendue entre le début et la fin est un arc qui peut être contraint à travers trois valeurs, minimale, maximale et optimale.

Des coûts associés à l'extension ou la réduction des durées permettent de fixer des préférences entre ces trois valeurs. Ces valeurs, tout comme l'occurrence des points, peuvent être indéterminées afin d'indiquer qu'elles dépendent de conditions qui ne seront résolues que lors de la phase de présentation.

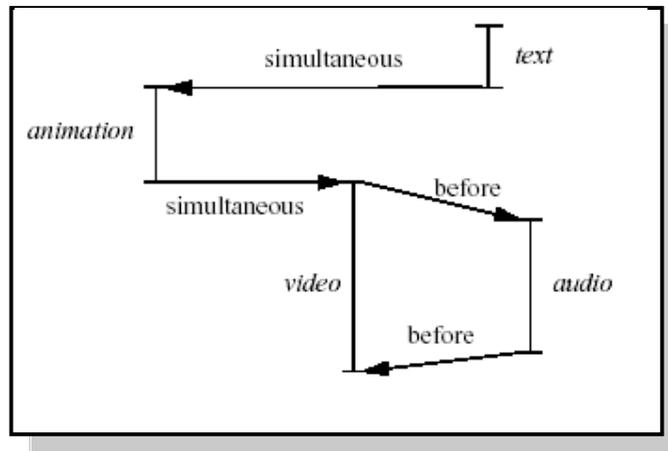


Figure 1.7 – Exemple de réseaux de points temporels

Un formatage de la spécification Firefly est nécessaire afin de déterminer des valeurs de l'ensemble des éléments qui respectent les contraintes spécifiées par le scénario. Un algorithme de programmation linéaire permet de trouver des valeurs lorsqu'il en existe, en minimisant les coûts. Les contraintes indéterministes sont mises à part et prises en compte uniquement lors de la présentation.

1.2.2 Modèles à événements

Un point de vue différent sur les points temporels est apporté par la notion d'événement. C'est une notion qui est née de l'abstraction de fonctionnalités matérielles, comme les interruptions, mais qui s'en démarque aujourd'hui. Un événement est instantané et est une occurrence dans le temps observable potentiellement plusieurs fois [Pérez-Luque 96]. Il peut être prédictible ou non, selon l'information à laquelle il est rattaché. Une de leurs

caractéristiques fondamentales est qu'ils sont observables. Ils ont parfois des structures typées qui permettent de leur associer des informations additionnelles [Lo Presti 02]. Divers mécanismes leur sont associés, d'une part leur gestion et d'autre part dans les conséquences de leur observation.

Les nombreux modèles à événements jouent sur ces différents aspects afin de fournir des relations de haut niveau d'abstraction tout en conservant l'intuitivité procurée par la notion d'événement.

Langages de script

Ces langages impératifs sont associés à des langages déclaratifs qui leur fournissent la notion d'événement. Ils définissent des couples événement- action afin de lier les événements entre eux et spécifier la synchronisation des données du document déclaratif. Ces langages étant peu complexes, il est possible de les interpréter au fur et à mesure de l'exécution de la présentation multimédia en laissant à un gestionnaire d'événements le soin d'appliquer les relations spécifiées. Chaque langage de script peut choisir entre des relations événement-action synchrones, c'est-à-dire qu'une seule relation peut être exécutée à un instant donné, ou bien asynchrones, plusieurs événements pouvant être observés à tout instant. La figure suivante représente l'exemple de description d'une partie de document en utilisant le langage de scripts associé au système de présentation de DMM qu'est Lingo. Il est, par exemple, spécifié à l'avant dernière ligne visible que l'audio "hello.au" n'est jouée qu'après que l'utilisateur clique sur le bouton d'interaction.

```
set window = main_win
set cursor = wait

clear win

put background "image.pic"

put text "heading.txt" at 10 10

start video "cannon.mpeg" at 30 30

if (user click) start audio
```

Figure 1.8 – Description d'un document au moyen de scripts (Système Lingo)

On retrouve souvent dans ces langages la même classification d'événements, les noms des événements indiqués provenant de divers langages:

- les événements liés au mécanisme de présentation : « onLoad » (chargement d'un élément du document), « onReload » (rechargement), « onLeave » (changement d'élément), « onFocus » (l'élément gagne le focus);
- les événements liés aux actions temporelles : « onFinish » (l'élément se termine);
- les événements liés à la structure du document : « new » (la création de la structure de donnée associée à l'élément est terminée), « change » (un champ de la structure de l'élément a changé);
- les événements utilisateurs : « onClick » (clic sur un élément), « onDrag » (déplacement d'un élément), « keyDown » (appui sur une touche).

Les langages OpenScript du logiciel Asymetrix Toolbook [Buford 94], LINGO du logiciel Macromedia Director [Persidsky 97] et ActionScript de la plate-forme Flash [Macromedia 01] sont trois exemples de langages de script.

1.2.3 Modèles à intervalles

Une façon différente de concevoir la synchronisation est apparue avec les travaux de Hamblin [Hamblin 72] puis Allen [Allen 83]. L'entité temporelle élémentaire manipulée par la synchronisation est ici un intervalle qui est défini comme une étendue ou portion de temps. Il est le plus souvent représenté par un intervalle numérique, celui entre les dates des instants de début et de fin de l'intervalle. Les intervalles sont un autre paradigme de synchronisation qui reposent sur une intuition différente du temps.

Logique d'intervalle

Les logiques d'intervalles, comme l' "interval temporal logic" [Dutertre 95] ou bien son extension : le "duration calculus" [Chaochen 91], abstraient clairement la notion d'intervalle en la définissant par un axiome mathématique de décomposition:

« Un intervalle est une entité qui peut être décomposée en deux sous- intervalles liés par un opérateur de coupure (chop noté;) qui représente leur adjacence ».

Ces logiques permettent d'élaborer des formules correspondant à une synchronisation désirée. Mais ces formules sont trop précises et complexes pour servir de base à une spécification de la synchronisation compréhensible par un programmeur. Ces logiques sont plutôt utilisées afin

de capturer les propriétés temporelles du modèle de synchronisation, comme dans [Blair 97] et [Campos 99], ce qui permet d'utiliser des techniques de « model checking » afin de vérifier l'incohérence des propriétés associées aux systèmes multimédias. D'autres logiques d'intervalles existent, comme « MetricInterval Temporal Logic » [Henzinger 98], et font l'objet d'axiomatisations différentes [Schobbens 02].

A noter que la décomposition des intervalles n'est en pratique plus possible au-dessous d'un certain niveau de précision, ce qui limite l'applicabilité de ces logiques. La plupart des modèles à intervalles considèrent l'intervalle correspondant à une donnée média comme entité atomique, c'est-à-dire non- décomposable dans le modèle. Une granularité plus fine n'est généralement utilisée que dans le cadre de la synchronisation fine.

Relations d'Allen

Les relations d'Allen [Allen 83] sont les relations sur les intervalles qui sont probablement les plus célèbres. De nombreux travaux non- informatiques ont utilisé ces relations afin de modéliser des phénomènes naturels et l'informatique ne dément pas cette tendance [Donikian 93, Euzenat 95, Layaïda 97, Mocellin 97].

Dans les relations à base d'intervalles, les relations possibles entre deux éléments multimédias se réduisent à toutes les combinaisons de placement possibles de deux intervalles sur une droite orientée. Le modèle le plus général, proposé par Allen, dresse la liste exhaustive de toutes ces relations [Allen 83]. La liste des combinaisons possibles entre éléments multimédias comporte ainsi 7 relations de base (voir figure 1.9) before, meets, overlaps, during, starts, finishes et la relation equals auxquelles sont ajoutées leurs 6 relations inverses, l'égalité étant équivalente à son inverse. Ces treize relations se répartissent en deux classes, celles des relations de séquentialité notée « Seq », à savoir before et meets, et les autres introduisant le parallélisme de présentation, notée « Par ».

Ces relations permettent de spécifier la synchronisation de manière qualitative sans devoir indiquer tous les détails, comme par exemple la valeur de l'écart entre deux intervalles dans la relation before. L'ensemble des conjonction de relations d'Allen forment une algèbre dénommée algèbre d'intervalles [Roşu 06, Vilain 86, Wahl 94]. D'autre part, Little et Ghafoor [Little 93] généralisent à un nombre quelconque d'arguments les relations d'Allen et définissent leurs inverses.

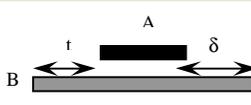
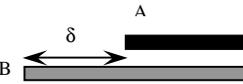
<i>Relation</i>	<i>Représentation Graphique</i>	<i>Classe</i>
A equals B		Par
A before B		Seq
A during B		Par
A overlaps B		Par
A meets B		Seq
A finishes B		Par
A starts B		Par

Figure 1.9 – Relations de base d’Allen

Les relations d’Allen s’adressent particulièrement aux tâches de raisonnements dans lesquelles on cherche à déduire des informations d’une spécification de la synchronisation, comme par exemple l’ensemble de toutes les contraintes satisfaisables entre des intervalles du scénario (appelé clôture).

Etant donné que les relations d’Allen modélisent toutes les relations possibles entre deux intervalles, un scénario à base de relations d’Allen peut être incohérent, c’est-à-dire qu’on peut ne pas trouver d’ordonnancement des intervalles satisfaisant ce scénario. La vérification de la cohérence est un des points les plus discutés dans la littérature des synchronisations multimédias c’est pourquoi, considérant l’importance du traitement à réaliser, une partie de ce travail s’y est intéressée.

Relations de Wahl- Rothermel

Les relations d’Allen sont insuffisantes pour ce qui concerne les tâches de planification mises en jeu dans l’exécution des présentations multimédias car elles ne permettent de

spécifier qu'une relation qui a eu lieu dans le passé et non d'indiquer une relation entre intervalles dans l'avenir. Wahl et Rothermel [Wahl 94] proposent de considérer des disjonctions de relations d'Allen afin de modéliser les incertitudes sur les relations entre intervalles.

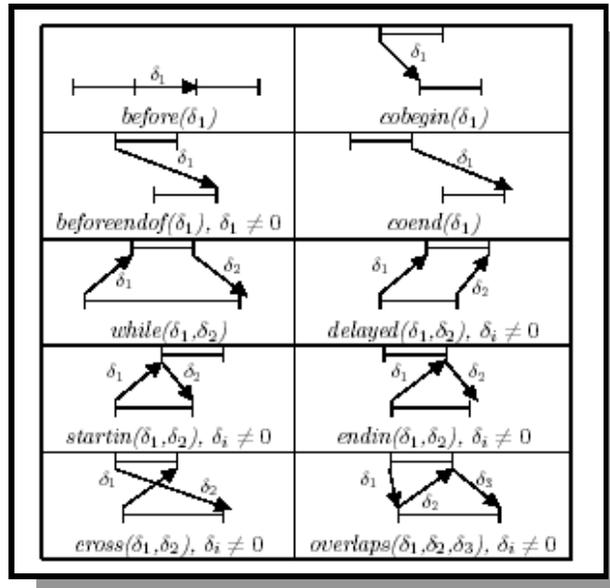


Figure 1.10 – Relations de Wahl-Rothermel

Ils dégagent un jeu de vingt-neuf disjonctions de relations d'Allen qui sont représentables en terme des relations sur les points $\{<, =, >, ?\}$, la relation ? correspondant à la disjonction des trois relations sur les points " $< V = V >$ ". Ces vingt-neuf relations sont modélisées par un jeu de dix opérateurs qui prennent comme arguments deux intervalles et de un à trois délais indiquant les écarts entre différents points limites des intervalles. Des contraintes de validité des opérateurs lient les connaissances des durées des intervalles aux valeurs des délais des opérateurs. Certains outils, tel que Madeus [Layaïda 97], intègrent des algorithmes incrémental de vérification de la cohérence des scénarios à base de relations de Wahl-Rothermel et des algorithmes de formatage des scénarios afin d'obtenir des valeurs d'intervalles leur correspondant. Ces opérateurs esquissés par Little et Ghafoor [Little 93] et appartenant à l'ensemble $\{before, cobegin, beforeendof, coend, while, delayed, startin, endin, cross, overlaps\}$ sont présentés dans la figure précédente.

Modèles de réseaux de Petri

Little et Ghafoor [Little 90] s'inscrivent dans un modèle à intervalles en définissant OCPN (Object Composition Petri Net). Leur idée est d'utiliser une structure de réseau de Petri

temporel afin de modéliser la synchronisation d'intervalles de durée fixe et connue. Les places des réseaux de Petri temporels correspondent à des intervalles ou à des délais et les transitions à des points de synchronisation.

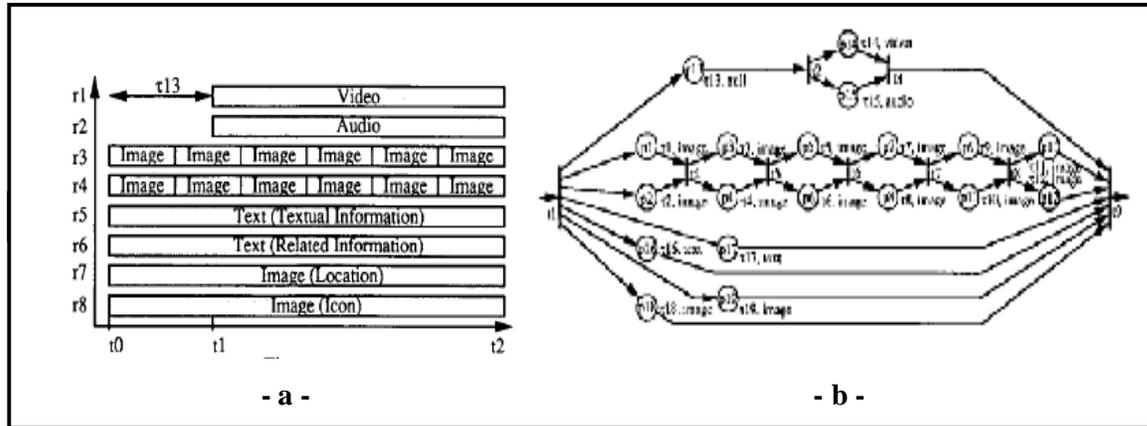


Figure 1.11 – Modélisation d'un DMM
 (a) Spécification sur axe temporel ; (b) Spécification par OCPN

Les jetons sont actifs lors de la présentation de l'intervalle et deviennent inactifs à la fin de celui-ci. Une transition est activée lorsque toutes ses places en entrée contiennent un jeton inactif, elle enlève alors ces jetons et introduit un jeton dans chacune de ses places en sortie. OCPN permet de décrire toutes les relations d'Allen et offre un formalisme graphique de spécification de la synchronisation.

Diaz et Sénac définissent des réseaux de Petri de flux temporels TSPN (Time Stream Petri Net) [Diaz 94]. Leur idée est de placer les intervalles sur les arcs et d'assigner à chaque transition une règle de déclenchement. Les intervalles, dits de validité temporelle, sont spécifiés selon leur durée minimale, nominale et maximale. Les règles de déclenchement se basent sur trois stratégies de base : au plus tôt des arcs, au plus tard des arcs ou selon les arcs maîtres. La combinaison de ces stratégies produit neuf règles qui précisent les priorités des différents arcs dans le déclenchement de la transition. Les TSPNs permettent de modéliser des scénarios indéterministes qui prennent en compte des délais sur les intervalles de manière complexe.

Les réseaux HTSPN (Hierarchical TSPN) [Willrich 01] ajoutent la possibilité d'encapsuler des places dans des places composites quand à la variation dite I-HTSPN (Interpreted HTSPN) [Willrich 02], elle attache une sémantique particulière aux places afin de modéliser

les structures conceptuelles, de contenu et de présentation des documents. Les I-HTPSN ont été adaptés aux langages MHEG et Java. Mais, bien que possédant une représentation graphique et des bases formelles, les réseaux de Petri temporels proposés, ne permettent pas de spécifier aisément la synchronisation des présentations multimédias. En effet, la sémantique derrière les symboles graphiques utilisés nécessite des adaptations particulières afin d'intégrer des formes d'indéterminismes relatifs, entre autres, aux interactions utilisateurs.

1.2.4 Les modèles hybrides

Il est difficile d'unifier les notions d'instant et d'intervalle car plusieurs problèmes difficiles se posent lorsque l'on considère les intervalles comme des événements de durée non nulle [Lo Presti 02]. D'une part, il est difficile de dire si les événements de durée nulle aux limites d'un événement de durée non nulle lui appartiennent [Allen 83]. D'autre part, leurs propriétés de densité sont opposées : on peut toujours décomposer un intervalle en deux unités consécutives alors qu'un événement n'est pas décomposable en plusieurs événements. La relation de composition entre "événement de durée non nulle" et "points" complique alors la sémantique du modèle [Hamblin 72]. Il est néanmoins possible de manipuler à la fois des instants et des intervalles au sein du même modèle grâce à des représentations particulières.

Multimedia Re-presentation Graph, Connecteurs hypermédias et DAMSEL sont des exemples de tels modèles sont présentés dans ce qui suit.

Multimedia Re-presentation Graph

L'auteur présente dans [Owen 98] un modèle général de synchronisation nommé Multimedia Re-presentation Graph (MRG). Les noeuds du graphe sont des éléments médias (Ang. : media elements) qui peuvent contenir des données à différentes granularités, ce qui offre des possibilités de modélisation à divers niveaux. Les arêtes représentent l'ordre des éléments médias en étant à la fois une transition causale et aussi un écoulement du temps. Les arêtes peuvent être pondérées par des probabilités et former, entre autre, des chaînes de Markov. Bien qu'initialement dédié à l'organisation d'informations textuelles, ce modèle permet de manipuler instants et intervalles. De nombreuses opérations sont définies sur les MRG afin de modéliser la corrélation de diverses données médias, par exemple l'alignement du son avec ses sous-titres. Néanmoins ce modèle n'explique pas complètement la sémantique des diverses relations, ce qui rend son usage difficile.

Connecteurs hypermédiats

Le travail présenté dans [Muchaluat-Saade 01] définit un langage d'édition qui repose sur la notion de connecteur hypermédia. Le modèle utilise une structure de composant qui peut être défini de manière composite et est relié à un connecteur hypermédia par le biais d'une liaison. Chaque événement correspond à une machine à état et possède des variables internes. Un connecteur hypermédia lie un nombre quelconque d'événements en attribuant à chacun d'eux des rôles et en indiquant leurs interactions à travers un attribut "glue". Le rôle d'un événement peut être de deux types: condition ou action. Un rôle condition correspond à un événement en entrée du connecteur hypermédia et définit les conditions qui doivent être successivement obtenues pour que l'événement participe à l'exécution du connecteur.

Un rôle action correspond à un événement en sortie du connecteur et indique quelles transitions effectuer sur cet événement. L'attribut glue combine les rôles conditions par des connecteurs logiques et peut appliquer un opérateur de retard " \oplus " afin de définir les délais entre deux événements. Cet attribut combine aussi les rôles actions par deux relations temporelles, notées \rightarrow pour la séquentialité et $|$ pour le parallélisme. Lorsque les événements en entrée sont activés, la « glue » du connecteur hypermédia indique quelles actions effectuer sur les états des événements en sortie.

La structure des connecteurs hypermédiats est proche de celle des réseaux de Petri temporels, à la différence que le connecteur hypermédia est plus riche qu'une transition puisqu'il permet à la fois de composer des événements de manière causale et de définir des liens hypermédiats. L'organisation des événements en machines à états ouvre de plus des perspectives de synchronisation différentes.

DAMSEL

Le langage DAMSEL [Pazandak 95a, Pazandak 95b] s'appuie sur la notion d'événement mais il permet de les composer entre eux : ainsi, étant donné deux événements a et b, $a > b$ constitue l'événement qui surviendra lorsqu'une occurrence de a surviendra après une occurrence de b. Les événements sont regroupés en catégories : produits par le système d'exécution ou de l'utilisateur, dépendant d'une condition logique ou provenant d'en dehors du système DAMSEL. Les événements sont étendus à des durées non nulles, modélisant ainsi des intervalles. Trois règles sont alors applicables sur les événements sous forme de prédicats

logiques : l'activation d'un événement par un autre selon un comportement permettant d'indiquer à l'ordonnanceur la manière de prendre en compte cette activation ; le "report" d'un événement hors d'un certain intervalle, qui réordonne un événement qui arriverait pendant cet intervalle à la fin de cet intervalle; et la synchronisation fine de deux intervalles, qui est un cas particulier d'activation.

D'autre part, DAMSEL envisage d'utiliser d'une logique temporelle (Conditional Temporal Logic) et une logique causale afin de contraindre un ordonnancement particulier d'événements dans le scénario. La synchronisation dans DAMSEL peut amener à des conflits dont la résolution peut être indiquée dans la spécification et possède deux modes, statique (à la compilation) ou dynamique (à l'exécution).

Algèbres de processus

Les algèbres de processus (process algebra) constituent des modèles complémentaires des logiques d'intervalles auxquelles elles sont parfois associées dans la définition des systèmes multimédias [Blair 97]. Ces techniques de description formelle (FDT, Formal Description Technique) sont initialement prévues pour spécifier l'ordonnancement des processus entrant en jeu dans une application.

LOTOS (Language of Temporal Ordering Specification) [Courtiat 96a] est une de ces algèbres de processus et est standardisée par l'organisme ISO. RT-LOTOS [Courtiat 94] est une extension de LOTOS aux applications temps réel dont l'ajout principal est l'opérateur delay (t) qui permet de fixer une durée de manière absolue [Courtiat 00].

Ce langage n'est pas utilisé directement pour spécifier la synchronisation, bien que cela soit possible mais peu intuitif. Il sert comme format intermédiaire afin de prouver de manière automatique la cohérence temporelle de présentations multimédias exprimées dans un autre langage, comme SMIL [Sampaio 00a, Sampaio 00b].

Les algèbres de processus font apparaître que la synchronisation des données médias peut être envisagée comme celle des processus qui les présentent. Les algèbres de processus peuvent être utilisées pour spécifier des systèmes multimédias utilisant à la fois des instants et des intervalles. Par exemple dans la Constraint Library [Courtiat 96b], le langage RT-LOTOS est utilisé pour fournir une bibliothèque de spécifications qui permettent d'organiser:

- des événements, selon des relations d'égalité (simultanéité ou bien écart fixe) ou d'inégalité (précédence avec une contrainte sur la valeur de l'écart qui peut être inconnue, supérieure et/ou inférieure à une valeur donnée);
- des intervalles, selon des relations de parallélisme (indiquant si la composition se termine lorsque la première, la dernière ou bien la présentation maître se termine) ou une des vingt-neuf relations de Wahl-Rothermel.

1.3 Contrôle du temps dans les DMMs

La présentation d'un DMM dépend essentiellement des contraintes de synchronisation temporelle établies entre les composants de ce document à la phase d'édition. Ces contraintes spécifiées via des relations temporelles et causales (voir paragraphes précédents), peuvent s'avérer incompatibles les unes avec les autres conduisant ainsi à des situations de dysfonctionnement sinon de blocage dites d'incohérence. Cependant, elles ne sont pas les seules à déterminer le risque potentiel pour créer des situations inconsistantes durant l'exécution. En effet, des interventions effectuées par l'utilisateur final, lequel interagit à l'aide d'opérations de contrôle, pour l'insertion ou la suppression d'éléments ou de relations peuvent aussi perturber ou arrêter l'exécution d'une présentation rendant ainsi le DMM inconsistant (ou incohérent).

1.3.1 *Qu'est ce qu'un document consistant ?*

Un document est dit « temporellement consistant » si :

- (i) l'action correspondant à la fin de présentation du document est atteinte à partir de l'état initial associé au déclenchement de l'exécution ;
- (ii) aucune interaction utilisateur ne conduit à une erreur d'exécution ou un état où les objets participants terminent leurs présentations de façon anormale.

Le contrôle d'intégrité nécessaire à la validation des documents permet donc de décider, à l'étape d'édition, si l'ordre des actions est consistant par rapport à un ensemble de contraintes ou non. Il est d'autant plus crucial lorsqu'il s'agit de documents avec scénarios de présentation complexes. La multitude d'événements et d'interactions pouvant s'effectuer durant une session peut alors créer un grand nombre de chemins d'évolution difficile à suivre au moment de l'édition si l'on considère toutes les implications possibles. C'est pourquoi, il

est nécessaire de fournir des outils pour aider les auteurs dans la conception de scénarios consistants ainsi que de vérifier et de valider l'intégrité de ceux qui existent déjà.

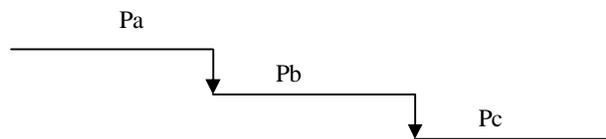
1.3.2 Classification

Dans la littérature multimédia plusieurs classifications de l'intégrité ont été proposées [Courtiait 96b, Ma 04, Sampaio 00b]. Par l'étude de leurs définitions, l'intersection des catégories proposées dans les différents travaux n'est en aucun cas nul et nous retrouvons bien des catégories communes aux uns et aux autres. De notre propre avis, la consistance d'un DMM peut être qualitative, quantitative, générique ou encore spécifique à l'application.

1.3.2.1 Consistance qualitative

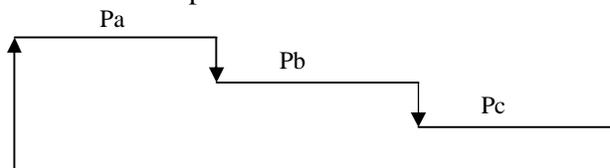
Ce type concerne la validité des combinaisons des relations temporelles parmi les objets médias de la présentation.

Exemple : soit trois médias 'a', 'b' et 'c', reconnus par leurs durées de présentations respectives : P_a , P_b et P_c , et reliés par les relations Meets (a, b) et Meets (b, c) comme schématiser dans la figure suivante :



Il est à remarquer qu'aucune inconsistance n'est relevée sur le scénario.

Variante : Par inadvertance, l'auteur rajoute une contrainte via la relation "*Meets (c, a)*" pour modéliser le souhait de reprendre l'exécution du média 'a' à la fin de la présentation de l'objet 'c'.

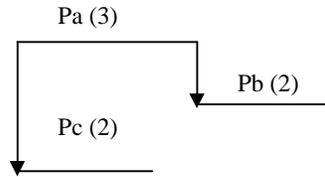


La fin du document n'est alors jamais atteinte c'est une **Inconsistance qualitative**.

1.3.2.2 Consistance quantitative

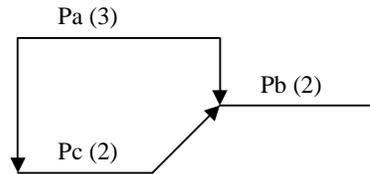
Cette consistance concerne la validité des combinaisons des relations temporelles parmi les objets médias de la présentation par rapport à leurs durées d'exécution.

Exemple : soit toujours nos trois médias 'a', 'b' et 'c', reliés par les relations Starts (a, c) et Meets (a, b) avec pour durées respectives 3, 2 et 2 unités comme schématiser dans la figure suivante :



On ne rencontre pas de problème d'inconsistance dans ce cas de figure.

Variante : Dans le cas où l'auteur rajoute une contrainte via la relation "Starts (c, b) " afin de spécifier le fait que la fin de l'exécution de l'objet 'c' doit enclencher l'exécution du composant 'b' comme dans la figure suivante :



L'objet 'b' ne peut pas être déclenché par 'a' et 'c', en même temps, car le premier termine le deuxième. C'est bien donc un cas de l'**Inconsistance quantitative**.

1.3.2.3 Consistance générique

Ce type est relatif à la validation des séquences d'actions d'interaction utilisateur (joue, pause, stop, retour en arrière, avancement, reprise, termine, changement de vitesse) par rapport aux règles d'activation.

En effet, en partant du fait qu'un média ne peut être que dans l'un des états : actif, inactif ou suspendu, la consistance est vérifiée à deux niveaux :

a Objet média : dans lequel cas, une interaction utilisateur est effectuée pour changer l'état dans lequel se trouve le média.

Exemple : soit un objet vidéo en cours d'exécution. Lors de l'interaction utilisateur « Pause », la vidéo est interrompue et l'objet passe de l'état actif à l'état suspendu.

Variante : Au même objet, mais qui vient de terminer son exécution, l'application de l'interaction « Pause » générerait une inconsistance vu qu'un média en état inactif ne peut passer à l'état suspendu.

b Scénario : dans ce cas, des contraintes sont associées à un document afin de préciser les scénarios possibles (particulièrement dans le cas des choix de présentations). Ainsi, par exemple, les durées associées aux médias ainsi que la durée maximale allouée à l'exécution du DMM peuvent être prises en considération dans le cas des interactions utilisateur.

Exemple : Soit un morceau de musique identifié par 'a', d'une durée de 4 minute 30 secondes, à présenter à la suite d'une vidéo 'v', d'une durée d'exécution de 15 minutes. Supposons que le lecteur du document, après la fin du film, ait le choix entre boucler sur la vidéo ou passer à la séquence musicale. Différents chemins se présentent alors (une fois la vidéo puis une fois l'audio, plusieurs fois la vidéo puis une fois l'audio, ou encore, avec les interactions permises, répéter la musique).

L'examen des différents chemins est alors nécessaire pour établir ceux qui sont consistants de ceux qui ne le sont pas.

1.3.2.4 Consistance spécifique à l'application

La consistance spécifique à l'application est relative aux contraintes additionnelles (que rajoute l'utilisateur au besoin) telles que :

- Deux sons 'a1' et 'a2' ne peuvent être activés en même temps ;
- Un certain clip vidéo 'c' ne peut être présenté avec un son 'a' ;
- Lors d'un partage d'application (exemple : en vidéoconférence) si un utilisateur manipule un bouton (il a donc pris momentanément l'interaction en charge) un autre perd le droit de l'utiliser jusqu'à la fin de l'interaction de celui qui l'a devancé.

1.3.3 Modèles et techniques de contrôle d'intégrité

En explorant les travaux réalisés dans la littérature multimédia on remarque que les travaux prenant en charge l'aspect vérification de l'intégrité des DMM ne sont pas suffisant. Les propositions faites se basent sur l'utilisation d'une mise en place de techniques de vérification et d'ordonnancement exécutées sur la structure interne qui représente le comportement temporel d'un document. Une solution courante consiste à utiliser un graphe temporel (tel qu'un graphe acyclique orienté) et des algorithmes spécifiques (tels que shortest

path solution, ou basé sur la programmation linéaire) pour déterminer l'ensemble des solutions valides pour la présentation d'un document.

Quelques approches qui réalisent la vérification et l'ordonnancement de documents multimédias s'appuient sur : (i) la projection sur des axes temporels (GRiNS [Bulterman 98]), (ii) la programmation linéaire (Firefly [Buchanan 93]), (iii) les graphes de contraintes (CHIMP [Candan 96], Madeus [Jourdan 98], Tiempo [Wirag 97], IMAP [Vazirgiannis 98]), (iv) les systèmes de transition (TOCPN [Yoon 98], machine à états finis étendus dynamiques – DEFMSMs [Huang 99]), (v) l'ordre partiel associé à la programmation linéaire et à des graphes de contraintes [Rivière 02] et (vi) les algorithmes basés sur l'état des composants des documents (FLIPS [Schnepf 96]) entre autres.

Parmi les travaux les plus connus -et les plus complets-, on retrouve le modèle proposé dans [Courtiat 96b], puis son extension proposée dans [Courtiat 00], où la formalisation – par l'auteur- des contraintes est automatiquement traduite en spécification formelle avec RT-LOTOS sur laquelle une recherche d'inconsistances temporelles est établie. Un autre travail est celui de l'approche proposée dans [Buchanan 05] pour la génération automatique de présentations consistantes. Dans ce travail, un algorithme de satisfaction temporelle génère des modules consistants selon les durées acceptables que l'auteur définit. Le principal reproche que l'on fait à cette proposition est l'incomplétude du modèle de composition associé.

Une autre proposition est celle de l'approche présentée dans [Prabhakaran 94] qui permet les présentations flexibles des DMMs avec une participation de l'utilisateur. Elle utilise une structure de réseaux de Petri temporisés dynamiques comme structure interne de la synchronisation sur laquelle sont appliquées quelques-unes des méthodes usuelles de vérification des réseaux de Petri. Dans l'esprit d'utilisation des réseaux de Petri, on retrouve les travaux présentés dans [Diaz 98, Willrich 01] et [Willrich 02] ainsi que dans [Guan 98, Guan 00, Guan 02] et de Vuong [Vuong 95], tous se basant sur différentes extensions (temporelles et temporisées) de réseaux de Petri et proposant des algorithmes de vérification cependant incomplètes par rapport aux exigences de DMMs.

Dans [Subrahmanian 97] les auteurs proposent une structure pour la spécification et présentation des DMMs, basée sur les contraintes temporelles et spatiales, et utilisent l'algorithme de Bellman-Ford pour résoudre les inéquations associées aux contraintes

exprimées par l'auteur du DMM. En outre, une approche basée sur les réseaux de contraintes temporelles (TCN pour Temporal Constraint Networks) est présentée dans [Dechter 91]. La vérification dans cette approche est réalisée de façon incrémentale à chaque fois qu'un objet est rajouté au document ce qui permet instantanément de reconnaître, au plus tôt, les inconsistances. Cependant les interactions utilisateurs ne sont pas prises en charges ce qui nécessite la construction d'un nouveau réseau à chaque modification sur la présentation souhaitée.

En dernier, sans avoir tout énuméré, nous pouvons citer le travail de Mirbel et al qui proposent dans [Mirbel 00] une méthodologie pour la vérification des consistances temporelles en s'appuyant sur des fragments de réseaux de contraintes (cnfs pour constraint network fragments). Plusieurs réseaux sont construits selon les unités fondamentales d'un scénario pour les différentes évolutions possibles durant l'exécution de la présentation.

1.3.4 Synthèse

Contrairement à ce qui est attendu -vu l'importance de la synchronisation (temporelle et spatiale) des médias des présentations multimédia- peu de travaux ont été consacrés, jusqu'à maintenant, à l'analyse des propriétés de cohérence et d'ordonnancement des DMMs [Buchanan 93, Courtiat 96b, Layaida 98, Willrich 02]. Cette situation est principalement due au fait que :

- ¶ la plupart des outils proposés pour la conception et la gestion des documents multimédias sont généralement orientés vers la facilité d'utilisation et ne possèdent aucune base formelle ni de support logiciel pour la vérification automatique des propriétés de cohérence des documents ;
- ¶ l'importance de ces propriétés a souvent été minimisée, en particulier lorsque l'on parlait de petites applications (de petites tailles) où sont utilisés des modèles de documents avec un pouvoir d'expression limités.

De plus, pour ce qui est de la synchronisation spatiale, différents travaux ont été proposés [Ghafoor 95 et Kwon 99] tous se basant sur le concept de région pour représenter la position spatiale d'un objet (voir chapitre 1 section 2.2). Cependant, la plupart de ces travaux ne considère que les contraintes spatiales statiques, il n'y a que le modèle STOG [Kwon 99] qui supporte les contraintes dynamiques.

2 SYNCHRONISATION SPATIALE

La spécification de l'organisation spatiale définit le placement des objets médias les uns par rapport aux autres sur l'écran pendant la présentation. Les données médias (si l'on excepte l'audio) sont plongées dans l'espace (ζ à d, présentée sur écran) et nécessite donc de préciser la composition spatiale de la présentation multimédia. Cette dimension des présentations multimédias a fait l'objet de moins d'études et de propositions que la dimension temporelle dans le domaine multimédia [Carcone 97]. Cette dimension n'étant pas sujette à un écoulement intrinsèque, le problème d'ordonnement des données est alors plus simple.

Tandis que les modèles temporels utilisent différentes approches (point, intervalle et événement) souvent complexes, la plupart des modèles spatiaux de document multimédia (HyTime, MHEG, SMIL, ZYX, Madeus) sont basés sur la notion de *région*. Celle-ci est simplement un rectangle (fenêtre) dans lequel un objet visuel peut être présenté. Les modèles spatiaux étant moins nombreux que les modèles temporels, ils sont plus aisés à classer. Une classification simple revient à distinguer les modèles absolus des modèles relatifs.

2.1 Modèles absolus

Ce type de spécification spatiale repose sur une description de l'espace en terme de référentiel, ζ à d, un système de coordonnées à deux ou trois axes. Les axes n'ont comme seul lien que le fait d'avoir une origine commune. Chaque axe a une direction et un ensemble de points. Les points spatiaux sur les axes peuvent avoir des valeurs, appartenants à l'ensemble des entiers naturels ou bien celui des réels, et sont ordonnés par les relations d'ordre "<", "=" et ">", avec la sémantique sous-jacente de l'ensemble de valeurs.

Un point est repéré dans l'espace par un ensemble de valeurs sur chaque axe appelé coordonnée. Il est possible de passer d'un référentiel à un autre en traduisant les coordonnées des points, généralement par l'usage d'une matrice de conversion. Les lignes brisées, polygones et polyèdres sont repérés par les coordonnées de leurs points caractéristiques. Les autres formes sont généralement définies à partir d'équations telles que les coordonnées de leurs points représentent les solutions. Les courbes de Bézier font parties des familles de courbes très utilisées dans le domaine du multimédia [Macromedia 01]. C'est ce modèle qui

représente les informations graphiques au niveau du système d'exécution de la présentation multimédia, deux axes suffisant dans la majorité des cas.

2.2 Modèles relatifs

Les modèles spatiaux relatifs utilisent des formes continues et précisent des jeux de relations entre elles. La position des objets découle des relations qui abstraient les dépendances spatiales entre objets. Un moyen simple de considérer ces objets est de les projeter sur les différents axes du référentiel. Chaque objet est alors représenté par un couple ou un triplet d'intervalles, ce qui est dénommé ombre temporelle dans [Adiba 96], dont la composition est désignée par le biais des relations d'Allen sur chaque axes. Un modèle spatio-temporel de ces relations est proposé dans [Adiba 96] dans le cadre du système de base de données STORM (Structural and Temporal Object oriented model for Multimedia data).

[Donikian 93] illustre ces relations dans le cas de la conception de scènes architecturales. Les autres modèles relatifs ne séparent pas les objets mais les considèrent comme les seules entités à organiser. Ces modèles sont plus proches de l'intuition que nous nous faisons de l'espace comme un médium uniforme et non comme la combinaison de deux ou trois dimensions.

Parmi les catégories de modèles qui se dégagent on trouve les modèles [Li 96]:

¶ Les modèles topologiques : sont fondés sur les notions de proximité et d'incidence. Le modèle RCC-8 (Region Connection Calculus) [Muller 98, Wolter 00] constitue le plus utilisé d'entre eux. Il définit les huit relations qui peuvent exister entre deux surfaces continues quelconques, comme illustré à la figure 1. 12. RCC-8 est la contrepartie des relations d'Allen en deux dimensions. [Gerevini 02] joint ces deux modèles en se limitant aux relations non- disjonctives et aux objets de taille fixe qui évoluent continûment. Tout comme pour les relations d'Allen, des disjonctions de relations RCC-8 permettent d'exprimer de l'indéterminisme sur la composition spatiale et sont la base de système de raisonnement sur cette composition [Wolter 00]. Dans [Bittner 02] l'auteur définit des relations d'inclusions hiérarchiques, ç à d, sans recouvrements, entre les régions dénommées granularités.

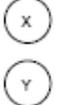
DC (disconnected)		TPP (tangential proper part)	
PO (partial overlap)		TPP ⁻¹ (TPP inverse)	
EC (externally connected)		NTPP (non-tangential proper part)	
EQ (equal)		NTPP ⁻¹ (NTPP inverse)	

Figure 1. 12 – Les relations spatiales RCC-8

- ¶ Les modèles directionnels : sont fondés sur l'ordonnement de l'espace, le plus simple d'entre eux divisant le plan (2D) en ses huit directions usuelles, nord, nord-est, est, sud-est, sud, sud-ouest, ouest et nord-ouest.
- ¶ les modèles basés sur la distance : se basent sur la définition d'une distance entre objets.

3 Conclusion

L'introduction de la dimension temporelle dans les documents conduit nécessairement à de nouveaux besoins d'expression : la durée des objets, leur placement temporel et leur synchronisation. Ces informations temporelles doivent s'intégrer dans l'ensemble des informations attachées aux documents. Par exemple, pour spécifier le déplacement d'un objet sur l'écran, il est nécessaire d'exprimer une information spatiale pour la trajectoire, comme 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.

Alors que ce chapitre s'est intéressé aux différents modèles permettant la spécification des synchronisations, le chapitre suivant présente les outils et standards réalisés dans les laboratoires de recherche ainsi que ceux proposés à la commercialisation.

Comme présenté dans les chapitres précédents, un document multimédia en cours d'édition englobe un certain nombre de fonctions. Au cours de cette section, nous allons en illustrer certaines qui ont fait l'objet de travaux de standardisation, que ce soit au niveau des langages de spécification ou au niveau des systèmes de présentation. Dans ce travail, nous présentons les six systèmes les plus connus et que l'on a jugé représentatifs : HyTime, PREMO, MHEG, Madeus, MPGS et SMIL.

1 HyTime

HyTime est un langage standard proposé par l'ISO (*International Organization for Standardization*) pour la structuration des documents multimédias et hypermédias, et notamment pour la représentation des liens hypertexte, de l'ordonnancement des événements spatiaux et temporels et de leur synchronisation [ISO 97, Erfle 93, Derose 94]. Il constitue une application de SGML (Standardized General Markup Language)[Goldfarb 90] qui est un standard de représentation de documents structurés statiques.

HyTime définit des éléments de type *espace fini de coordonnées* (*Finite Coordinate Space : FCS*) qui supportent la spécification d'un scénario en définissant des événements à des dates précises du temps absolu et/ou à des positions géométriques sur l'écran. Chaque type d'objet média (texte, audio ou vidéo) est associé à un FCS particulier. Par exemple un objet vidéo peut avoir quatre axes de coordonnée dans son FCS où chacun de ces axes a une unité de mesure différente (temps, position x et y sur l'écran et numéro d'image dans une séquence). Tout point d'un FCS correspond donc à un événement potentiel de la présentation d'un document. De ce fait, la spécification d'un événement se fait 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).

Le mode de désignation des objets médias à plusieurs niveaux de granularité offre un bon support d'hyperliens géré par le **module d'hyperliens** (*Hyperlinking Module*). Les informations sur les liens HyTime peuvent être stockées à l'intérieur (comme les liens contextuels *clink*) ou à l'extérieur (comme les liens indépendants *ilink*) du document contenant les ancrs sources. Ce dernier type de liens permet la modification de l'ensemble des liens sans modifier les documents liés.

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 rencontrent pour le lire et le comprendre [HyTime 97]. Néanmoins, quelques travaux ont été effectués à partir de HyTime comme par exemple les spécifications SMDL (*Standard Music Description Language*) [SMDL 98] 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 le travail de thèse de P. François [François 97] qui a étudié, spécifié et prototypé une structure d'accueil SGML/HyTime répondant aux besoins de stockage et d'exploitation de la documentation technique des avions. Un dernier exemple est la spécification du langage *XLink* (*XML Linking Language*) [W3C 98a] qui se base, en partie, sur la même façon de spécifier des liens que celle utilisée par HyTime. XLink est utilisé par le standard XML [W3C 97] pour décrire les hyperliens. Un exemple simple de synchronisation utilisant les définitions de HyTime est donné dans la figure 1. 13 qui spécifie la présentation de deux objets "extent1" et "extent2" qui sont à présenter en parallèle pour la même durée de temps.

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 médias ont des comportements temporels déterministes. En revanche, le standard HyTime ne supporte pas les objets médias indéterministes à cause de l'utilisation des axes de temps absolus qui minimisent la capacité d'adaptation aux conséquences d'indéterminisme.

```

<timefcs>
  <evsched>                               événement
  <event exspec="extent1">
  <event exspec="extent2">
  <extlist id="extent1">                   définition de extent1
    <dimspec id="dimension1"> 30 210      début :30; durée : 210
  <extlist id="extent2">                   définition de extent2
    <dimspec id="dimension2">
  <dimref elemref="dimension1"             début : égal au dernier
  selcomp="last"                          quantum de dimension1
  flip="flip">
  <dimref elemref="dimension1"             durée : même quantité de
  selcomp="qcnt">                          quantum que pour dimension1

```

Figure 1. 13– Exemple de présentation HyTime

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.

2 PREMO

PREMO (Presentation Environment for Multimedia Objects) [Herman 94] est un standard ISO (ISO/IEC 14478) qui se focalise sur la présentation des objets multimédias. L'objectif de la norme PREMO est de fournir un environnement de développement standard et portable pour les applications multimédias. PREMO sort du traditionnel paradigme du document pour s'inscrire dans le domaine des applications multimédias en ayant pour objectif de fournir un cadre de programmation flexible et ouvert. Il constitue une tentative pour fournir un middleware multimédia susceptible de permettre l'émergence de nombreuses applications et systèmes multimédias.

PREMO s'intéresse essentiellement aux techniques de présentation des données multimédias selon une approche objet issue du modèle développé par le consortium OMG (*Object Management Group*) et qui permet de spécifier les caractéristiques visibles des objets

(PREMO objects) indépendamment de leur réalisation. Le modèle objet PREMO est défini à partir de trois composants (hiérarchies de classes d'objets) [ISO 96a] : le composant *Foundation* pour la spécification abstraite des informations multimédias [ISO 96b], le composant *Multimedia System Services* pour la spécification des traitements de bas niveau sur les données multimédias [ISO 96c], et le composant "Modelling, Presentation, and Interaction" pour la présentation des médias en scènes [ISO 96d].

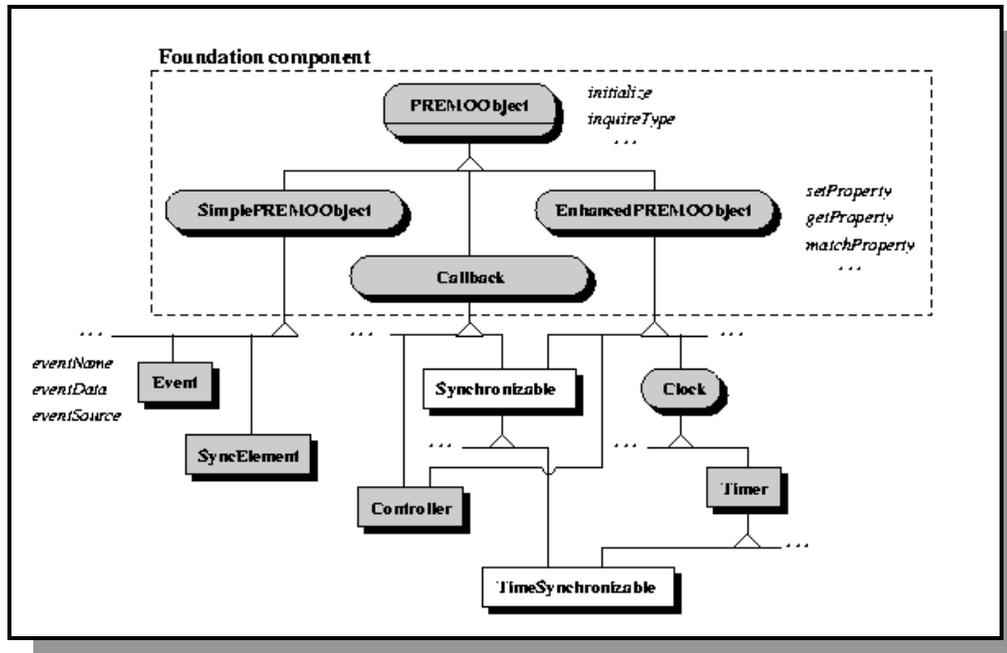


Figure 1.14 – Hiérarchie des objets PREMO du composant Foundation

Ainsi, le composant *Foundation* est une collection d'objets réalisant des services utiles pour une large variété d'autres composants. Ces objets, comme présentés dans la figure 1.14 ci-dessus, sont :

- *PREMOObject*, qui définit le comportement de base des objets PREMO : initialisation, destruction et fourniture d'informations sur le type d'objet ;
- *SimplePREMOObject*, qui permet de spécifier les structures de données (sans méthode) qui seront utilisées par les autres objets PREMO ;
- *EnhancedPREMOObject*, qui sert de base pour des objets comme les gestionnaires d'événements, les horloges ainsi que les objets médias temporisés (*TimeSynchronizable*) comme la vidéo ;

- *Callback*, utilisé pour la notification entre objets lors des changements d'état de ces objets ou lors de l'occurrence de certains événements. La classe *Controller* par exemple spécifie les événements qui la concernent et les comportements correspondants sont la réalisation des méthodes héritées de la classe *Callback*.

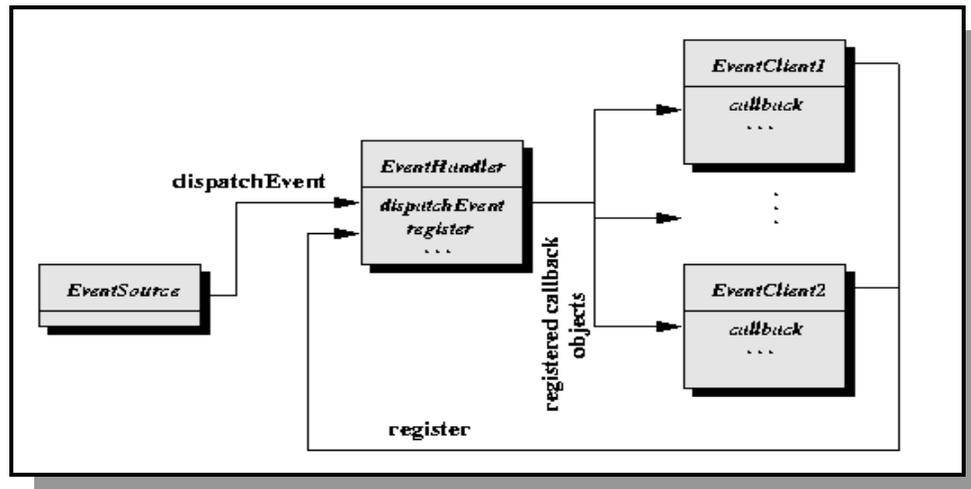


Figure 1.15 – Modèle d'événements de PREMO

Dans PREMO, la synchronisation est exprimée en termes d'objets "containers" groupant un ensemble d'objets médias et de classes "parallel" et "sequential" représentant, respectivement, les relations temporelles "par" et "seq". De plus, PREMO utilise un modèle d'événements, illustré dans la figure 1.15, qui réalise la synchronisation à base d'événements. Le modèle d'exécution spécifié par PREMO est donc fondé sur la programmation par événements.

En ce qui concerne les périphériques entrée/sortie de présentation, PREMO les représente de façon abstraite sous la forme de périphériques logiques afin de maintenir la généralité de la norme. En revanche, la gestion de la distribution des objets médias n'est pas clairement traitée.

Les diverses fonctionnalités proposées par le standard PREMO permettent aux programmeurs d'implémenter une multitude d'applications multimédias aux politiques de synchronisation très diverses. Afin d'assurer que ces applications n'aient qu'une unique interprétation, la sémantique des divers classes d'objets PREMO et de leurs méthodes est formellement spécifiée dans le langage Object-Z [Duke 97]. Chaque objet a un état interne dont l'évolution, après appel de ses méthodes, est décrite par une machine d'état finis. Des notes sont ajoutées à la spécification Object-Z des objets PREMO et de leurs méthodes afin d'exprimer en langue

naturelle des aspects fondamentaux qui ne sont pas clairement capturés en Object-Z, comme la progression du temps. Néanmoins, au niveau de la réalisation, il n'y a aucune application réelle basée sur PREMIO. Les travaux ont essentiellement portés sur la description formelle de PREMIO en utilisant le langage *Object-Z*.

3 MHEG

MHEG, du même nom que le groupe (Multimedia and Hypermedia Information Coding Expert Group) est un standard de l'ISO (ISO/IEC 3522) qui étend les préoccupations des standards JPEG et MPEG aux données multimédias et hypermédias. Le principal but est la définition et la standardisation d'un format portable de représentation multimédia qui seront utilisées dans des environnements aussi divers que les ordinateurs multimédias, des assistants numériques PDA, des bornes interactives ou bien les set-top box de télévision numérique.

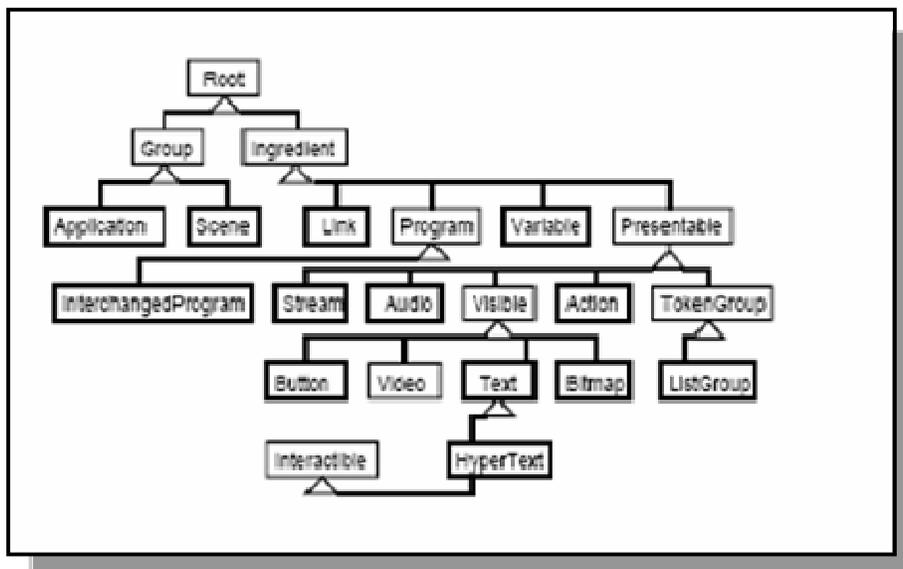


Figure 1. 16 – Hiérarchie des objets MHEG

Il faut noter que MHEG, par opposition à d'autres systèmes, gère la forme finale de la présentation, c'est-à-dire qu'il garde seulement les informations sur les relations temporelles et spatiales mais il supprime les informations sur la structuration logique de la présentation. Ceci a abouti à un modèle objet constitué d'une hiérarchie de classes *objet MHEG* comprenant les classes de contenu '*Content*' (qui permettent de décrire le contenu d'objets médias), *Script* (pour définir des comportements sous forme de programmes), *Composite*, *Link* et *Macro*.

MHEG définit donc une hiérarchie de classes dont la racine est "Root". La hiérarchie des objets est présentée dans la figure ci-dessus où l'on retrouve entre autres les objets "Bitmap" et "video" ainsi que "text". Il est bien clair sur dans la structure que ce dernier peut contenir des liens hypertextes comme on le voit au niveau de plus bas de la figure. Cette hiérarchie ne sert qu'à présenter de manière structurée les classes car il est impossible de définir de nouvelles classes, la notion d'héritage étant traduite par des importations et exportations de types de données. Le début d'une présentation correspond à un objet composite nommé l'objet racine (Root).

```
{:scene
  :object-identifiant:Example          la scène « Example » est constituée
                                       de:
  :group-items(
    {:bitmap                            l'image "Pic.jpg"
      :object-identifiant 1
      :initially-active true
      :original-box-size (320 240)
      :original-position (0 0)
      :content-data
      :referenced-content "Pic.jpg"
    }
    {:text                               un texte situé sur cette image
      :object-identifiant 2
      :original-box-size (280 20)
      :original-position (40 50)
      :content-data
      :included-content "Next"
    }
    {:link                               et un lien qui joue l'objet
      :event-source Example             « nextPart » lorsque l'on clique
      :event-type #UserInput
      :event-data #Left
      :link-effect :action
      :transition-to nextPart
    }
  )
})
```

Figure 1.17 – Exemple de présentation MHEG

Les objets composites sont liés entre eux à travers des pointeurs. De plus, les objets composites permettent la synchronisation entre parties d'une application multimédia à l'aide de primitives de composition spatiale et temporelle. Deux primitives temporelles binaires : "*serial*" (qui définit une exécution en séquence entre des objets médias) et "*parallel*" (qui indique une exécution simultanée des objets médias) sont utilisées pour spécifier les scénarios multimédias.

Les *objets MHEG links* ne sont pas les liens hypermédia (hyperliens) conventionnels mais permettent de spécifier les relations spatiales, temporelles et conditionnelles entre les objets médias ainsi que les actions effectuées par ou sur ces objets (condition – effet). Un exemple de synchronisation simple en MHEG-5 est présenté dans le scénario de la figure 1. 17 qui spécifie la scène "Example" constitué d'une image "Pic. jpg" qui doit occuper une fenêtre rectangulaire dont les points de référence sont les origines de l'écran de présentation. Sur cette image, presque à son milieu, on retrouvera le mot "Next" (objet de type text) qui lorsque l'on clique dessus (ç à d, à l'intérieur de l'espace fenêtre qu'il occupe) permet de présenter "nextPart" une autre partie du document.

Au niveau de l'interaction utilisateur, MHEG permet à l'utilisateur d'effectuer deux types d'interaction :

- la *sélection* qui lui permet de choisir un item dans un menu d'options;
- la *modification* qui permet de saisir des valeurs pour les paramètres de la présentation, comme exemple le volume d'une audio ou le nom d'un clip vidéo à jouer.

Le standard MHEG se compose de plusieurs parties notamment les parties MHEG-5 et MHEG-6. La partie MHEG-5 est destinée à supporter la distribution des applications multimédias dans une architecture client/serveur à travers différents types de plates-formes. Alors que la partie MHEG-6 est destinée à apporter la portabilité à MHEG-5 en ajoutant des fonctions de traitement de données, des fonctions de communication entre les serveurs et les clients et des fonctions d'interface avec les périphériques.

MHEG est à l'origine de plusieurs prototypes d'application multimédia. Le projet GLUE de DeTeBerkom est une réalisation du système MHEG-5 décrit en [Berkom 98]. Le projet MAJA [Bitzer 96], qui est un projet plus récent chez DeTeBerkom, 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 projet GLASS (GLobally Accessible ServiceS) [Geyer 97] 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.

4 Madeus

Madeus est un système d'édition et de présentation de documents multimédias interactifs [Layaïda 97]. Cet outil a été développé au sein du projet Opéra [Tardif 00, Tran_Thuong 01] avec une première version développée dans le cadre des travaux de [Layaïda 97] et [Sabry-Ismaïl 99] puis étendu par [Tardif 00], qui intégra des résolveurs de contraintes pour renforcer les capacités d'édition et de formatage du système et encore plus tard dans les travaux de Villard [Villard 02]. Ce travail posait les bases d'une édition à base de relations spatiales et temporelles et applique des résultats de recherche du domaine des documents structurés [Roisin 99]. Il prend donc en compte les trois types d'organisation des documents multimédias : logique, temporelle et spatiale, ainsi que l'aspect hypermédia [Sabry-Ismaïl 98].

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). 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 se basent sur les opérateurs de l'algèbre d'Allen [Allen 83], augmentées de relations temporelles causales qui permettent d'exprimer des effets de causalité entre les différents instants du document :

Parmaster (A, B) : signifie que l'occurrence de l'instant de fin de l'objet A provoque la terminaison de l'objet B. Si B est de type composite, la terminaison de A provoque la terminaison de tous les intervalles englobés dans B.

Parmin (A, B) : signifie que l'occurrence de fin de l'objet A ou B provoque respectivement la terminaison de B ou A.

Parmax (A, B) : signifie que l'occurrence de fin de l'objet de plus longue durée des deux objets provoque la terminaison de la construction.

```

<Composite Name      = "Générique"
  FontFamily = "Old English Txt MT"
  FontSize   = "16"
  FontStyle  = "bold" >
  <Audio Name      = "Audio"
    Source       = "generique.au"
    Duration     = "30 35 40" />
  <Image Name      = "Image"
    Source       = "last_shoot.jpg"
    Duration     = "10 35 100"
    Left        = "20"
    Top         = "40" />
  <Text Name       = "Text"
    Source       = "actors_names.html"
    FontSize    = "16"
    Width       = "100" />
  <Relations>
    <Temporal>
      <Equals Intervall1 = "Audio"
        Interval2 = "Image" />
      <Equals Intervall1 = "Image"
        Interval2 = "Text" />
    </Temporal>
    <Spatial>
      <Top_align Intervall1 = "Image"
        Interval2 = "Text" />
      <Right_spacing Intervall1 = "Image"
        Interval2 = "Text"
        distance = "10" />
    </Spatial>
  </Relations>
</Composite>

```

Figure 1. 18 – Exemple de présentation Madeus

Les relations spatiales adoptées décrivent l'alignement d'objets médias sur l'écran dans les deux dimensions : horizontale et verticale [Carcone 97]. Les relations représentent les types

de placement suivants : centrage (Center_v et Center_h), alignement (Left_align, Right_align, Top_align et Bottom_align), espacement (Left_spacing, Right_spacing, Top_spacing et Bottom_spacing) et décalage (Left_indent, Right_indent, Top_indent et Bottom_indent).

Par exemple, le code de la figure 1.18 est la spécification de l'objet composite Générique de film qui se compose de trois objets : Audio "generique.au", Image "last_shoot.jpg" et Texte "actors_names.html", avec des relations temporelles exprimant la présentation parallèle des trois objets. De même, la partie marquée <Spatial> définit les relations spatiales entre l'image et le texte. Les deux objets doivent être alignés par le haut et le texte est à une distance de 10 points à droite de l'image.

L'interface d'édition et de présentation de Madeus (proposée dans [Sabry-Ismaïl 99]) est construite au-dessus de la boîte à outils graphique de OSF/Motif [Motif 92]. Cette interface présente, principalement, deux vues :

- Une fenêtre « principale » qui permet de jouer et d'éditer le document. Elle s'appuie sur un service de pré-chargement des médias pour assurer une bonne qualité de présentation du document (respect des synchronisations). A cette fenêtre sont associées des palettes pour gérer les opérateurs temporels et spatiaux associés à la présentation ;
- Une fenêtre « source » associée à la structure hiérarchique permettant de visualiser et d'éditer les différentes structures du document.

5 MPGS

MPGS (Multimedia Presentation Generator System), comme son nom l'indique, est un système de génération de documents (ou présentations) multimédias [Bertino 00]. Le modèle de présentation de ce système s'appuie sur la notion d'objet où un objet multimédia est classé selon son appartenance à l'un de ces types :

- Objets de présentation (Display Objects) incluant tous les objets multimédias visualisés sur écran. Ils sont représentés par la classe DISPLAY_OBJ spécialisée en sous classes : VIDEO_OBJ, IMAGE_OBJ, TEXT_OBJ et GRAPHIC_OBJ. La spécification de ces objets se fait via un « monitor » identifié par "*mbr*" (minimum bounding rectangle) délimitant la hauteur, la largeur et la distance entre le coin supérieur gauche de l'objet et

celui de l'espace de présentation (écran). Formellement la fenêtre "mbr" est identifiée par le quadruplet :

"mbr = (x, y, w, h) " tel que : (x, y) définit le coin supérieur gauche, w : la largeur et h : la hauteur de l'objet.

- Objets audio représentés par la classe AUDIO_OBJ.

Cette classification permet une gestion très fluide, avec gain temporel, des propriétés associées aux médias de la présentation selon leurs types.

Une deuxième classification d'objet est possible dans MPGS. Un objet multimédia peut être considéré comme :

- Statique : S'il n'a pas de dimension temporelle, ç à d, que la durée de sa présentation n'est pas en fonction de son contenu. (TEXT_OBJ, GRAPHIC_OBJ, IMAGE_OBJ).
- Dynamique : S'il a une dimension temporelle implicite tel que les objets VIDEO_OBJ et AUDIO_OBJ.

La synchronisation temporelle entre objets d'un document MPGS est réalisée en s'appuyant, principalement, sur :

- Des relations qualitatives entre objets basées sur les relations d'Allen, en précisant leur type temporel par l'ajout du préfixe T (exp. : T_Before, T_Start, etc.) ;
- Une relation T_Delay qui spécifie le délai d'attente entre deux objets distincts. Cette relation donne une dimension quantitative aux relations temporelles ci-dessus ;
- Des primitives particulières à MPGS :
 - § T_UN_PAired : pour spécifier que deux objets ont des présentations disjointes ;
 - § T_EMPTYD : pour spécifier l'absence de présentations sur écran durant une certaine durée ;
 - § T_EMPTYA : pour spécifier l'absence de présentations audio durant une certaine durée.

Les contraintes spatiales dans MPGs sont quand à elles classées en deux groupes :

- Contraintes spatiales simultanées définies entre des objets simultanés appartenant à la classe DISPLAY_OBJ. Elles sont au nombre de cinq toutes avec le préfixe 'SS' : SS_MEET, SS_OVERLAP, SS_INCLUDE, SS_EQUAL, SS_DISTANCE ;

- Contraintes spatiales asynchrones, avec préfixe SA, définies entre toute paire d'objets visualisables : SA_DISJOINT, SA_MEET, SA_OVERLAP, SA_INCLUD, SA_EQUAL et SA_DISTANCE.

En regard aux besoins des utilisateurs, une notion d'une importance toute particulière a été introduite par les auteurs de MPGS à savoir le « Composant d'intérêt ». Cet ajout est dû à l'existence de certaines conditions où la présentation des objets ne peut être réalisée comme elle a été spécifiée (exp. image trop grande, séquence audio trop longue, etc.). La notion de composant (ou objet) d'intérêt a été introduite dans MPGS pour donner aux utilisateurs la possibilité de présenter ce qui, à son avis, a le plus d'intérêt dans l'objet (exp. Faire un zoom sur l'élément le plus important d'une image). La spécification du composant d'intérêt se réalise selon le type de l'objet :

- pour les objets dynamiques (vidéo, audio) la spécification se fait par un triplet (O, tmin, tmax) tel que tmin est la distance temporelle entre le début de l'objet et celui du composant d'intérêt et tmax la distance temporelle entre le début de l'objet et la fin du composant d'intérêt ;
- pour les objets statiques la spécification se fait par la paire (O, mbr_spec) où mbr_spec est la spécification du mbr relatif au composant d'intérêt.

Une deuxième proposition faite par les auteurs de MPGS afin de gérer des situations où les spécifications ne peuvent être exécutées telles qu'elles, repose sur le « Niveau de priorité », ou 'relevance' de l'objet ou de la contrainte. Ce paramètre détermine l'importance de présenter un objet dans une scène ou encore l'importance d'appliquer une contrainte entre les objets d'une spécification. Le niveau de priorité est une valeur de l'intervalle [1..100]. Une valeur par défaut = 50 est automatiquement attribuée si l'utilisateur ne précise pas le niveau de priorité.

En se basant sur les notions citées ci-dessus, l'environnement de spécification de MPGS permet de réaliser :

§ La spécification d'un objet par le quintuplet (O, Duration, mbr_spec, ic_spec, Prio) tel que ic_spec est l'ensemble des composants d'intérêt d'un objet.

§ La spécification de contrainte par le triplet (cnstr_name, params, Prio) tel que cnstr_name est le nom de la contrainte et params est l'ensemble des paramètres de la contrainte.

Exemple : (T_DELAY, texte1, image3, 5).

§ La spécification de la présentation globale par la quadruplet (OS, TRS, SRS, MFS) tel que OS est l'ensemble des spécifications d'objets, TRS est l'ensemble des spécifications des contraintes temporelles, SRS est l'ensemble des spécifications des contraintes spatiales et MFS l'ensemble des fonctions de déplacement des objets de la présentation.

Cette spécification est donc le résultat fourni par l'environnement de spécification de MPGS dont le prototype développé a été construit sur « ObjectStore- DBMS » pour la restitution des objets et l'outil MSF/ Motif pour l'interface graphique.

6 SMIL

SMIL (*Synchronized Multimedia Integration Language*) [Ayars 01, Bulterman 02, W3C 98b] est un langage déclaratif développé par le W3C, à base de balises défini en XML (eXtensible Markup Language), dont le but est de spécifier des documents multimédias sur le Web et de les jouer par des navigateurs SMIL.

SMIL est un format d'intégration, c'est-à-dire qu'il ne décrit pas le contenu des objets médias faisant partie d'une présentation multimédia, mais plutôt leur composition temporelle et spatiale ainsi que les hyperliens qui lient ces objets. En effet, les données à présenter sont encapsulées dans diverses balises qui indiquent leur type, comme par exemple 'animation', 'text' ou 'video', et sont référencées via des identificateurs URI (Uniform Resource Identifier). Les données étant référencées et non comprises dans le document SMIL, elles peuvent être partagées et réutilisées entre plusieurs documents.

La synchronisation des objets se fait à travers:

- § des balises qui définissent un conteneur temporel, les fils de cet élément étant les éléments encapsulés dans le conteneur; trois balises sont définies dans SMIL 2.0:
 - <seq> pour la présentation en séquence;
 - <par> pour la présentation en parallèle;
 - <excl> pour la présentation en mode exclusif, un seul des fils étant présenté à la fois avec la possibilité d'indiquer des priorités;

- § des attributs, définis pour la plupart des balises SMIL, qui permettent d'indiquer:
- l'instant de début (begin) ou de fin (end); ces attributs peuvent comporter une liste de valeurs qui sera interprétée comme une liste d'occurrences parmi laquelle tous les éléments ou bien uniquement le premier d'entre eux compte.
 - la durée de l'élément ;

```

<smil>
  <head>
  <layout>
    <root-layout height="350" width="600"
    background-color="#ffffff"
    title="Exemple de document SMIL"/>
    <region id="a" height="47" width="63" z-index="1"/>
    <region id="b" height="49" width="55" z-index="2"/>
    <region id="c" height="47" width="44" z-index="3"/>
  </layout>
  </head>
  <body>
    <par>
      <audio id="a" src="media/a.au" begin="5s" end="11s"/>
      <seq>
        
        <video id="c" src="media/c.mov" region="c" end="a.end"/>
      </seq>
    </par>
  </body>
</smil>

```

Figure 1. 19 – Exemple de présentation SMIL

Le code précédent (figure 1. 19) présente un exemple de spécification de document SMIL. Dans cet exemple, trois régions 'a', 'b' et 'c' sont définies afin d'afficher trois données médias identifiées par les mêmes noms. La composition temporelle consiste en la mise en parallèle de l'audio 'a' avec la séquence de l'image 'b', présentée cinq secondes, et la vidéo 'c', arrêtée par l'audio.

Les apports principaux de SMIL se situent au niveau de la portabilité du langage de spécification des documents multimédias et de la spécification de la navigation. En effet, dans SMIL, les ancres des liens hypermédia peuvent être définies activables pendant des intervalles

de temps et/ou sur des sous-régions de la fenêtre où l'objet média est affiché. Cependant, ce que l'on reproche entre autres à SMIL c'est le risque d'exprimer des incohérences temporelles ("deadlock" potentiel, objets jamais joués, ...) qui existe dans la première version de ce standard mais est rendu encore plus important dans sa deuxième version par l'intégration des événements puisque ceux-ci rendent le comportement du document beaucoup moins prévisible statiquement et donc plus difficile à appréhender pour l'auteur.

SMIL est en cours de construction [Layaïda 98, Tran_Thuong 03] et plusieurs sociétés d'informatique ont déjà annoncé leur support pour SMIL et un certain nombre de réalisations sont déjà effectuées, par exemple :

Le système HPAS (*Hypermedia Presentation and Authoring System*) [Yu 97] se charge de la présentation, de l'intégration et de la gestion des documents hypermédia temporisés. Il est réalisé en Java par Compaq [DEC 98] ;

Le système GRiNS (*GRaphical iNterface for creating and playing Smil documents*) [Bulterman 98] se charge de l'édition et de la présentation des documents SMIL. Il est réalisé en Python par le CWI [CWI 98] ;

Le RealSystem G2 [Real 98a] permet de jouer une vidéo et/ou une audio en temps réel en direct et à la demande. Il est réalisé en C par RealNetworks [Real 98b].

Ces réalisations ne supportent ni les mécanismes d'extensibilité pour les objets médias de base, ni la navigation dépendante du contexte, et les deux systèmes HPAS et GRiNS ne supportent pas la synchronisation de lèvres (*lip-sync*) entre une vidéo et une audio associée. Dans la suite de cette section, nous présentons brièvement le système GRiNS comme un exemple d'une réalisation qui s'appuie sur le langage SMIL.

GRiNS est un éditeur de documents SMIL issu des travaux menés par l'équipe du CWI d'Amsterdam autour de CMIFed [Hardman 93c]. Il est en cours de commercialisation par la société Oratrix. Grins permet à l'auteur de spécifier de manière indépendante les informations spatiales et temporelles et fournit les vues qui permettent d'éditer facilement des documents dans tous les axes de spécification : le temps, l'espace, les hyperliens, les attributs, les effets d'animation et de transition, etc. Toutes ces vues sont synchronisées. La vue la plus intéressante du système est la vue temporelle hiérarchique. Cette vue permet de visualiser très

facilement le scénario temporel du document (spécifié selon le modèle de SMIL). Par cette vue, l'auteur peut aussi percevoir très facilement le contenu du document grâce à des icônes représentant le contenu des médias. L'édition dans cette vue est aussi très conviviale. L'auteur peut prendre et déposer un nouveau contenu directement dans la vue. La modification des informations temporelles d'un objet média peut s'effectuer facilement par déplacement et modification de la taille de la boîte représentant l'objet. GRiNS fournit les vues qui permettent de parfaitement créer des documents multimédias SMIL à partir un ensemble de médias, néanmoins il n'est adapté qu'à la composition de scénario à gros grain. En effet, il est difficile avec cet outil de composer des synchronisations sophistiquées ou fines dans les documents. De plus l'éditeur GRiNS ne fournit aucun moyen pour la recherche ou le traitement des médias.

On peut trouver une situation similaire dans plusieurs autres environnements auteur comme Director [Macromedia] qui offre une vue temporelle très intuitive et une vue spatiale qui permet d'éditer directement des objets médias visuels et même des animations sur ces objets ; LimSee [LimSee] fournit une vue temporelle hiérarchique très puissante. En conclusion, les outils de cette famille offrent de bons moyens pour l'édition temporelle mais celle-ci reste au niveau des médias car les modèles sous-jacents (SMIL, Madeus) ne permettent pas un accès plus fin aux médias.

Dans GRiNS, la gestion des objets médias est limitée par un nombre fixe de types d'objets et par conséquent, le système n'est pas extensible face à l'évolution de nouveaux types et formats d'objets médias. Au niveau de la synchronisation, GRiNS gère uniquement la synchronisation à gros grain qui garantit seulement la synchronisation des débuts d'objets commençant simultanément. Le support de navigation est conçu sur les mêmes principes que ceux utilisés dans HTML, c'est-à-dire qu'il ne tient pas compte de la dimension temporelle d'un document multimédia. Au niveau de la distribution, les objets médias distants sont accédés par le protocole HTTP.

7 Discussion

Dans les sections précédentes, nous avons pu voir qu'il existe aujourd'hui de nombreux outils auteurs et systèmes d'édition et de présentation de documents multimédias. Le tableau

de la figure 1.20, présente une comparaison de ces travaux. En fait, il s'agit plus d'un résumé qui détermine si chacun d'eux permet de gérer :

- ¶ les types d'objets et opérations associées (1);
- ¶ le contrôle de la présentation des objets dynamiques (2) ;
- ¶ l'interactivité associée aux objets (3) ;
- ¶ les objets imprédictibles (4) ;
- ¶ la composition temporelle (5) ;
- ¶ les interactions utilisateurs (6) ;
- ¶ les différents supports de lecture (7) ;

et permet :

- ¶ l'utilisation facile de l'interface par des non informaticiens (8)
- ¶ la conception facile des documents (9)
- ¶ la présence d'un scénario indéterministe (10)
- ¶ la modification locale facile des propriétés à des fins d'amélioration de la QoS (11)
- ¶ la rapidité d'édition/ cycle de présentation (12)
- ¶ la présence d'un modèle de document multimédia (13)

<i>Propriétés</i>													
<i>Systèmes</i>	1	2	3	4	5	6	7	8	9	10	11	12	13
HyTime	+ / -	-	-	-	+ / -	-	?	-	?	-	-	-	-
PREMO	+ / -	+ / -	-	+ / -	+	+ / -	?	?	?	?	?	?	+
MHEG	+ / -	-	+	+	+ / -	+	-	+ / -	-	?	-	-	+
Madeus	+ / -	+ / -	+	+ / -	+ / -	+ / -	+ / -	+	+	+ / -	+	+	-
MPGS	+ / -	+ / -	+ / -	+	+	+ / -	?	+	?	+ / -	+ / -	?	+
SMIL	+ / -	-	?	+ / -	-	+ / -	-	+ / -	+ / -	?	-	+ / -	+ / -

Figure 1. 20 – Tableau récapitulatifs des propriétés des systèmes étudiés

La notation utilisée dans le tableau est "-" quand la propriété n'est pas supportée par le système, comme dans le cas du contrôle des objets dynamiques dans MHEG ; "+" quand elle l'est tout à fait comme avec la gestion des interactions associées aux objets de Madeus ; et "+/-" quand elle est partiellement supportée comme dans le cas de PREMO qui supporte

partiellement les objets imprédictibles. Lorsque aucune information n'est fournie cela est traduit par "?" comme dans le cas de HyTime sur lequel il n'existe pas d'informations quant à la difficulté de conception de ses documents.

En fait, ces systèmes ne sont en réalité qu'un échantillon représentatif des travaux de pionniers. En effet, il en existent bien d'autres qui peuvent être cités tels que TIEMPO [Wirag 97], ISIS [Kim 95] et Cuypers [Celentano 02]. Chacune des équipes de recherche présente une originalité par rapport aux autres mais toutes oeuvrent vers le même objectif : modéliser, éditer et gérer les DMMs en offrant à l'auteur le choix entre :

(i) Spécifier son document de manière absolue à l'aide d'une interface de type vue temporelle. L'expressivité des systèmes appliquant ce genre d'approche est relativement faible. L'auteur, dans ce cas, ne peut que placer dans le temps les objets et ne peut pas définir de relations entre eux. Par contre, les interfaces fournies sont relativement simples d'utilisation. Parmi ces propositions nous pouvons citer : SmilWizard, RealSlideShow et Powerpoint.

(ii) Spécifier son document à l'aide d'un langage de programmation. Ce choix offre une très grande expressivité à l'auteur, par contre les interfaces proposées sont des environnements de programmation classique tels que C++ et Java, qui sont difficilement exploitables pour des non informaticiens.

(iii) Spécifier son document à base d'approches relationnelles. Dans ces approches, l'objectif est d'offrir le meilleur compromis entre l'expressivité et la simplicité d'édition puisque l'auteur définit des relations entre les objets. En contrepartie, les systèmes d'éditions proposés fournissent des mécanismes de visualisation plus complexes pour aider l'auteur à manipuler les relations et à comprendre et maîtriser l'espace des solutions.

Il faut cependant noter qu'il n'existe pas encore de standard intégré qui puisse satisfaire efficacement et simultanément les besoins des deux aspects principaux d'une application multimédia, c'est-à-dire la spécification et la présentation tout en prenant en compte les besoins des utilisateurs (interface, temps de réponse aux requêtes et validation) et les caractéristiques des plateformes sur lesquelles les systèmes s'exécutent (distribution).

De plus, peu de travaux ont été consacrés, jusqu'à maintenant, à l'analyse des propriétés de cohérence et d'ordonnancement des DMMs [Buchanan 93, Courtiat 96b, Layaïda 97, Willrich

02]. En effet, la plupart des outils de conception proposés sont orientés vers la facilité d'utilisation et ne possède aucune base formelle ni de support logiciel pour la vérification automatique de la cohérence des documents. La principale raison de cette négligence est principalement due à la taille assez limitée des applications considérées (on parlait jusqu'à récemment de documents simples et pas trop complexes) ce qui n'est plus le cas avec l'évolution des technologies de communication.

Nous avons tenté de dresser dans cette partie un panorama aussi fidèle que possible des documents multimédias, de leur composition et du grand nombre de solutions existants afin de spécifier la composition et la diversité des approches pouvant être analysées en regard de certains critères simples.

De cette étude, nous pouvons déduire qu'un système auteur pour l'édition/présentation de DMMs doit fournir les fonctions suivantes :

- (i) la gestion des objets médias et l'évolution vers de nouveaux types et formats, c'est-à-dire l'extensibilité de l'application ;
- (ii) la gestion de la navigation intra- et inter- document en tenant compte de l'aspect temporel des documents ;
- (iii) la gestion de la portabilité à trois niveaux : le langage de spécification de l'organisation des informations multimédias, les différents formats d'objets médias et le code d'exécution ;
- (iv) la gestion de la synchronisation au niveau de la spécification sans la négliger au niveau de la présentation ;
- (v) la gestion de la distribution d'objets médias en tenant compte de l'échéance temporelle du scénario du document et des objets médias eux-mêmes ;
- (vi) la gestion de l'indéterminisme de la durée des objets incontrôlables en réduisant son effet sur la présentation dans le futur ;
- (vii) l'interaction de l'utilisateur au niveau de l'exécution et des différents impacts qui en résultent.

Selon notre étude de l'état de l'art, nous pouvons constater que bien du travail reste à réaliser afin d'aboutir à un système d'édition et de gestion de documents multimédias d'utilisation simple et offrant un maximum de fonctionnalités. Pour cela nous présentons dans les prochains chapitres notre contribution qui se concrétise par un modèle de document orienté objet ainsi qu'un modèle de vérification de la cohérence des synchronisations. Ces propositions représentent le noyau de notre système auteur "LUMIERE" sujet de la deuxième partie de cette thèse.

Au cours de la précédente partie, nous avons pu voir les différents formalismes utilisés pour définir des documents multimédias, les types d'environnement d'édition qui existent, ainsi que les différents besoins des auteurs au cours du processus d'édition. Nous avons vu comment ces besoins étaient plus ou moins couverts par les environnements existants. À partir de ces informations, nous allons à présent définir "*Lumière*" notre système auteur pour l'édition et la présentation des documents multimédias qui visent à satisfaire au mieux les besoins des auteurs identifiés auparavant.

Lumière est un système auteur d'édition et de présentation de documents multimédias interactifs. Il est dit "auteur" du fait que les besoins des utilisateurs (qu'ils soient auteurs ou utilisateurs finals) sont ceux qui priment sur toute autre considération. En effet, l'idée même de Lumière vient du constat que les systèmes de conception et de gestion des documents multimédias proposés rivalisent entre eux par le nombre des fonctions assurées à l'utilisateur mais sans prendre en charge ses besoins qui peuvent changer avant, durant ou même après la présentation du document.

De par ce qu'il propose comme outils et modèle de documents : (i) un modèle de document pour la modélisation générale (MDO2) ; (ii) le format interne spécifié en une extension des réseaux de Petri pour la vérification (RdP_MAL) ; et (iii) une interface utilisateur simple et répondant aux critères d'unification établis dans le domaine des IHMs ; Lumière se veut interactif, portable et d'utilisation facile pour les gens qui se disent non spécialistes.

Lumière prend en compte les trois types d'organisation des documents multimédia identifiés au chapitre 1 : logique, temporelle et spatiale. Son principe est de fournir un ensemble de fonctions pour :

- ¶ une édition de haut niveau avec une interface utilisateur permettant l'édition incrémentale ;
- ¶ une vérification efficace de la cohérence opérationnelle à l'édition et à la présentation;
- ¶ une estimation, rapide et multicritères, des coûts associés aux modélisations ; et

¶ un support efficace de présentation. Ce support permet d'offrir non seulement un environnement d'édition, mais aussi un environnement de présentation qui sont fortement couplés l'un à l'autre de façon à approcher un service « WYSIWYG » permettant ainsi à l'utilisateur d'être à la fois un auteur et un lecteur.

La structure de cette partie se présente en un chapitre (4) dédié au modèle de document de base de Lumière (MDO2), un chapitre 5 consacré au modèle de vérification basé réseaux de Petri, ainsi que la synthétisation de nos travaux présentée dans le chapitre 6 sous la forme de la présentation du cycle de vie d'un document de Lumière ainsi que l'architecture et l'interface du système. Le travail réalisé est illustré dans les chapitres (au fur et à mesure) par un exemple explicatif.

En se basant sur l'étude des modélisations existantes, nous proposons un **Modèle de Document Orienté Objets (MDO2)** afin de spécifier les scénarios et les contraintes qui s'y attachent dans un document multimédia. Le choix de l'approche orientée objet est motivé par les apports qu'offre la technologie associée :

- ¶ Puissance des concepts tels que la réutilisation ;
- ¶ Facilité de conception, de développement, de maintenance et d'évolution des systèmes ;
- ¶ Possibilité de gestion du dialogue homme-machine.

De plus, la technologie orientée objets permet l'extensibilité et offre ainsi la possibilité de définir et d'utiliser de nouveaux types de données, propriété très importante avec le développement continu observé dans le traitement des données. L'implémentation du modèle en Java permet d'obtenir un modèle de document générique et portable apte à être utilisé dans un système d'édition et de présentation de documents multimédias.

Un document multimédia modélisé par MDO2 est défini comme "*un ensemble d'éléments de base organisés selon une structure hiérarchique et ordonnés selon des contraintes dans le temps et l'espace*" [Labeled 04b, Labeled 05, Labeled 07]. Le modèle MDO2 offre une description à trois niveaux incluant : la **structure du contenu** qui décrit les informations lesquelles constituent les composants du document, la **structure de synchronisation** qui décrit les dimensions temporelle et spatiale du document ; ainsi que la **structure de présentation** qui décrit où et comment les composants seront présentés en réponse aux interactions des utilisateurs.

1 STRUCTURE DU CONTENU

La structure du contenu décrit les éléments multimédias qui composent un DMM. Elle s'intéresse aux types des composants, leurs formats ainsi qu'aux attributs qui leurs sont associés.

1.1 Types des composants

Les composants multimédias représentent les données utilisées afin de constituer un document multimédia. Pour cela et afin de produire des documents expressifs, un bon système auteurs doit offrir une riche palette des types de média [Buchanan 05].

Le modèle MDO2 supporte, actuellement, six types de base : *texte* et *image* (dits statiques) ainsi que *audio*, *vidéo*, *clips* et le type *composé* (dits dynamiques).

$$\text{Obj_MDO2} = \{X_i / X_i = \text{texte} / \text{image} / \text{audio} / \text{vidéo} / \text{clip} / \text{composé et } i \in \mathbb{N}^+\}$$

Alors que les quatre premiers types sont largement connus dans la littérature multimédia, nous définissant les deux derniers comme suit :

† « Un *clip* est une composition parallèle de séquences vidéo avec des séquences audio pour former une seule entité. » Cette composition est réalisée de façon à ce qu'une altération subie par un élément de l'entité à un moment donné se répercute sur toute la composition. Cette synchronisation est connue sous le nom de synchronisation fine (ou de lèvres) et diffère de la synchronisation synthétique établie entre les instants de début et/ ou de fin des objets média. Un exemple des clips peut être un film publicitaire réalisé de manière à ce qu'une demande d'arrêt d'image implique directement l'arrêt de la bande son associée, ce qui ne serait pas obligatoirement le cas s'il s'agissait de séquences audio synchronisées synthétiquement avec des séquences vidéo.

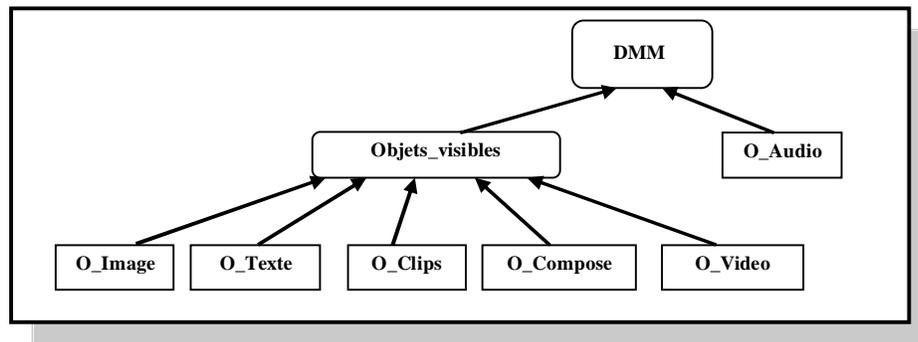


Figure 2.1 – Classification des objets de MDO2

† D'un autre côté, en considérant qu'un DMM édité peut être lui-même un composant d'un autre document, « le type *composé* regroupe des éléments avec formats externes tels que des pages HTML et XML, des scripts et programmes ou même des documents réalisés par des éditeurs multimédias tels que SMIL, Madeus et autres ». Ce type qui retrouve son origine dans

l'idée de l'objet à insérer dans Word, PowerPoint et même les éditeurs HTML, nous permet d'enrichir les documents multimédias par la diversité des informations manipulées.

Dès lors, la classification que nous adoptons pour un document multimédia est celle qui le présente comme une composition d'objets (dans le sens de la programmation orientée objets) multimédias regroupés en deux classes (*visible* et *audibles*) selon leur mode de perception (présenté dans la figure 2.1).

- ‡ Les *objets visibles* que l'on visualise sur écran. Cette classe regroupe les sous classes **O_Texte** pour les textes, **O_Image** pour les images et graphes, **O_Video** pour les films et vidéos, **O_Clips** pour les séquences audio-visuelles, et **O_Compose** pour les données externes.
- ‡ Les *objets audibles* ne pouvant disposer d'apparition sur le moniteur. C'est de cette classe (**O_Audio**) que sont instanciés tous les objets de type audio.

1.2 Attributs des composants

La spécification des propriétés des médias est d'une importance capitale pour les DMMs. En fait, cela traduit les capacités d'expression du modèle de document : plus grand est le nombre de propriétés d'un objet prises en compte, mieux ces objets sont traités (dégradation de couleur, mise en majuscule, etc.) et plus le modèle est dit riche.

Dans MDO2, un élément du document est caractérisé par un ensemble d'attributs pouvant être globales ou spécifiques [Labeled 04b, Labeled 05]. Les attributs dits globaux sont ceux associés aux objets quel que soit leur type tels que : le *nom* de l'objet (ou identifiant), qui doit être unique au sein d'un document, son *type* de base et la *source* qui sert à désigner le chemin d'accès dans le dépositaire où l'élément est stocké (c'est une URL si l'objet est récupéré sur le réseau ou le chemin local s'il se trouve dans une base de donnée locale). L'utilisation de l'attribut **source** permet de ne pas stocker les données (généralement volumineuses) mais juste de les référencer sans les contenir ce qui facilite l'apparition d'un élément plusieurs fois dans un document sans qu'il y ait de problèmes de stockage ou de redondance. Cette caractéristique, héritée de l'esprit orienté objets, donne une force supplémentaire au modèle.

A ces paramètres, MDO2 rajoute le **nom_d_auteur**, la **taille** (en Ko) et la **date_de_creation**, qui sont facultatifs mais pouvant être utiles dans une recherche, par exemple par mots clés, ainsi que des attributs spécifiques que l'on utilise pour certains types de médias tels que le **format** et la **taille_de_donnee** (exp : `taille_police` pour le texte), très utiles pour réaliser des effets de style lors de la présentation et à la gestion de la QoS qui doit être respectée lors de la présentation du document (voir chapitre 1).

D'autres attributs sont associés aux objets média selon leur type de base (c'est ce que nous avons nommé les attributs spécifiques) tels que l'attribut "**volume**" pour les objets audio, utilisé pour augmenter et diminuer le son lors de la présentation d'une audio. Se sont ces paramètres qui permettent d'accéder aux média, de les gérer et d'apporter des effets spéciaux lors de l'étape de présentation du document comme c'est le cas avec l'attribut "**qualite**" correspondant au format de codage (avec des valeurs telles que : MP3, WAVE, AIFF ou encore CDA pour les objets sons, jpeg ou bmp pour les images, etc.)

Une étude des attributs associés aux objets médias de MDO2 relève qu'il en faut d'autre afin de permettre une spécification plus complète des besoins des utilisateurs. Nous avons regroupé ces attributs sous trois classes : temporelle, spatiale et autre.

1. 2. 1 *Attributs temporels*

Pour chaque objet de base, il existe trois attributs à partir desquels la dimension temporelle de l'élément est spécifiée :

O L'attribut **duree** précise la durée de l'exécution (ou présentation) d'un objet média selon les bornes *inférieure* **di**, **dp** et **ds** qui représentent respectivement *les durées inférieure, préférable (ou nominale) et supérieure*. Dans le cas des médias dynamiques, la borne préférable est généralement la durée réelle de la présentation de l'objet qui s'attache au média à la fin de sa création physique comme pour l'enregistrement d'une vidéo. Dans le cas des objets statiques, ou même dynamiques dont la durée dépend des besoins de l'utilisateur, c'est ce dernier qui décide de la valeur qu'il accorde à l'exécution de chaque média. Mais bien que cette valeur soit suffisante dans certains cas, la spécification de la durée d'un élément est présentée en trois valeurs afin de permettre la gestion des incertitudes occasionnées par les ressources à partir desquels le composant est récupéré, ou même de celles où il sera exécuté. Partant du fait qu'un objet de MDO2 peut être

présenté plus d'une fois dans un document, sa durée est alors spécifiée par un ensemble de triplets tel que : $duree = \{(di_1, dp_1, ds_1), \dots (di_n, dp_n, ds_n)\}$ et $n \geq 1$.

- 0 L'attribut **controlable** spécifie si la durée d'exécution est fixée ou peut être modifiée, à l'exécution, par le système de présentation (en respectant les plages de **duree**).

Paramètre	Classe(s)	Valeurs / Commentaires
<i>police</i>	O_Text	Times New Roman/ Courier New/ Arial ...
style	O_Text	Simple / Gras/ Souligné / Italique
taille_police	O_Text	[6, 500]
coin-sup	Objets_visibles	Position initiale de l'objet
largeur	Objets_visibles	Emplacement de la fenêtre de présentation
hauteur	Objets_visibles	
duree	Tous les objets	
volume	O_Audio, O_Clip	[0, Maxint]
qualite	O_Image	Couleurs/ Noir et Blanc/ Nuances de gris
	O_Audio	CDA/ AAC/ WAV/ AIFF/ MP3/ WMA...
	O_Video	VHS/ HI-8 / DVD/ IMX/ HDV/ DVCAM...
vitesse	O_Audio, O_Video	± 30 frames /seconde
controlable	O_Audio, O_Video, O_Clip, O_Compose	Oui/ Non
iteration	O_Audio, O_Video, O_Clip, O_Compose	Nombre d'itérations ($i \in \mathbb{N}$)

Figure 2. 2 – Ensemble d'attributs d'objets MDO2

- 0 L'attribut **iteration** représente le nombre de fois qu'un élément est présenté en itération dans un scénario. Dans le cas où $iteration = 0$, l'objet ne sera présenté qu'une seule fois,

autrement à la fin de sa présentation il sera "rejoué" en boucle à partir du début. A chaque itération, le paramètre sera décrémenté d'une unité.

1. 2. 2 *Attributs spatiaux*

Pour chaque objet ayant une présence sur le moniteur, le système associe une fenêtre initiale de présentation selon les besoins de l'utilisateur. Trois attributs spatiaux à partir desquels la disposition de l'élément sur l'écran est réalisée :

- 0 L'attribut **coin-Sup** (x, y) qui représente les coordonnées du coin supérieur gauche de la fenêtre, ç à d, le déplacement horizontal et vertical à partir du point (0, 0) du moniteur.
- 0 L'attribut **largeur** qui représente la largeur de la fenêtre de présentation de l'élément.
- 0 L'attribut **hauteur** qui représente la hauteur de la fenêtre de présentation de l'élément.

Une fenêtre *fp* {*coin-sup*, *largeur*, *hauteur*} est ainsi définie pour chaque objet visible qui peut, selon le cas, subir une réduction ou un agrandissement dans les limites contrôlées à l'exécution.

La table de la figure 2.2 présente quelques attributs des médias de MDO2, alors que l'exemple suivant (figure 2. 3) représente l'implémentation de la spécification d'une vidéo, récupérée du disque dur et devant être visualisée, à une vitesse légèrement inférieure à la normale (25 frames /seconde au lieu de 30), une seule fois pendant 1 minute dans un coin de l'écran.

```
Video Videol {
    nom = Bébé_joue;
    source = C:\Dossier_Bébé\jeux.mpeg;
    .....
    coin_sup = (12, 5);
    largeur = 200;
    hauteur = 200;
    duree = (60, 60, 60);
    qualite = couleurs;
    controlable = Non;
    vitesse = 25;
    .
    .
}
```

Figure 2. 3 – Spécification d'un objet vidéo dans MDO2

1. 2. 3 Autres

D'autres attributs sont jugés nécessaires pour une spécification complète des besoins exprimés par l'auteur du document. Nous citons ceux relatifs à la notion de composant d'intérêt et ceux associés à la gestion du déplacement d'objets sur l'écran.

a Composants d'intérêt

La notion de stratégies de relaxation est apparue dans le système MPGS [Bertino 00]. Ces stratégies sont spécialement utilisées dans le cas où il s'avérerait impossible de générer le document selon les contraintes établies dans le scénario de l'utilisateur. Un exemple sera celui d'un document qui nécessite une heure pour être entièrement présenté alors que l'utilisateur ne dispose que de 25 minutes. La solution proposée est de réaliser un "Zoom" sur un (ou plusieurs) composant(s) en spécifiant la (les) partie(s) d'intérêt du (des) composant(s) qui sera (seront) présentée(s) si la génération de la globalité de l'objet (des objets) devient impossible.

Comme exemple plus précis, considérons un utilisateur qui demande à écouter une certaine musique en précisant que les 13 premières minutes sont les plus intéressantes. Dans le cas où, pour certaines contraintes, le composant audio ne pourrait être exécuté en entier, une méthode de relaxation est invoquée qui effectuera des modifications quant aux paramètres de l'objet à présenter (voir ici modifier les valeurs de l'attribut duree). Un autre exemple peut être celui d'une image dont les dimensions de la fenêtre de présentation si elles sont maintenues débordent, de façon gênante, sur celles des autres éléments de la présentation (voir figure 2.4.a). Une solution possible serait de ne présenter que ce qui est vraiment important (un zoom sur la cabane) ce qui diminuerai la taille de la fenêtre et permettra une meilleure vue des paysages (voir figure 2.4. b).

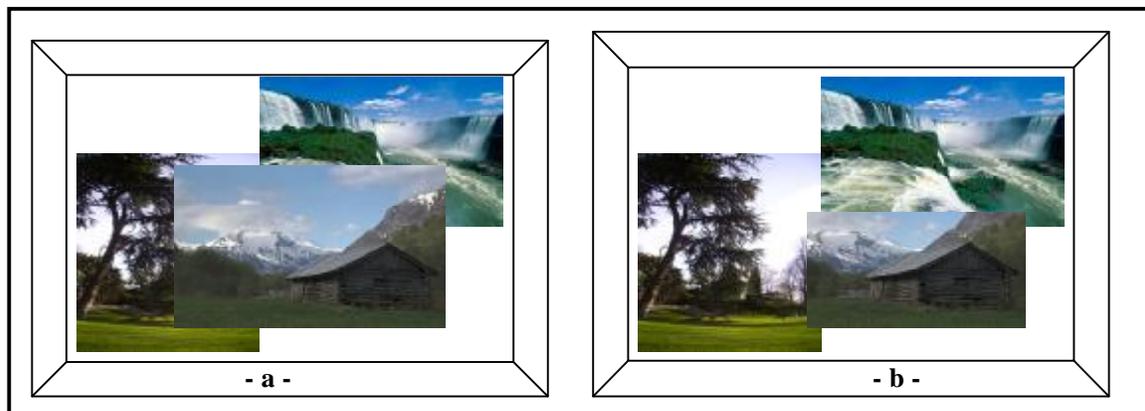


Figure 2. 4 – Exemple de composant d'intérêt

La spécification du composant d'intérêt dans un objet, dépend de son type. En effet, dans le cas des objets dynamiques, elle est relative à la dimension temporelle et consiste en la spécification des instants de début et de fin du composant d'intérêt, en respectant, bien sur, les attributs temporels de l'objet d'origine. Dans le cas des objets statiques (image, texte, etc.), le composant d'intérêt peut être spécifié selon la dimension spatiale en établissant une nouvelle fenêtre dont les dimensions (coin_sup, hauteur, etc.) sont incluses dans celles de la fenêtre réservée à l'objet d'origine.

Le modèle MDO2 permet la spécification des composants d'intérêt par une méthode facultative **obj-Int** invoquée pour les objets avec des paramètres variant selon la nature de l'objet d'origine :

- | **obj-Int (tmin, tmax)** pour les objets dynamiques, *tmin* et *tmax* étant respectivement la durée minimale et la durée maximale du composant d'intérêt ;
- | **obj-Int (dim-int)** pour les objets de nature visible, dim-int {Largeur, Hauteur} regroupant les attributs spatiaux du nouvel objet.

b *Déplacement de composants*

Il s'avère de plus en plus intéressant d'introduire dans les documents multimédias des objets visibles qui ne maintiennent pas la même position durant leur activation mais se déplacent selon une fonction de déplacement prédéfinie. Par exemple, il serait plus attractif que le texte d'un générique de fin se déplace de bas vers le haut (ou dans le contraire) ou encore qu'une image dans une publicité fasse son apparition d'un coin et se déplace en cadence vers sa position finale.

MDO2 introduit une relation **deplacement {fp, vd, td, retour}** à appliquer sur les objets visibles. Ces derniers peuvent se déplacer dans un espace (fenêtre) dont les attributs spatiaux sont retrouvés dans **fp** (voir section 1.2.2), à une certaine vitesse **vd** et un décalage **td** (**Hav, Har, Vav, Var**) dont les paramètres sont les angles de déplacements, respectivement, Horizontal, et Vertical (avant et arrière). L'objet atteignant les coordonnées maximales de l'espace de déplacement, peut se figer ou refaire le même parcours selon la valeur (faux ou vrai) du paramètre "**retour**".

2 STRUCTURE DE SYNCHRONISATION

La structure conceptuelle du modèle spécifie les compositions temporelles et spatiales en s'appuyant sur un ensemble de relations implémentées en méthodes MDO2.

2.1 Synchronisation temporelle

Un document **Lumière** est défini comme un ensemble d'éléments organisés selon un scénario. Ce dernier regroupe un ensemble d'objets de différents types et avec différentes particularités. Le modèle de temps attaché aux objets ainsi que les compositions temporelles des objets, exprimées sous forme de contraintes, sont décrits dans les sections suivantes.

2.1.1 Principe de la gestion du temps

Un objet dans MDO2 peut être projeté sur l'axe du temps. Pour cela, il est représenté par (i) un temps d'exécution (**Tex**) qui correspond à l'instant où la présentation de l'objet débute, et (ii) un attribué "*duree*" (voir section 1.2.1) identifié par un intervalle qui précise les bornes de durée d'un élément : durée inférieure, valeur préférable et borne supérieure.

Cette forme d'intervalle permet à l'auteur du DMM de spécifier la durée d'exécution (ou présentation) d'un média selon les formes suivantes :

- [di, dp, ds] : cas général d'un intervalle de valeurs possibles : une borne inférieure *di*, une borne supérieure *ds* avec une valeur préférable *dp*. Dans ce cas, l'auteur a donné le temps durant lequel il préfère que l'objet soit présenté mais il permet au système de présentation de modifier, au besoin, cette durée (dans les limites de la plage de temps qu'il a permise via les valeurs de "*di*" et "*ds*", ç à d, " $di \leq dp \leq ds$ ") ;
- [x, x, x] : une seule valeur *x* est spécifiée explicitement par l'auteur (une seule valeur pour la borne inférieure, la borne supérieure et la valeur préférable). Cette forme est employée lorsque l'auteur tient à ce que la durée de présentation soit respectée. Dans ce cas, si au moment de l'exécution une sérieuse contrainte survient lors de l'exécution pour empêcher le média de s'accomplir selon les souhaits de l'auteur, le système doit obligatoirement se référer à l'utilisateur pour que soit il donne son accord afin de modifier cette durée (selon

les propositions qui lui sont présentées) ; ou qu'il accepte de terminer la présentation du document.

- 0 [-, dp , -] : seulement la valeur préférable est spécifiée par l'auteur. Le système utilise une valeur par défaut (*def*) en fonction du type d'objet média pour calculer les bornes inférieure et supérieure [dp -def, dp , dp +def]. Il est à noter que ce calcul n'est réalisé, dans le cas des objets dynamiques, que dans le cas où l'attribut contrôlable de ce média (voir section 1.2.1) est initialisé à la valeur oui.
- 0 [di , -, ds] : l'auteur n'a aucune préférence pour la valeur de durée. Dans ce cas, le système calcule la moyenne des valeurs et l'affecte à la durée préférable, ç à d , $dp = (di + d) / 2$.

Une dernière possibilité est celle :

- 0 des durées non spécifiées. A première vue cela peut se rencontrer pour les médias statiques (image, texte, etc.), cependant il a été constaté que même pour les objets dynamiques le cas s'est présenté. Dans ce cas, les valeurs de la borne inférieure et de la borne supérieure de durée sont initialisées respectivement à zéro et à l'infini quand à la valeur préférable de ce média, elle est calculée en se basant sur les valeurs préférables des objets qui sont reliés à cet objet par des relations temporelles (voir section suivante).

Dans le but de contrôler les présentations des médias d'un document multimédia, particulièrement dans un environnement distribué, il est impératif de surveiller le temps afin d'éviter, ou au moins gérer, les problèmes tels que les gîgues et les dérivées. Aussi, est-il essentiel de relier chaque objet à une horloge locale (**CK**) qui surveillera l'avancement de sa présentation dans le temps (voir chapitre 6). Cette horloge :

- § gardera une initialisation à zéro tant que l'exécution de l'objet n'a pas commencé ;
- § s'incrémentera d'une unité à chaque seconde jusqu'à la fin de l'exécution ;
- § gardera la valeur d'arrêt en cas d'interruption (provisoire ou définitive) qui sera utilisée au besoin selon les actions réalisées dans la suite du temps ou même pour une reconstitution de l'historique du document.

Chaque horloge (CK_i) d'un objet "i" est gérée par plusieurs méthodes qui permettent de suivre l'exécution des objets :

- o démarre (CK_i) : démarre l'horloge interne de l'objet ;
- o tic (CK_i) : fait progresser l'horloge d'un quantum de temps (une unité ou tic).

- termine (CK_i) a pour effet d'interrompre (terminaison définitive) la progression de l'horloge.
- pause (CK_i) : suspend l'horloge de l'objet suite à une interaction utilisateur.
- redemarre (CK_i) : relance la progression de l'horloge après une suspendue.
- change_vitesse (CK_i , Speed) : modifie la cadence d'une horloge dans le cas d'une (dés) accélération selon un taux (négatif ou positif).
- fin (CK_i) : indique si l'horloge a atteint sa durée d'échéance.

En complément, et toujours dans un souci de contrôle de la présentation du DMM, une horloge globale (qui répond aux mêmes méthodes ci-dessus) est associée au document qui permet de suivre l'exécution de la présentation.

2.1.2 Mise en œuvre de la synchronisation

Deux grands courants se sont distingués pour la spécification temporelle dans les documents multimédias. Chacun d'eux met en évidence un type de modèle temporel supportant un ensemble de relations nécessaire à l'organisation explicite des composants d'un document dans le temps. Ce processus nommé **orchestration** est donc effectué selon un modèle basé sur les points ou selon un modèle basé intervalles (voir chapitre 1). Cependant, il a largement été prouvé, dans [Jourdan 98, Little 93, Willrich 01], que les modèles basés sur les intervalles sont les plus appropriés à la spécification de la composition temporelle des documents multimédias. Cette affirmation est justifiée par le grand nombre de relations offert par ce type de modèles (relations d'Allen) ainsi que par l'abstraction de haut niveau de la description de ces relations. C'est pour cette raison que nous avons adopté la modélisation par les intervalles temporels que l'on combine grâce aux relations d'Allen (egale, avant, durant, chevauche, rencontre, termine et débute).

Cependant, il a aussi été prouvé que ces relations de synchronisation sont principalement qualitatives, c'est à dire, qu'elles permettent de spécifier les contraintes entre les composants d'un document sans donner d'indications quantitatives sur ses contraintes. En effet, si l'on considère, par exemple, la relation **avant** (O_i , O_j) indique bien que l'objet O_i doit être présenté avant l'objet O_j , mais ne permet pas de préciser le délai à respecter entre la fin du premier objet et le début du deuxième.

De par l'importance de l'aspect quantitatif dans les DMMs, le modèle de document MDO2 adopte les relations d'Allen en les augmentant du paramètre "delai" qui permet de représenter l'aspect quantitatif des relations et modélise les contraintes numériques associées (Voir figure 2.5).

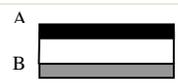
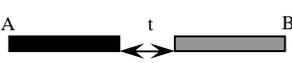
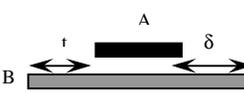
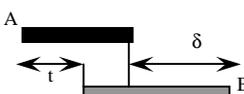
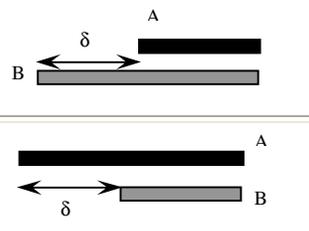
<i>Relation</i>	<i>Représentation Graphique</i>	<i>Contraintes Numériques</i>
A égale B		durée (A) = durée (B)
A avant (t) B		$t > 0$
A durant (t) B		durée (B) - durée (A) $\geq t$ & $t > 0$ & $\delta \geq 0$
A chevauche (t, delta) B		$t > 0$ & $\delta > 0$
A rencontre B		Tex (A) + durée (A) = Tex (B)
A termine B		Tex (A) + durée (A) = Tex (B) + durée (B)
A débute B		Tex (A) = Tex (B)

Figure 2. 5 – Relations d'Allen quantifiées

La relation représentative "Re" modélise la forme générale des relations de synchronisation temporelle dans MDO2 :

Re (C1, C2, /delai₁/delai₂) tel que :

- (i) $Re \in \{\text{avant, rencontre, egale, debute, durant, chevauche, termine}\}$;
- (ii) $C1$ et $C2$ sont les composants interagissant selon la relation Re ;
- (iii) $delai_1$ est l'attribut local associé aux relations *avant*, *durant* et *chevauche*. Il représente (en secondes) la durée qui sépare la fin du composant $C1$ et le début de $C2$ pour la relation *avant* alors que pour les deux autres associations c'est la durée à attendre après le début de $C1$ pour commencer à "jouer" $C2$. Notons qu'il n'y a pas de délai d'attente entre les médias reliés par les relations parallèles *egale*, *debute* et *termine* ainsi que la relation séquentielle *rencontre* ;
- (iv) $delai_2$ est l'attribut local associé à la relation *chevauche*. Il représente (en secondes) la durée qui sépare la fin du composant $C1$ de celle de $C2$.

La spécification **{avant (P, I, 5); termine (M, I); avant (P, M, 12)}** représente un exemple de spécification dans MDO2. Elle se décompose en trois relations qui permettent de présenter un poème (P) suivi après 5 unités d'une image (I) dont la présentation se termine à la fin de l'exécution d'une musique (M) qui elle même a débuté après 12 secondes de la fin de la présentation du poème.

L'étude que nous avons réalisée sur l'ensemble des relations de synchronisations proposées précédemment démontre qu'elles ne sont pas suffisantes pour faire face à l'imagination des auteurs pour composer les éléments d'un document entre eux. En effet, afin de modéliser une situation où l'exécution d'un objet implique le début de l'exécution d'un autre, l'ordre n'étant connu que suite à une interaction utilisateur ou à l'achèvement de certaines condition, laquelle de ces relations doit-on utiliser : *debute (A, B)* ou *debute (B, A)* ? La même situation peut se rencontrer avec les méthodes *rencontre*, *avant* et *termine*. Pour cela et afin de fournir à l'utilisateur une palette aussi complète que possible, nous avons introduit les méthodes ***one-rencontre***, ***one-avant***, ***one-debute***, et ***one-terme*** [Labeled 07] qui donnent l'avantage d'exprimer cette incertitude. La table de la figure 2. 6 qui résume toutes les relations de MDO2, présente les conditions associées à chacune de ces relations temporelles.

Une autre contrainte que l'on juge nécessaire afin de compléter la panoplie des relations proposées aux auteurs est celle du *choix*. En effet, arrivé à une (des) partie(s) de la spécification d'un document, le concepteur document peut s'offrir le plaisir d'établir

différents scénarios parmi lesquels l'utilisateur final (lecteur) fera son choix au moment de l'exécution (un exemple est présenté dans la section 4 de ce chapitre).

Cependant, en suivant le principe "*le confort de l'utilisateur prime*", nous avons remarqué que cette contrainte, lorsqu'elle est utilisée sans aucune restriction, peut produire un chaos que même l'auteur du document ne peut contrôler. Pour cette raison, MDO2 propose la méthode *alternative* qui relie entre eux au moins deux objets au plus trois qui correspondent aux scénarios de choix proposés. Cette contrainte répond à la spécification suivante :

alternative [$X_i / i = 1, 2, 3$] tel que $X_i \in \{\text{Obj_MDO2}\}$ et/ ou $X_i = \{\text{Re}_j(A, B) / j \in \mathbb{N}^+\}$.

On remarque que les composants, sur lesquels le choix doit porter, peuvent être de simples objets MDO2, des compositions synchronisées ou encore un panaché des deux.

Méthode	Conditions
avant (A, B, t)	$t > 0$ et $t_{\max}(A) < \text{Tex}(B)$
durant (A, B, t)	$t > 0$ et $(\text{Tex}(A) < \text{Tex}(B))$ et $(\text{duree}(B) - t) \geq \text{duree}(A)$
chevauche (A, B, t_1, t_2)	$t_1 > 0$ et $t_2 > 0$
rencontre (A, B)	$t_{\max}(A) \Rightarrow \text{Tex}(B)$
debute (A, B)	$\text{Tex}(A) \Rightarrow \text{Tex}(B)$
termine (A, B)	$t_{\max}(A) \Rightarrow t_{\max}(B)$
<i>egale</i> (A, B)	$((\text{Tex}(A) = \text{Tex}(B))$ ou $(t_{\max}(A) = t_{\max}(B)))$ et $(\text{duree}(A) = \text{duree}(B))$
one- {rencontre, avant, debute, termine} (A, B)	Dupliquer la relation dans le sens inverse. Comme exemple: one- rencontre (A, B) : $(t_{\max}(A) \Rightarrow \text{Tex}(B))$ ou $(t_{\max}(B) \Rightarrow \text{Tex}(A))$

Notes : $t_{\max}(X) = \text{Tex}(X) + \text{duree}(X)$ & le signe " \Rightarrow " force, selon le cas, le début ou la terminaison des exécutions (effet sur l'horloge locale).

Figure 2. 6 – Relations temporelles de MDO2

2.2 Composition spatiale

Comme présenté dans la section 1. 2. 2, la position d'un composant multimédia visible dans un document est spécifiée par les attributs `coin_sup`, `largeur` et `hauteur` définissant la position et la taille de la fenêtre qui lui sont associés. De plus, MDO2 offre des relations spatiales permettant de spécifier l'alignement d'objets média sur l'écran les uns par rapport aux autres au moyen de trois relations principales : **disjoint**, **superpose** et **remplace**, schématisées dans la figure 2.7.

La relation "disjoint" spécifie que l'intersection des positions des fenêtres associées aux objets constitue l'ensemble vide. Au contraire, les fenêtres des objets reliés par les relations **superpose** et **remplace** ont un espace commun. Cet espace est égal à la surface de la plus petite fenêtre des objets reliés par la relation "remplace". En effet, l'objet présenté en première position est remplacé en totalité (utilisation de **remplace**) ou partiellement (cas de **superpose**) par le deuxième. La combinaison de ces relations constitue une palette très suffisante pour modéliser toutes les compositions possibles d'alignement d'objets.

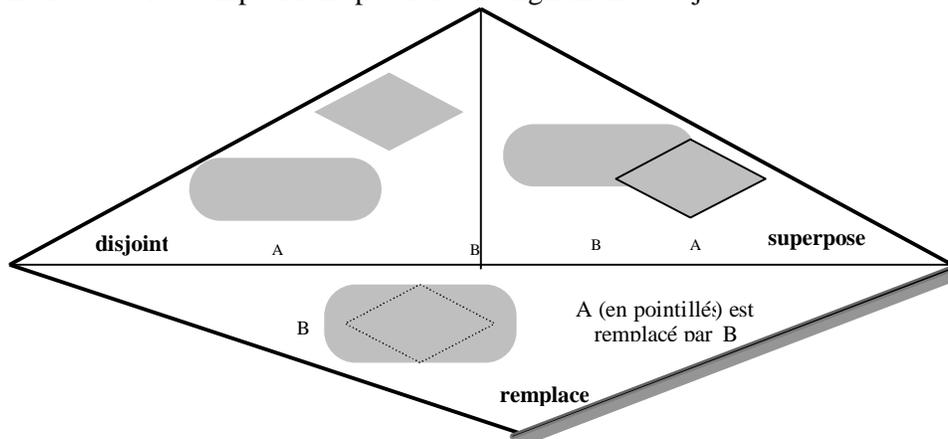


Figure 2.7 – Relations spatiales de MDO2

Si l'on reprend une image du chapitre 1 (figure 1. 2) et que nous la mettons à côté d'une autre image nous obtiendrons une composition assez simple (figure 2.8) par l'exécution de la spécification : **{disjoint (a, b); remplace (i₂, i₁); superpose (i₃, i₅); superpose (i₅, i₄)}** de façon à ce :

- § La partie "a" est la photo, en nuances de gris, du célèbre Mickey (i_1) qui en cache une autre (i_2) qu'elle remplace. L'existence de " i_2 " n'a de sens que si les deux objets " i_1 " et " i_2 " soient reliés par l'une des relations temporelles "avant" ou "rencontre".

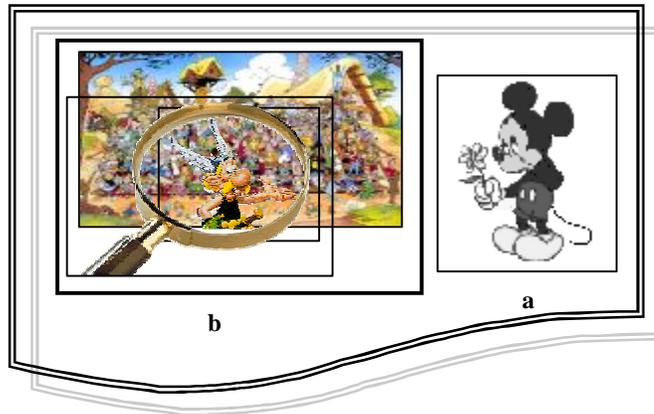


Figure 2. 8 – Exemple de médias "Image" assemblés

- § La partie "b" est la superposition de trois objets, toujours de type "image" : (1) une reproduction en couleurs des résidents du fameux village gaulois (i_3) ; (2) la photo d'une loupe (i_4) et (i_5) le dessin du valeureux guerrier Asterix. Dans cette partie et afin d'avoir l'impression qu'il y a eu zoom sur la présence d'Asterix dans la "photo de groupe", les trois parties ont été placées en superposition.
- § La partie "c" qui est le rassemblement des deux parties "a" et "b" placées à une certaine distance l'une de l'autre constitue le document que l'on a construit.

Nous avons jugé, selon les études que nous avons réalisé, que les trois relations présentées dans cette section sont suffisantes pour la modélisation des emplacements des médias visibles proposés par MDO2.

3 STRUCTURE DE PRESENTATION

La modélisation de la structure de présentation d'un document englobe la description des caractéristiques temporelles et spatiales ainsi que les interactions utilisateurs permises sur ce document et ses composants. Prenons l'exemple d'un utilisateur qui suit un concert de musique classique dans une application vidéo à la demande (de l'audio combinée avec des séquences vidéos). Il faut prendre en compte le fait que lors de l'exécution de cette application

l'auditeur décide à un instant donné d'interagir avec l'application par arrêt momentané ou un retour en arrière vers un mouvement très plaisant, etc. Cette interaction constitue une modification du scénario temporel du document exécuté.

Dans MDO2, une présentation particulière est exprimée par un ensemble d'objets (présentés dans les sections précédentes) qui lorsqu'ils sont créés, englobent les caractéristiques des médias qu'ils représentent. La manipulation de ces instances de classes se fait par l'invocation de méthodes (correspondants à des actions) et qui diffèrent d'un type de média à un autre. A première vue, certaines de ces méthodes permettent d'effectuer des changements sur les propriétés (tels que modification d'une taille de police) alors que d'autres (invoquées lors d'interactions) opèrent les contrôles admis sur les composants du modèle (tels que l'incrémentación du volume d'une audio –sans dépasser le nombre maximum de décibels permis par le dispositif de lecture-). A bien regarder, ces méthodes qui répondent à des interactions des utilisateurs ou encore correspondants à des stimulus émanant des ressources utilisées pour la présentation du document, peuvent être classés en deux familles selon le contrôle qu'elles réalisent : (i) le contrôle de la présentation ; et (ii) le contrôle de l'environnement.

3.1 Contrôle de la présentation

Par la notion de contrôle de la présentation nous distinguons toutes les interactions ayant un effet sur la durée de la présentation d'un objet multimédia et donc sur son horloge locale (voir section 2.1.1). Dans certains cas, ces interventions sont valables pour le document en entier (ç a d, sur l'ensemble des objets du document au même moment en se répercutant sur l'horloge globale du document) autant que pour un objet.

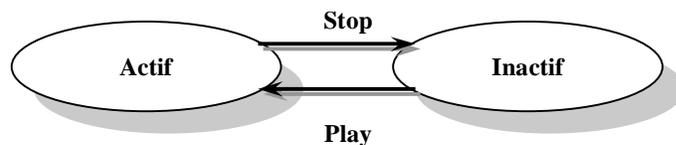


Figure 2.9 – Etats de base des entités multimédias

Les interactions effectuées sur les entités leur permettent de passer d'un état à un autre sachant que chaque entité se trouve dans un état *Actif*, pour désigner une entité en exécution au moment considéré et *Inactif* si l'exécution de l'entité est terminée ou qu'elle n'a pas encore

commencé. La figure suivante schématise les principaux changements qui s'opèrent suite à l'exécution des opérations de bases : Play et Stop (implémentée respectivement par les méthodes debuter et arreter) valables pour toute entité (multi)média : document, image, audio, script, etc.

Un état intermédiaire dit *Suspendu* est identifié pour toute exécution ayant commencée mais qui, pour certaines raisons, est momentanément stoppée. L'état "Suspendu" est clairement dissocié de celui de "Inactif" de par la gestion de l'horloge, dans le premier cas c'est la méthode "termine" qui est activée alors que dans le second c'est "figer" qui l'est.

L'état suspendu est identifié dans le cas des objets multimédias pré-établis (Scripts, documents SMIL, document Flash, etc.), identifiés dans MDO2 par objets composés. Sur ces derniers l'opération "Pause" peut être établie. Elle s'effectue via la méthode "figer" qui est inversée (annulée) par la méthode "reprendre" comme schématisée dans la figure 2. 10.

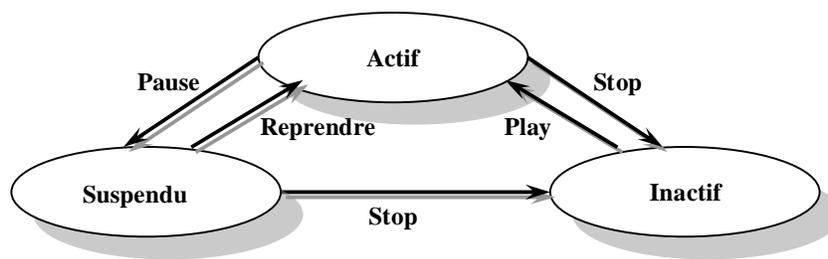


Figure 2. 10 – Etats et Actions sur objets composés

Note : Comme présenté ci-dessus, de l'état suspendu on peut passer vers l'état inactif pour terminer une exécution qui était suspendue mais il n'est pas concevable de suspendre une présentation qui n'est pas en cours.

Pour les objets des types audio-visuel (audio, video et clip), des actions permettant des retours arrière (Rewind: Rew), des avancements (Forward: FF) et des accélérations (Acc) sont établies dans MDO2 respectivement par les méthodes : arriere, avant et accelere (figure 2. 11). Dans ce cas les horloges locales sont, selon le cas, avancées ou retardées d'un certain nombre d'unités. Dès lors, si ces opérations ne sont pas admises pour les objets composés c'est que ces derniers disposent de leurs propres logiciels de présentations ce qui ne permet que des départs et des arrêts des horloges associées.

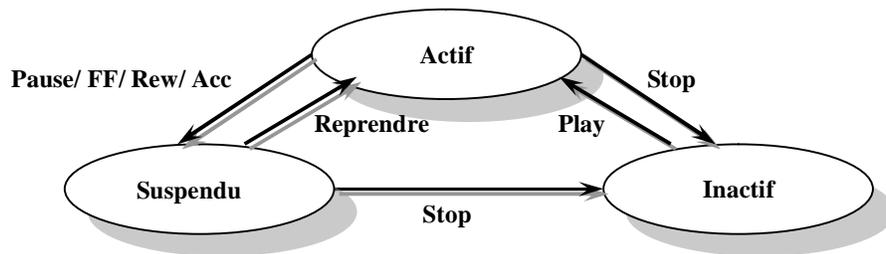


Figure 2. 11 – Etats et Actions sur objets audio-visuels

Note : Comme établi précédemment, le passage de l'état **Suspendu** à **Actif** se fait suite à l'action reprendre mais encore à la fin de l'exécution des actions FF, Rew et Acc. Une différence existe quant à la position de reprise du média.

3. 2 Contrôle de l'environnement

Le contrôle de l'environnement est relatif aux interactions ayant un effet sur les ressources utilisées tels l'écran (dimension spatiale) et le lecteur audio (propriétés sonores).

Dans le premier cas tout déplacement d'un objet visible à partir d'une position initiale identifiée par les propriétés de la fenêtre "*fp = coin-sup, largeur et hauteur*" se fait selon les axes : horizontal, vertical ou diagonal (principal ou secondaire), voir figure 2.12. Les méthodes MDO2 à invoquer au besoin sont respectivement : *dep_haut*, *dep_bas*, *dep_droite*, *dep_gauche*, *dep_diag_p* et *dep_diag_s* ; qui permettent via un calcul approprié un décalage de la fenêtre vers le sens désiré par l'utilisateur. Une autre interaction s'inscrivant dans le même volet est celle établie pour le re-dimensionnement de la fenêtre de présentation *fp* où les valeurs des composants des propriétés *coin-sup*, *largeur* et *hauteur* sont modifiées selon le besoin. Par exemple diminuer de moitié la hauteur d'une fenêtre de présentation revient à diviser la "hauteur" par deux.

Dans le cas des interactions sur les objets audio et/ ou clip la gestion du volume se fait par modification de la propriété volume (voir section 1.2) par l'invocation de l'une des méthodes : *mute*, *reprise*, *vol_plus* et *vol_moins* qui respectivement permettent de mettre en silencieux (en diminuant le volume à zéro), de reprendre la valeur avant le mute, de l'augmenter ou le diminuer d'un certain taux précisé par l'utilisateur.

4 ILLUSTRATION DE LA SPECIFICATION D'UN DMM DANS MDO2

Afin d'illustrer la spécification d'un document multimédia via le modèle objet MDO2, nous présentons le cas d'un petit document multimédia des prévisions météorologiques proposées par une chaîne de télévision en vidéo à la demande.

4.1 Spécification textuelle et graphique

Le document qui sera présenté dans un espace rectangulaire inférieur de quelques unités de la surface globale de l'écran, se compose de deux principaux éléments à présenter en parallèle : (1) une image statique (i_0) qui représente le logo de la télévision et qui s'affiche au début du document et ne disparaît qu'à la fin de son exécution ; et (2) une séquence de composants (mete0) dont la durée dépend de celles des composants séquentiels qu'elle inclue, à savoir :

- Un générique (gen_deb) décrivant la présentation en parallèle de deux objets médias : (a_1) une audio correspondant à la musique du générique et une page texte (Word) associée (t_1) se déroulant de bas vers le haut.

Délais et Contraintes : La présentation du texte se termine lorsque c'est le cas de l'audio qui est d'une durée de 45 secondes mais qui peut aller jusqu'à 60 secondes (sans les dépasser) en cas de retard. Lorsque le texte est présenté en entier, il se fige.

- Une séquence d'images satellites (sat) présentant séquentiellement trois vues envoyées par les satellites, et enregistrées sur le réseau, en début de matinée (i_1 : durée 45 secondes), à midi (i_2 : durée 90 secondes) et en fin de journée (i_3 : durée 76 secondes). Un enregistrement audio (a_2) accompagne les images qui sont présentées sur un fond d'image statique (i_4).

Délais et Contraintes : le fond d'écran remplace le texte immédiatement après la fin de l'audio du générique puis les images satellites sont à présenter (en superposition) au milieu de la fenêtre du moniteur. De plus, la présentation des images satellites est associée à celle de l'enregistrement audio qui les accompagne lui même composé de trois séquences (a_{21} , a_{22} et a_{23}).

- q Une partie (*scenario*) proposant des choix de scénarios:
 1. Une composition parallèle (*Comp*) constituée d'une vidéo (v_1) des températures de la journée avec un enregistrement audio (a_3) dont la durée est de deux minutes et 20 secondes (à savoir 140 secondes) en admettant une variabilité de ± 45 secondes ;
 2. Un clip (c_1) des prévisions du lendemain d'une durée égale à 125 secondes ;
 3. ou encore la présentation de (*comp*) suivie par celle de (c_1).

Délais et Contraintes : la vidéo (v_1), si elle est exécutée débutera avec l'audio à laquelle elle est associée et se terminera en même temps aussi. Si le choix est porté sur le troisième cas de figure, le clip (c_1) remplace la vidéo (v_1) et est donc présenté dans le même espace réservé.

- q Quelque soit le chemin suivi, un composant (*gen_fin*) terminant le document. Il est constitué d'un texte (t_2) d'une durée de 28 secondes incluant les informations pour contacter le service de météo et une musique de fin (a_4).

Délais et Contraintes : Un délai (d) de 15 secondes est respecté après l'exécution de (*scenario*) et avant de le remplacer sur l'écran par le générique de la fin.

Si l'on suit la représentation de ce document sur l'axe du temps (figure 2. 12) on remarquera que la présentation du DMM durera au maximum 609 secondes si le lecteur du document choisi de visualiser tous les composants sans exception (dans le cas où le choix se porte sur le scénario le plus court, la durée minimale serait de 379 secondes).

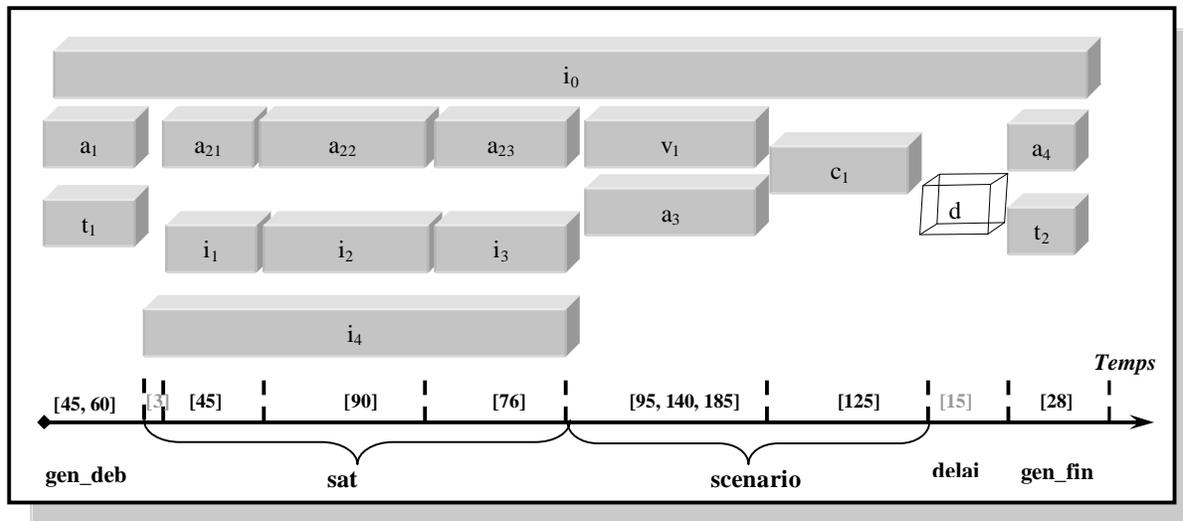


Figure 2. 12 – Positionnement des médias sur le Timeline

D'autre part, et bien que les séquences audio ne soient pas visibles, l'exécution de la présentation donnera lieu à des affichages qui se superposent et se remplacent selon les contraintes établies par le concepteur. Le schéma suivant (figure 2. 13) est un exemple de la disposition des différentes fenêtres (i_1, i_2, i_3) des média de la partie (*sat*) contenue dans le document et exposées sur une images de fond (i_4).

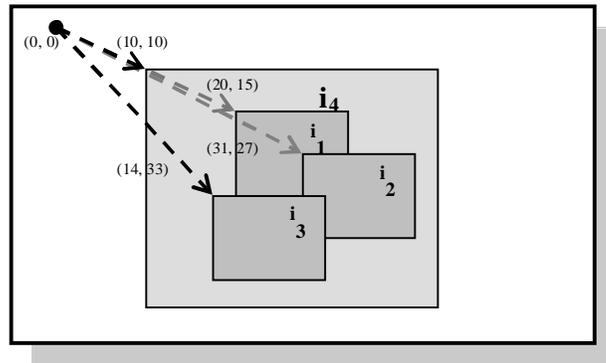


Figure 2. 13 – Positionnement des images satellites sur un fond en image

4.2 Modélisation en MDO2

La première phase de modélisation en MDO2 correspond à la dérivation des objets MDO2 qui constituent le DMM. La table suivante résume les médias utilisés dans cet exemple ainsi que leurs principales propriétés.

Les contraintes temporelles exprimées dans le texte peuvent être modélisées en MDO2 via l'ensemble des spécifications temporelles suivantes regroupées selon l'ordre chronologique des parties constituant le DMM :

‡ *La générique de début (gen_deb)*

debute (a_1, i_0)

termine (a_1, t_1)

rencontre (a_1, i_4)

‡ *La météo satellites (sat)*

durant ($i_4, a_2, 3$)

debute (a_2, i_1)

rencontre (a_{21}, a_{22})

rencontre (a_{22}, a_{23})

egale (a₂₁, i₁)

egale (a₂₂, i₂)

egale (a₂₃, i₃)

Type	Objet	Durée	Fenêtre (FP)	Autres
Texte	t ₁	[45, 45, 60]	{{(10,10), 50, 50}}	deplacement {fp, 2, (0, 0, 0, 5), faux}
	t ₂	[28]	{{(10,10), 50, 50}}	—
Image	i ₀	∞	{{(5,5), 5, 5}}	Durée indéterminée car l'image ne disparaît qu'à la fin de l'exécution
	i ₁	[45]	{{(20, 15), 25, 25}}	—
	i ₂	[90]	{{(31, 27), 20, 20}}	—
	i ₃	[76]	{{(14, 33), 10, 10}}	—
	i ₄	[214]	{{(10,10), 50, 50}}	—
Audio	a ₁	[45, 45, 60]	—	—
	a ₂	[211]	—	Synchronisation intra objets (a ₂₁ , a ₂₂ , a ₂₃)
	a ₃	[95, 140, 185]	—	—
	a ₄	[0, 28]	—	Peut ne pas être joué
Vidéo	v ₁	[0, 140, 185]	{{(10,10), 50, 50}}	Peut ne pas être joué
Clip	c ₁	[0, 125]	{{(10,10), 50, 50}}	Peut ne pas être joué

Figure 2. 14 – Objets MDO2 de l'exemple

‡ *Le choix des (scenario) et le générique de fin*

alternative [{ *rencontre* (a₂, a₃), *rencontre* (a₂, v₁), *termine* (a₃, v₁), *avant* (a₃, t₂, 15)},

{ *rencontre* (a₂, c₁), *avant* (c₁, t₂, 15)},

{ *rencontre* (a₂, a₃), *rencontre* (a₂, v₁), *termine* (a₃, v₁), *rencontre* (v₁, c₁),

avant (c₁, t₂, 15)}]

egale (t₂, a₄)

termine (a₄, i₀)

La synchronisation spatiale concerne, quand à elle, les objets visibles du document à savoir : les textes des deux génériques ainsi que les images, la vidéo et le clip des prévisions météorologiques. Les relations MDO2 de la figure suivantes modélisent ces contraintes de l'espace.

disjoint (i_0, t_1)	remplace (t_1, i_4)	superpose (i_4, i_1)	superpose (i_1, i_2)
superpose (i_2, i_3)	superpose (i_3, v_1)	superpose (i_3, c_1)	remplace (v_1, c_1)
	remplace (v_1, t_2)	remplace (c_1, t_2)	

Figure 2. 15 – Relations spatiales MDO2 de l'exemple

4.3 Interactions admises

En règle générale, les interactions utilisateurs admises dans un DMM dépendent du type de médias que l'y retrouve sauf si le concepteur d'un document précise ses propres contraintes de manipulation. Dans l'exemple modélisé précédemment, l'auteur n'a rien énoncé de particulier pour la visualisation et la gestion des objets utilisés sachant que ces derniers appartiennent aux deux types admis par le modèle proposé : des données dynamiques (des séquences audio, qui accompagnent la plus part des parties spécifiée, une vidéo ainsi qu'un clip) et des données statiques (texte et images). En conséquence, les contrôles pouvant être effectués sur les composants du document sont multiples et répondent aux règles établies dans la section 3 de ce chapitre.

5 CONCLUSION

La prise en compte de la complexité des données manipulées et des problèmes engendrés par une mauvaise orchestration, qu'elle soit temporelle ou spatiale, oriente les recherches vers des modèles de documents aptes à supporter toutes les contraintes engendrées. Dans ce chapitre, nous avons présenté un modèle orienté objet (MDO2) palliant les insuffisances soulevées dans les études des modèles existants. Ce modèle présente l'avantage d'être extensible d'un point de vue types de données, très riche en relations exprimant les contraintes et portable, caractéristiques souvent recherchées par les modèles et systèmes réalisés.

Ce modèle s'intègre dans une approche de conception de documents multimédias en cours de réalisation. La méthodologie suivie pour cela repose sur l'acquisition des besoins utilisateurs (éléments média, synchronisation,..) et la modélisation de ces spécifications en MDO2. Le chapitre suivant traitera la vérification de ces spécifications afin d'éviter les inconsistances qualitatives et quantitatives pouvant être associées au synchronisations temporelles établies dans un document.

La présentation d'un document multimédia dépend essentiellement des contraintes de synchronisation temporelle établies entre les composants de ce document à la phase d'édition. Ces contraintes spécifiées via des relations temporelles et causales dans le modèle MDO2, présenté dans le chapitre précédent, sont l'expression de l'intention de l'auteur et peuvent s'avérer incompatibles les unes avec les autres conduisant ainsi à des situations de disfonctionnement sinon de blocage. Cependant, elles ne sont pas les seules à déterminer le risque potentiel pour créer des situations inconsistantes durant l'exécution. En effet, des interventions effectuées par l'utilisateur final, lequel interagit à l'aide d'opérations de contrôle, peuvent aussi perturber ou arrêter l'exécution d'une présentation rendant ainsi le DMM inconsistant (voir chapitre 2).

Le contrôle d'intégrité nécessaire à la validation des documents permet donc de décider, à l'étape d'édition, si l'ordre des actions est consistant par rapport à un ensemble de contraintes ou non. Il est d'autant plus crucial lorsqu'il s'agit de documents avec scénarios de présentation complexes. La multitude d'événements et d'interactions pouvant s'effectuer durant une session peut alors créer un grand nombre de chemins d'évolution difficile à suivre au moment de l'édition si l'on considère toutes les implications possibles, le contrôle de cohérence est de l'ordre de complexité de $O(n^2)$ pour les relations d'Allen [Gerevini 02, LoPresti 02]. C'est pourquoi, il est nécessaire de fournir des outils pour aider les auteurs dans la conception de scénarios consistants ainsi que de vérifier et de valider l'intégrité de ceux qui existent déjà.

Aussi, pour représenter les propriétés de synchronisation des objets multimédias, il est essentiel de disposer d'un modèle permettant d'une part à l'auteur d'un document multimédia de spécifier sans ambiguïté des scénarios de présentations multimédias et d'autre part de les simuler et de les valider.

Dans ce chapitre nous présentons le modèle de **Réseaux de Petri Temporisé Multimédias avec Arc de Lecture** (*RdPT_MAL*), une variation des réseaux de Petri avec arcs de lectures qui ont été présentés dans un cadre non temporisé -successivement dans le temps- dans [Montanari 95, Janicki 95, Busi 99, Baldan 01, Vogler 02, Bouyer 06]. Nous adoptons

l'adaptation proposée afin de modéliser des scénarios de synchronisation de forte complexité dans les DMMs et de prouver leurs validités temporelles. Cette vérification se fait par l'intermédiaire de techniques conçues dans ce but et que l'on détaillera dans ce chapitre.

1 RESEAUX DE PETRI TEMPORISES AVEC ARC DE LECTURE

L'un des formalismes les plus intéressants pour la conception et l'analyse des systèmes à événements discrets est le modèle *des réseaux de Petri* (PN) [Brams 83, Diaz 01] étendu par le temps afin d'exécuter, sur ce modèle, la vérification (correctness) et analyse des performances. Ainsi, afin de modéliser les systèmes concurrents avec des contraintes temporelles et d'analyser leur fonctionnement et performances, plusieurs extensions de PN synchronisés ont été proposées, en particulier les différentes formes des réseaux temporels [Berthomieu 91, Merlin 74] et réseaux temporisés [Ramchandani 74, Sifakis 77]. Le principe de base étant de faire apparaître le paramètre temps dans le modèle, selon le besoin, en associant des temps (en durées fixes ou en intervalles) aux transitions, places ou arcs.

1- *Les réseaux de Petri temporels* (TPN pour *Time Petri nets*), introduits dans les années 70, associent à chaque transition un intervalle de temps [Merlin 74]. Une transition peut être mise à feu si la durée qui lui est associée est incluse dans son intervalle de tir et que le temps peut s'écouler sans désensibiliser d'autres transitions : le franchissement d'une transition peut dépendre du franchissement d'autres transitions même si elles ne partagent pas de places d'entrée ou de sortie, ce qui limite de beaucoup l'application des méthodes à ordre partiel dans ce modèle. De plus, avec ce besoin 'd'urgence', tous les problèmes importants deviennent indécidables pour les TPN illimités.

2- *Les réseaux de Petri temporisés* (TdPN pour *Timed Petri nets*), également appelés *les réseaux de Petri à arc temporisés*, associent à chaque arc un intervalle (ou un ensemble d'intervalles [Bag]) [Sifakis 77]. Dans les TdPNs, à chaque jeton est associé un âge qui, au début, est initialisé à (i) une valeur appartenant à l'intervalle de l'arc qui l'a produit ou (ii) zéro s'il fait partie du marquage initial. Puis ces âges évoluent en synchronisation avec le temps. Une transition peut être tirée s'il existe, dans la configuration actuelle, des jetons avec des âges inclus dans les intervalles de ses arcs d'entrées.

Note : Les jetons dit "vieux" peuvent mourir (ç à d, qu'ils ne peuvent plus être utilisés pour permettre le franchissement d'une transition mais ils restent cependant à leur place). Ces règles indiquent bien qu'au contraire des PN, les conditions de tir sont locales et ne dépendent pas de la configuration globale du système. Ce comportement 'paresseux' a des conséquences assez importantes.

Mais alors que d'importants problèmes, entre autres la couverture 'coverability', sont décidables dans les TdPNs, il s'avère que le problème d'accessibilité 'reachability' reste quand à lui indécidable. Dans [Bouyer 06] une autre faiblesse des TdPNs est présentée. En effet, les auteurs démontrent que les TdPNs ne possèdent pas le pouvoir d'expression qu'offrent les automates temporisés (ATs) et ne peuvent donc être transformés en AT (d'un point de vue de l'équivalence de langages). Afin de pallier cette faiblesse, une nouvelle extension des TdPNs est proposée qui se base sur l'introduction de la notion d'arcs de lecture "*les réseaux de Petri temporisés arcs de lecture*". Les réseaux de ce type, qui lorsqu'ils sont bornés sont équivalents au ATs, sont présentés dans la section suivante.

1.1 Présentation

Un Réseau de Petri avec Arc de Lecture (*RdPT_AL*) est un réseau de Petri temporisé étendu avec des arcs de lecture qui permettent de vérifier, sans les consommer, la présence (ou l'absence) de jetons dans une place. Ces arcs permettent donc de modéliser une opération de lecture de jeton (opération non destructive) par une consommation de jeton (opération destructive) suivie d'une régénération, dans la même place, du même nombre de jetons consommés (opération de réécriture). Cette modélisation permet un test de présence de jeton(s) spécifique(s) et joue un rôle important dans la représentation de la dynamique.

La partie quantitative de ces *RdPT_AL* est décrite par des contraintes temporelles établies sur les arcs. En d'autres termes, lorsqu'une transition est tirée, les jetons consommés sont ceux dont l'âge satisfait les contraintes temporelles associées aux arcs d'entrée après avoir vérifié si les contraintes spécifiées par les arcs de lecture sont satisfaites. Des jetons sont alors générés selon les contraintes établies sur les arcs de sortie. Une définition formelle est proposée dans [Bouyer 06].

Définition 1 : un réseau de Petri temporisé avec arc de lecture N est un tuple tel que :

$N = (P, T, M_0, Pre, Post, Read, \lambda, Acc)$ avec

- $P = \{p_1, \dots, p_i, \dots, p_n\}$ un ensemble fini de places;
- $T = \{t_1, \dots, t_i, \dots, t_m\}$ ensemble fini de transitions, avec $P \cap T = \emptyset$;
- $M_0 \in \text{Bag}(P)$: marquage initial des places ;
- Pré : application d'incidence avant ;
- Post : application d'incidence arrière ;
- Read : application d'incidence de lecture ;
- $\lambda : P \rightarrow \Sigma_e$ fonction d'étiquetage;
- Acc : condition d'acceptation exprimée par un ensemble fini de formules généré par la grammaire " $Acc ::= \sum_{i=1}^n p_i \succ k \mid Acc \wedge Acc$ " tel que $p_i \in P, k \in \mathbb{N}$ et $\succ \in \{\leq, \geq\}$.

Les valeurs du paramètre I représentent les intervalles associés aux contraintes temporelles associées au graphe ; alors que $\text{Bag}(\xi)$ est un ensemble (dont les éléments appartiennent à \mathbb{N}) obtenu suite à l'application d'une fonction de correspondance avec ξ comme ensemble de départ (il est possible que $\xi = \xi_1 \times \xi_2$, dans ce cas écrire $\text{Bag}(I)^P$ équivaut à écrire $\text{Bag}(P, I)$ par isomorphisme). La relation d'ordonnement quant à elle se présente est modélisée par l'expression : si $x, y \in \text{Bag}(X)$, alors $x \leq y$ si et seulement si $\forall e \in X, y(e) \geq x(e)$; et que si $y \leq x$ alors $x - y \in \text{Bag}(X)$ est défini par $\forall e \in X, (x - y)(e) = x(e) - y(e)$.

1.2 Propriétés et particularités

Un réseau de Petri avec arc de lecture étant un réseau temporisé, il conserve les caractéristiques et propriétés essentielles des TdPNs tout en respectant les extensions établies sur les TdPNs.

Partant de cet état de fait, la configuration V d'un RdPT_AL " $N = (P, T, M_0, \text{Pré}, \text{Post}, R, \lambda, \text{Acc})$ " est définie comme un marquage auquel sont rajoutées les informations d'âge relatives aux jetons du réseau. C'est donc un ensemble fini de paires (p_i, x_i) tel que p_i est la place où est disposé le jeton i auquel est associé un âge x_i . On dit alors qu'un jeton (p, x) appartient à une configuration v si $(p, x) \leq V$ (en terme de Bags). La configuration initiale $V_0 \in \text{Bag}(\mathfrak{R}_{\geq 0}^P)$ est définie par l'expression : $\forall p \in P, V_0(p) = m_0(p).0$ (ç à d, qu'il existe $m_0(p)$ jetons d'âge zéro à la place p).

La définition suivante a alors été proposée pour définir la sémantique d'un RdPT_AL, en terme de système de transitions :

Définition 2 [Bouyer 06] : soit $N = (P, T, M_0, \text{Pré}, \text{Post}, \text{Read}, \lambda, \text{Acc})$ un RdPT_AL. Sa sémantique est le système de transition $(Q, \Sigma_e, \rightarrow)$ avec $Q = \text{Bag}(\mathfrak{R}_{\geq 0})^P$, et " \rightarrow " défini par les propositions :

- Pour tout $d \in \mathfrak{R}_{\geq 0}$, $v \xrightarrow{d} v + d$ tel que $v + d$ est défini par $(v + d)(p) = v(p) + d$ pour tout $p \in P$.
- Une transition t est franchissable de v si pour tout $p \in P$, il existe $x(p), y(p) \in \text{Bag}(\mathfrak{R}_{\geq 0} \times I)$ tel que :

$$\left\{ \begin{array}{l} \pi_1(x(p)) + \pi_1(y(p)) \leq v(p), \\ \pi_2(x(p)) = \text{Pré}(t)(p) \text{ et } \pi_2(y(p)) = \text{Read}(t)(p), \\ \forall (\tau, I) \in \text{dom}(x(p)) \cup \text{dom}(y(p)), \tau \in I \end{array} \right.$$

En supposons que $z(p) \in \text{Bag}(\mathfrak{R}_{\geq 0} \times I)$ soit tel que

$$\pi_2(z(p)) = \text{Post}(t)(p) \quad \text{et} \quad \forall (\tau, I) \in \text{dom}(z(p)), \tau \in I,$$

définissent pour tout $p \in P$, $v'(p) = v(p) - x(p) + z(p)$;

alors $v \xrightarrow{I(t)} v'$.

Un chemin fini dans N est considéré comme acceptable s'il appartient à une configuration qui satisfait une des formules qui spécifient les conditions de passage (voir formule associée à Acc) ; alors qu'un chemin infini est accepté si chaque formule de " Acc " est satisfaite tout au long du chemin (Acc est alors considérée comme la généralisation des conditions de Büchi).

En résumé, un réseau de Petri temporisé avec arcs de lecture $N = (P, m_0, T, \text{Pre}, \text{Post}, \text{Read}, \lambda, \text{Acc})$ est :

- (i) un réseau de Petri temporisé (répondant à la définition standard dans [Boucheneb 02, Boucheneb 06] si pour toute transition $t \in T$, $\text{size}(\text{Read}(t)) = 0$, c à d, qu'il n'existe pas d'arc de lecture) ;
- (ii) k - borné si toutes les configurations V de toutes les séquences de franchissement de N se présentent telles que pour chaque place $p \in P$, $\text{size}(v(p)) \leq k$;

- (iii) borné s'il existe $k \in \mathbb{N}$ tel que N est k -borné ;
- (iv) sain s'il est 1-borné.

1.3 Exemple de RdPT_AL

La figure suivante représente un RdPT_AL. Ce réseau modélise une information produite par un serveur et consultée, de manière asynchrone, par des clients (c'est la transition "read").

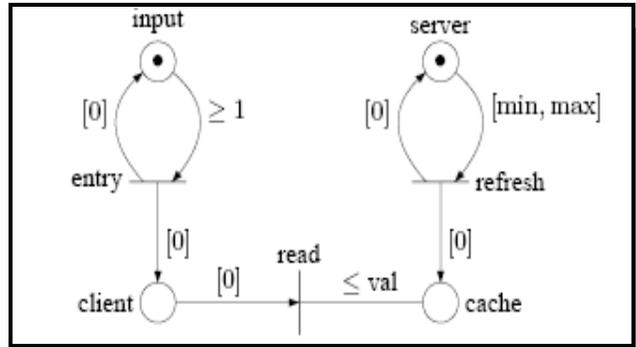


Figure 2. 16 – Exemple de réseau de Petri temporisé à arc de lecture.

Puisque l'information peut être désuète après une durée de validité "val", le serveur régénère périodiquement la valeur, mais la fréquence de cette régénération change selon la charge de travail du serveur (la transition "refresh"). La commande d'admission s'assure qu'au moins une unité de temps s'écoule entre deux arrivées de clients (transition "entry"). La présence de l'arc de lecture entre la place "cache" et la transition "read" permet de ne pas réinitialiser l'âge de la place "cache" lorsque la transition "read" est mise à feu.

2 DOCUMENTS MULTIMEDIAS ET RdPT_MAL

Dans Lumière, la structure temporelle d'un document est maintenue à deux niveaux :

- 0 un niveau au sein de la spécification MDO2 sous forme d'un ensemble de relations à base d'intervalles (voir chapitre précédent) ;
- 0 un niveau interne sous la forme d'un *réseau de Petri temporisé multimédia à arc de lecture* (RdPT_MAL) qui représente des contraintes temporelles aussi bien de synchronisation que d'interaction.

L'approche adoptée dans le système repose donc principalement sur la dérivation d'une spécification en réseau de Petri à partir de la spécification d'un DMM par une mise en correspondance entre la représentation de haut niveau à base d'objets de ce document et une spécification ayant pour principal objectif un contrôle parallèle des contraintes établies dans les scénarios. Puis, à partir de la spécification RdP, il est possible de réaliser la vérification formelle des propriétés de cohérence. Dans cette section, nous présentons les règles de construction du modèle interne avant d'exposer les techniques de contrôle de cohérence et de validation.

2.1 Traduction d'un document multimédia vers un RdPT_MAL

Dans le cadre de la vérification de documents multimédias, l'idée est d'utiliser la structure de réseau de Petri temporisé "RdPT_AL" afin de modéliser la synchronisation de médias auxquels sont associées des attributs temporels. La spécification proposée permet de modéliser, et de réaliser les contrôles, des scénarios indéterministes qui prennent en compte :

- (i) les délais sur les intervalles de temps associés aux médias qui composent le document ;
- (ii) les relations temporelles qui déterminent l'ordre selon lequel ces médias sont joués; ainsi que
- (iii) les éventuelles actions (interactions utilisateurs) ayant une répercussion sur l'ordonnement établi.

2.1.1 Correspondance et règles de construction

Un RdPT_MAL est constitué d'un ensemble de places, de transitions et d'arcs, chacun de ces composants ayant une correspondance avec une entité d'un document multimédia.

2.1.1.1 Les places

Dans Lumière, un document est un ensemble d'objets élémentaires (O_texte, O_image, O_audio, O_vidéo) et d'objets composites (O_clips et O_composés). A chacun de ces objets correspond une place où l'arrivée d'un jeton correspond au début de l'exécution du média.

Il existe plusieurs types de place dans nos réseaux de Petri :

- *Place atomique* : correspond à un média élémentaire et répond aux particularités ci-dessus, sauf dans le cas où elle est silencieuse, dans ce cas ce type (noté "ε") est nécessaire

à la spécification des attentes conceptuelles associées aux relations de synchronisation (voir chapitre 4).

- *Place composite* : représente un sous réseau contenant d'autres places atomiques et/ou composites. Ce type de place sert à modéliser la synchronisation de lèvre reproduite pour les objets composites. Dans ce cas, si un jeton arrive à la place composite, c'est la place d'entrée du sous réseau qui le reçoit, il n'est définitivement consommé que lorsque les jetons du sous réseau (produits durant l'exécution de ce sous- réseau) auront disparu.
- *Place de début (fin) de présentation* : modélise le point de départ (fin) d'une présentation multimédia. Son emploi permet d'appliquer les contraintes temporelles et de les ajuster aux besoins.

La figure suivante regroupe les représentations schématiques des différents types de places présentées ci-dessus.

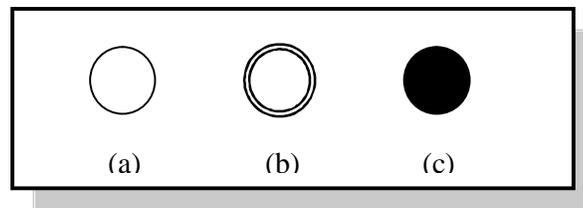


Figure 2. 17 – Places des RdPT_MAL

(a) Place objet ; (b) Place composite ; (c) Place début/ fin.

La présence d'un jeton dans une place (atomique ou composite) correspond au début de l'exécution (présentation) d'un média (respectivement un ensemble de médias). Cette présence est d'une durée incluse dans l'intervalle [**Min**, **Max**] avec une valeur nominale **Nomi** avec $\text{Min} \leq \text{Nomi} \leq \text{Max}$. Pour cela, à chaque jeton d'une place donnée est associé un âge (initialisé à 0 dans le cas général) qui s'incrémente avec l'évolution du temps, on parle alors de *place active* et de *jeton verrouillé*. A la désactivation de la place -correspondant à la terminaison (naturelle ou forcée) de l'exécution du média- *le jeton est débloqué*. On dit alors que le jeton est consommé si l'exécution se déroule jusqu'à la fin sinon il devient vieux et meurt. Dans ce cas, et bien que la transition dont il est un point d'entrée est franchie, il ne peut plus être utilisé mais il reste à sa place et conserve l'âge qu'il aurait atteint afin que ce dernier puisse être utilisé, au besoin, dans la gestion des interactions utilisateurs. Une configuration

est alors définie comme un marquage étendu par les informations d'âge des jetons modélisé par un ensemble de paires (p, x) représentant chacune un jeton d'un âge x sur place p .

2.1.1.2 Les transitions et les arcs

Dans notre modèle, une transition est, dans le cas général, franchissable si tous les jetons qu'elle consomme sont débloqués, c à d , ont une date comprise dans l'intervalle de temps spécifié par les arcs d'entrée. Sur chaque arc "place - transition" est donc placé un intervalle $[\min_A, \max_A]$ les bornes étant respectivement la date de tir au plus tôt (durée de vie minimale) et au plus tard (durées de vie maximale) du média modélisé par la place A . Les jetons produits par le franchissement d'une transition, selon les contraintes établies sur les arcs de sorties, prennent une date appartenant à l'intervalle de temps de ces arcs de sortie. Ainsi il est possible de tester la valeur d'une date et de la réinitialiser à une valeur choisie.

Note : Une transition est dite "*terminale*" si elle produit zéro jetons. Schématisée par une double ligne, elle modélise la fin de l'exécution d'un média "A" terminant son exécution parallèle avec un autre et n'apparaissant plus dans aucune autre relation de synchronisation.

Afin de faciliter la lecture des réseaux la schématisation a été allégée selon les règles ci-dessous :

§ omettre la représentation des intervalles temporels de valeurs $\min = \max = 0$ qui à l'origine doivent être représentés sur les arcs par $[0, 0]$;

§ abréger à " $[x]$ " la représentation des intervalles temporels de valeurs $\min = \max = x$.

Les arcs de lecture sont qu'à eux des arcs non dirigés qui permettent de vérifier l'existence d'un jeton sans le consommer. Ces arcs partants d'une place à une transition (les sens inverse n'est pas possible) permettent de forcer le tir d'une transition même si les jetons des autres places d'entrée ne sont pas disponibles, ce sont donc des arcs de priorité. Il est à noter que même un arc de lecture peut s'attacher un intervalle temporel afin de modéliser l'attente qui précède la vérification de l'existence ou l'absence d'un jeton (toujours selon le principe de l'attente conceptuelle).

2.1.1.3 Les représentations des relations de synchronisation

Pour chacune des relations de synchronisation proposées dans MDO2, nous proposons une représentation équivalente dans le modèle de RdPT_MAL. Il s'agit de correspondances

binaires entre la spécification des relations parallèles et leurs schématisations ainsi que celle entre les deux relations séquentielles et leurs représentations graphiques que l'on généralise afin d'obtenir la spécification du scénario en entier.

Ces représentations sont résumées dans les deux tableaux des la figure 2. 18 (pour les relations séquentielles) et la figure 2. 19 (pour les parallèles). On note pour les deux cas que 'A' et 'B' sont deux média et que t_A et t_B représentent, respectivement, la durée nominale de A et celle du média B, sachant que : " $\min_x \leq t_x \leq \max_x$ ".

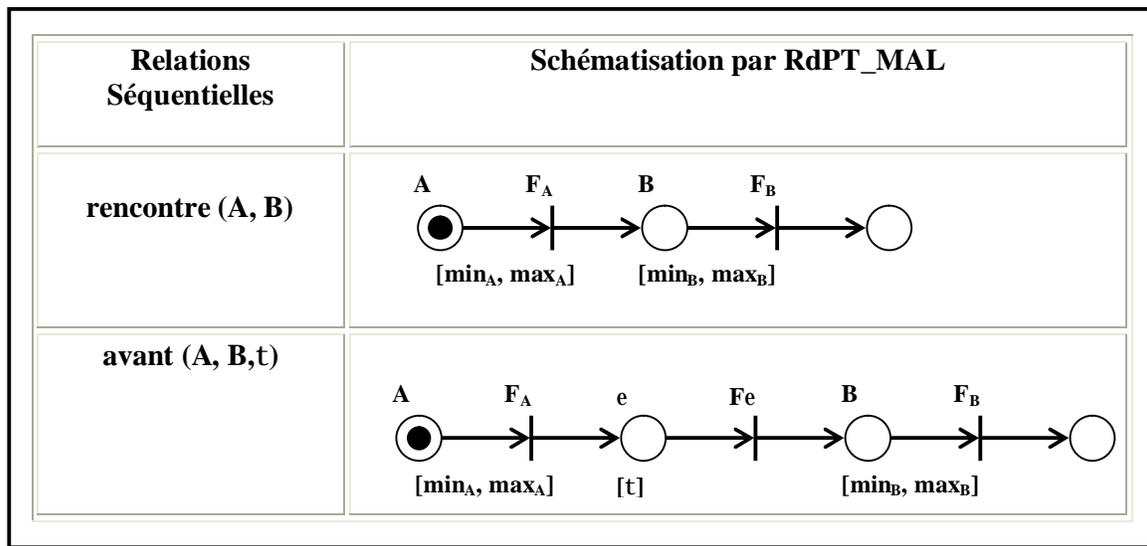


Figure 2. 18 – Représentation en RdPT_MAL des relations de synchronisation séquentielles.

Remarque : La schématisation de la relation **rencontre** peut être substituée par celle de la relation **avant**. En effet, il suffit que la valeur de la variable τ est fixée à zéro (pour formuler un délais nul) la représentation devient celle d'une relation séquentielle sans délais entre les média et la représentation devient alors équivalente à celle de la relation **rencontre** présentée ci dessus.

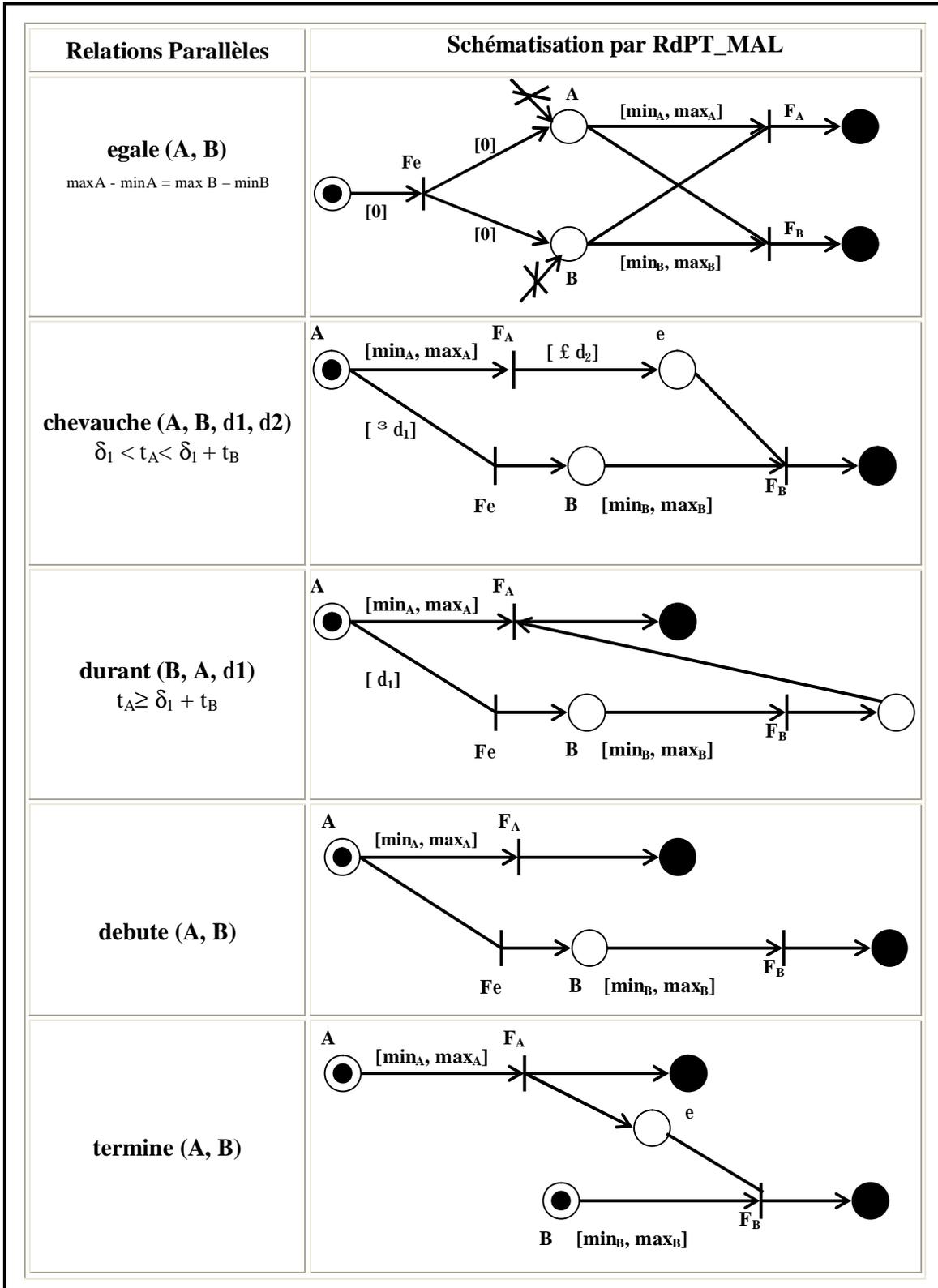


Figure 2. 19 – Représentation en RdPT_MAL des relations de synchronisation parallèle.

2.1.1.4 La définition formelle

Définition 3 : Un réseau de Petri multimédia avec arc de lecture "N" est formellement défini par le tuple :

$$N = \langle P, T, M_0, \text{Pré}, \text{Post}, R, A_{\text{elem}}, A_{\text{lect}} \rangle \text{ avec}$$

¶ P est l'ensemble des places pouvant figurées dans un RdPT_MAL.

$P = (DP, P_{\text{el}}, P_{\text{fi}}, P_{\text{at}})$ tel que :

- DP est une place unique dans un RdPT_MAL qui correspond au point de départ de tout document (début d'une présentation).
- P_{el} est l'ensemble fini des places élémentaires correspondant aux objets médias de la présentation.
- P_{fi} est l'ensemble fini des places finales correspondant, chacune, à la fin (normale ou forcée) de l'exécution d'un objet média (ou d'une synchronisation d'objets).
- P_{at} est l'ensemble fini des places définies pour marquer les attentes conceptuelles "ε" (cet ensemble peut être vide).

¶ T est l'ensemble fini des transitions pouvant figurées dans un RdPT_MAL.

$T = (T_{\text{el}}, \tau)$ tel que:

- T_{el} est l'ensemble des transitions élémentaires franchies à la fin de l'exécution des objets (ou la synchronisation d'objets) médias.
- τ est l'ensemble des transitions silencieuses " F_ϵ " définies pour marquer la fin des attentes conceptuelles

¶ M_0 est le marquage initial qui permet le déclenchement de la présentation. Il correspond à la présence d'un jeton dans la place unique DP et la mise à zéro des horloges globale et locales.

¶ M_f est le marquage final qui, s'il est atteint, représente une terminaison normale du document (pas d'incohérence et/ ou arrêt forcé de la présentation). Il correspond à la seule éventuelle présence des jetons morts dans les places du réseau.

¶ Pré est la fonction d'incidence avant ($T \rightarrow \text{Bag}\{P\}$).

¶ Post est la fonction d'incidence arrière ($T \rightarrow \text{Bag}\{P\}$).

¶ R est la fonction d'incidence de lecture (relative aux transitions avec arc de lecture en entrée).

¶ A_{elem} est la fonction d'étiquetage pour l'ensemble des règles de franchissement des transitions élémentaires associées aux contraintes établies lors de la synchronisation.

¶ A_{lect} est la fonction d'étiquetage pour l'ensemble des règles associées aux arcs de lecture et permettant le franchissement des transitions.

2.1.2 Exemple

Le scénario temporel de l'exemple "Météo", présenté dans le chapitre précédent, peut être schématisé en RdPT_MAL comme présenté dans la figure suivante.

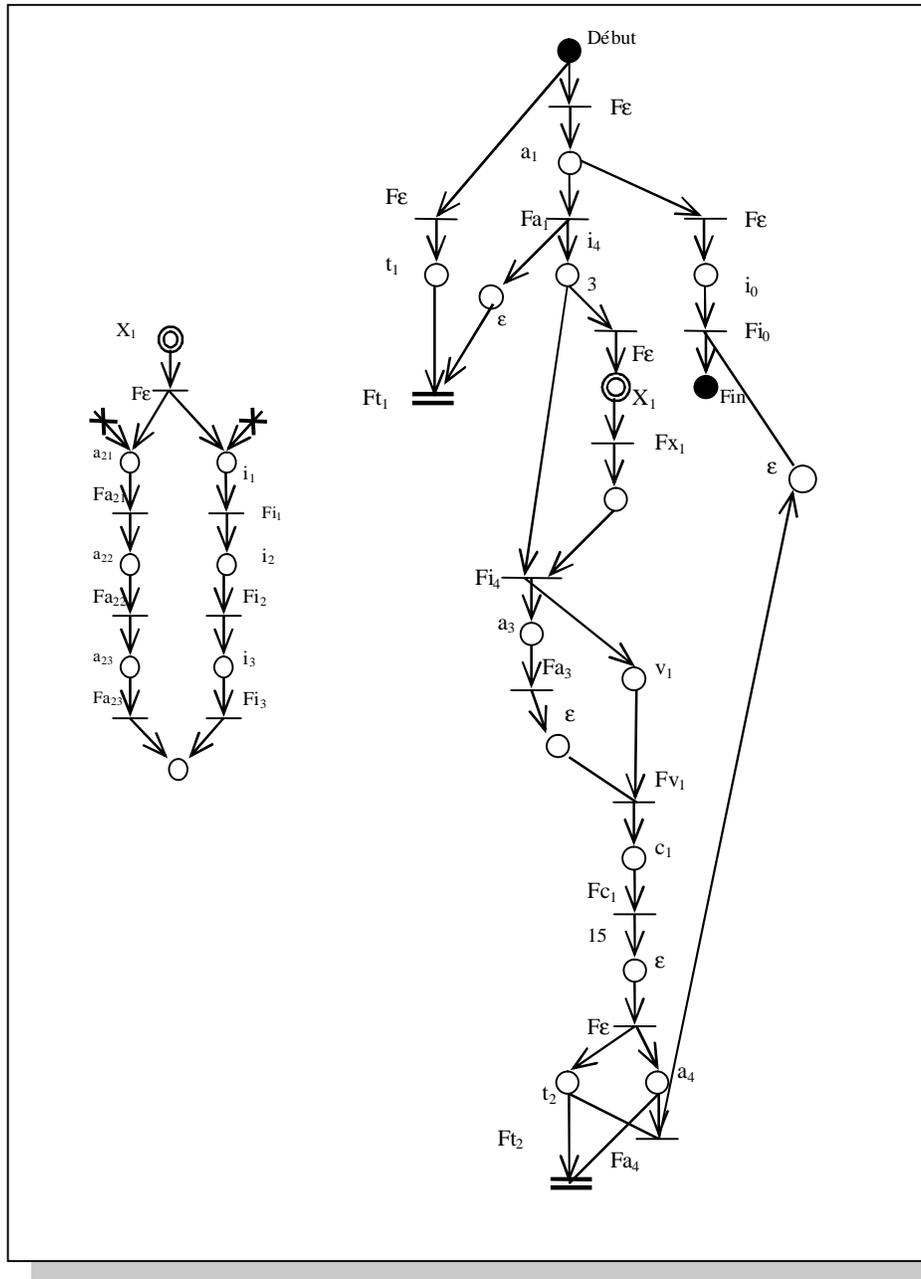


Figure 2.20 – Modélisation en RdPT_MAL de l'exemple

2. 1. 3 Gestion des interactions

Un document multimédia est représenté par l'ensemble des objets à "jouer" ainsi que les primitives de synchronisation qui leurs sont associées. Si une interruption se produit lors de la présentation de ce DMM on dit que le document s'est terminé dans un état incorrect.

Prenons exemple d'un vidéoclip (ensemble de scènes X_k , $k \in [1, N]$) qu'un lecteur (spectateur) a arrêté avant la fin pour passer à autre chose. Dans ce cas, l'interruption s'étant effectuée dans le temps de présentation d'une scène X_i , qui est gelée sur écran, toutes les scènes X_j qui suivent ($j > i$) ne sont pas activées et donc non visualisées. A première vue, tout paraît normal cependant à bien considérer les ressources utilisées pour le vidéoclip (par exemple, le canal utilisé pour l'acheminement des données) n'ont pas été libérés ce qui peut produire des gênes (retard, modification de la QoS, etc.) dans l'exécution de la prochaine requête de l'utilisateur.

Une exception à cette situation est faite pour toute terminaison (ou toute autre modification dans le temps) conséquence d'une interaction utilisateur prise en charge et gérée par le modèle interne et par la suite par le système d'exécution. Nous avons introduit au chapitre précédent les interactions admises pour le contrôle de la présentation lesquelles doivent être prises en considération au moment de la vérification du document.

Le modèle RdPT_MAL supporte donc la spécification de ces actions d'utilisateur, à savoir : Play (pour jouer), Stop (pour arrêter définitivement), Pause (pour arrêter momentanément avec Reprise pour reprendre ultérieurement), FF (pour avancer), Rew (pour revenir en arrière) et Acc (pour accélérer la vitesse de présentation). Pour cela des places (d'interaction et de type d'interaction) ainsi que des transitions dites d'interaction sont rajoutées au modèle afin de modéliser les processus possibles qui diffèrent selon les types des places- médias du modèle.

Lorsque l'utilisateur réagit pendant l'exécution d'un composant (multi)média, un jeton est déposé dans une place d'interaction qui, étant relié à une transition d'interaction par un arc de lecture, perd tout de suite son jeton qui permet de franchir la transition. Cette dernière produit un certain nombre de jetons (de 1 à 4) consommés par le processus approprié selon que l'interaction est, en ordre inverse, de type **Avancer** regroupant les interactions « ff, stop » cette dernière pouvant être considérée comme une avance rapide jusqu'à la fin ; **Reculer** pour le retour en arrière avec comme paramètre le taux de recul ; **Vitesse** qui permet de réaliser des

accélérations (positives / négatives) et donc ayant pour paramètres le sens de déplacement temporel et le taux décrété ; et enfin **Attente** pour les interactions « pause, play, reprendre ». La figure suivante résume le cas général où toutes les interactions sont exécutable sur l'objet.

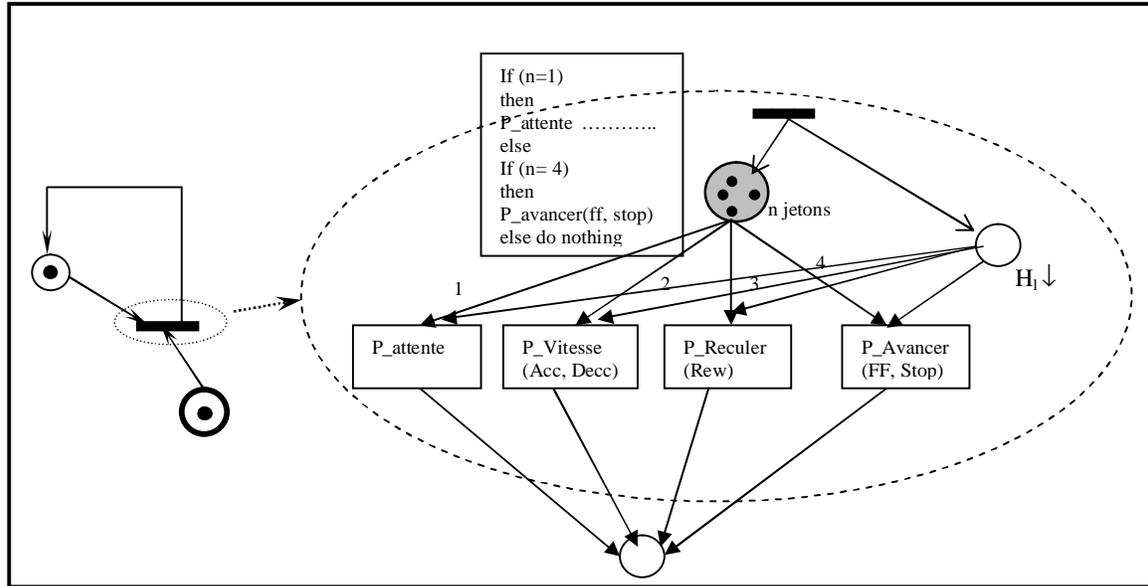


Figure 2. 21 – Processus d’interactions dans RdPT_MAL

On étend donc un réseau de Petri Multimédia avec arc de lecture présenté dans la définition 3 ci-dessus, par l’adjonction de deux types de places et d’un type de transitions :

- (i) P_{in} est l’ensemble fini des places d’interactions correspondant à des actions utilisateur (schématisés par des cercles en gras) ; et
- (ii) P_{ti} est l’ensemble fini des places type d’interaction définies pour établir le processus à exécuter selon le type de l’interaction établie par l’utilisateur à un moment donné (schématisés par des cercles gris).
- (iii) T_{in} est l’ensemble des transitions exécutant, selon son type, l’interaction établie à un moment par l’utilisateur (schématisé par un rectangle).

L’ensemble des places d’un RdPT_MAL est donc spécifié par : $P = (DP, P_{el}, P_{fi}, P_{at}, P_{in}, P_{ti})$; alors que l’ensemble des transitions $T = (T_{el}, \tau, T_{in})$.

On augmente aussi le tuple N par l’ensemble F qui est l’ensemble des processus établis selon l’interaction utilisateur. Chacun des processus d’interactions utilisateur consistent en un ensemble d’actions :

- Consommation du jeton associé à la place d'interaction ;
- Modification des valeurs des horloges concernées ;
- Traitement de l'interaction selon son type et le type de média sur lequel elle s'applique (speed, ...) en utilisant les paramètres appropriés ;
- Remise d'un jeton à la place média ayant reçu l'interaction.

Définition 4 : Un RdPT_MAL est donc défini dans la définition 3 mais étendu comme suit :

$$N = \langle P, T, M_0, Pré, Post, R, A_{elem}, A_{lect}, F \rangle, \text{ avec}$$

$$P = (DP, P_{el}, P_{fi}, P_{at}, P_{in}, P_{ti}) ;$$

$$T = (T_{el}, \tau, T_{in}) ; \text{ et l'ensemble des fonctions :}$$

$$F = \{\text{play, stop, rew, ff, pause, reprendre, acc}\}.$$

2.2 Analyse des scénarios temporels

Nous proposons des techniques d'analyse et de validation d'un scénario multimédia qui contient des objets médias ainsi que des relations de synchronisation. Ces techniques s'appuient sur l'analyse des propriétés du RdPT_MAL associé au scénario temporel. L'objectif est de détecter les incohérences possibles à l'étape d'édition des synchronisations puis en réponse à toute interaction utilisateur au moment de la présentation du document.

2.2.1 Propriétés vérifiées

Dans la littérature multimédia, plusieurs classifications de l'intégrité ont été proposées. Nous avons précisé dans le chapitre 2 de ce travail que nous avons adopté celle qui reconnaît les consistances : (i) **qualitative** (validité des relations de synchronisations entre objets), (ii) **quantitative** (validité des durées d'exécution des relations associées aux contraintes utilisateur), (iii) **générique** (validité des TAC) **et spécifique à l'application** (validité des contraintes sur ressources).

La vérification de ses propriétés par RdPT_MAL consiste en une étude structurelle du réseau selon des techniques d'analyse spécifiques. De plus, l'analyse du réseau construit à partir d'un scénario temporel doit inclure des procédés permettant de :

- s'assurer de l'inexistence de contraintes redondantes et/ ou inutiles (exemple : pas de relations transitives) ;

- 0 Editer la durée maximale d'une présentation dynamique ainsi que sa durée minimale, paramètre pouvant être utilisé dans la bonne gestion de la QoS proposée pour le document.

2. 2. 2 *Processus et techniques de vérification*

L'approche proposée pour la vérification des propriétés de cohérence temporelle d'un DMM repose sur le principe (prouvé dans [Bouyer 06]) d'équivalence des réseaux de Petri avec arcs de lecture bornés et les automates temporisés. Ces derniers ont fait l'objet d'un grand nombre de recherches particulièrement sur l'ensemble des analyses réalisées dessus [Aceto 03, Behrmann 04, Bouyer 02].

La vérification repose sur la dérivation d'un AT à partir du RdPT_MAL construit pour une présentation multimédia et puis de réaliser une analyse d'accessibilité de cet automate. Cette analyse peut être effectuée sur le graphe minimal d'accessibilité résultant où chaque *nœud* représente une classe accessible qui peut inclure un nombre infini de configurations selon la valeur du temps courant. Chaque arc de ce graphe correspond à une progression globale du temps et à chaque état de contrôle de l'automate peut correspondre zéro, un ou plusieurs nœuds dans le graphe d'accessibilité [Sampaio 03]. Les événements correspondent aux instants début et fin d'une présentation d'un média, aux interactions des utilisateurs ainsi qu'aux actions de progression du temps.

L'idée de base pour la vérification de la cohérence temporelle d'un document est d'analyser l'accessibilité de l'action caractérisant la *fin de la présentation* (celle qui inclue la place Fin). Nous savons qu'un chemin du graphe est cohérent si partant du nœud initial, il mène à l'occurrence de l'action représentant la fin (ç à d, une transition du graphe d'accessibilité étiquetée par l'action *end*). Nous pouvons donc affirmer que la synchronisation temporelle d'un DMM est cohérente si l'action caractérisant le début de la présentation (incluant la place Début) est nécessairement suivie, après un certain temps fini, par l'action déterminant la fin de la présentation du document (incluant la place Fin). Le calcul du temps d'exécution de la présentation peut être réalisé par l'utilisation d'une variable "*temps-effectif*" initialisée à zéro et incrémentée jusqu'à arrivée à l'action finale du graphe.

La dérivation et visualisation du graphe d'accessibilité est proposé et utilisé par un ensemble d'outils tels que ALDEBARAN [CADP], DOTTY [DOTTY], et ATG [ATG-FCTOOLS]. Il

faut cependant noter que ces outils ne permettent que de générer le graphe d'accessibilité. Pour effectuer l'analyse demandée, il faudrait donc implémenter dans notre système les algorithmes de vérification qu'il appliquera sur le graphe récupéré. Une meilleure solution qui s'est présentée à nous est l'utilisation de l'outil UPPAAL (Tool Environment for Validation and Verification of Real-Time Systems) qui permet de générer, en utilisant une interface graphique, un automate temporisé puis grâce à son simulateur et "model-checker" de vérifier les propriétés de l'AT [UPPAAL]. Le simulateur est un outil de validation qui permet très tôt, à l'étape de spécification, d'examiner l'exécution dynamique d'un système alors que le deuxième composant vérifie les invariants et les propriétés de vivacité. De plus, dans le cas d'une incohérence, le "checker" produit une trace de diagnostic pour expliquer pourquoi une propriété n'est pas vérifiée et permet donc de détecter rapidement et sans grands efforts la contrainte source de l'embrouillement.

2.3 Gestion de l'indéterminisme

Un document multimédia contient souvent des objets incontrôlables caractérisés par leurs durées indéterministes, comme un flot vidéo ou audio subissant les délais d'accès distants. L'existence de ce type d'objets au sein d'un scénario perturbe la synchronisation de la présentation du document. En effet, il devient plus difficile de vérifier statiquement la cohérence du document et d'assurer lors de sa présentation un formatage dynamique des objets qui permettent de respecter le scénario spécifié.

Deux propriétés caractérisent le scénario temporel d'un document multimédia : la *flexibilité* et la *contrôlabilité*. La *flexibilité* mesure l'aptitude d'un scénario temporel à pouvoir être adapté aux contraintes numériques induites par les spécifications de l'auteur tout en garantissant sa cohérence globale. Elle dépend de l'élasticité des objets média (définie par la différence entre la borne supérieure et la borne inférieure de leur durée contrôlable) présents dans le DMM. Cette flexibilité du document peut être mise à profit pour compenser le comportement indéterministe de sa présentation. Cela introduit la notion de *contrôlabilité* qui mesure l'aptitude d'un scénario à pouvoir être adapté dynamiquement, à l'exécution, au comportement indéterministe des objets médias qu'il contient. Il s'agit donc d'une généralisation de la notion de cohérence pour les scénarios indéterministes.

La gestion de l'indéterminisme est un processus assez compliqué qui n'a pas, à ce jour, trouvé de solution satisfaisante [Buchanan 05, Bulterman 05, Falkovych 06, Sabry-Ismaïl 99, Sampaio 03]. On ne propose pas de solution pour le traiter dans notre travail mais nous avons tenu à en parler de par son importance.

3 SYNTHÈSE

Le modèle réseaux de Petri multimédia avec arcs de lecture a été proposé pour la vérification des contraintes temporelles d'un DMM. Extension des réseaux de Petri temporisés avec arcs de lecture, il s'inscrit comme modèle noyau sur lequel s'applique l'approche de vérification que nous préconisons. Cette dernière s'appuie sur une génération du RdPT_MAL modélisant la présentation du document. Une analyse fonctionnelle est alors appliquée sur le modèle pour prouver la cohérence de ses synchronisations temporelles et au besoin de détecter les incohérences introduites par l'auteur. L'équivalence des RdPT_MAL et des automates temporisés nous a permis l'utilisation directe de l'outil UPPAAL qui génère automatiquement les résultats de l'analyse qu'il effectue sur l'automate.

Cet outil offre l'avantage de la vérification pas à pas, ç à d, au fur et à mesure de la construction de l'AT et permet un gain de temps par l'automatisation de l'analyse. La limitation de son utilisation se présente dans la construction manuelle de l'automate à analyser. Une solution serait de réaliser une passerelle qui accomplirait la génération automatique des automates à partir de spécifications des RdPT_MAL (ou d'autres type de RdP équivalents).

Lumière a été imaginé et créé dans un souci d'aide aux utilisateurs (concepteurs et lecteurs) en présentant une interface facile à manipuler pour toute personne sachant manier l'outil informatique de base. Dans ce chapitre, nous présentons une vue globale de ce système auteur dédié à l'édition et la présentation des documents multimédias et pouvant être utilisé dans un environnement interactif. Il a pour noyau un modèle de document caractérisé par son appartenance à l'école orientée objet ce qui lui permet la composition et la gestion temps réel d'un grand nombre de composants tout en permettant, à différents niveaux, des modifications (en rajouts ou suppression) ainsi que des vérifications assurant la validité des présentations ou documents produits. Dans ce chapitre, qui constitue une synthèse de la thèse, nous présentons (i) les principes généraux ou la démarche adoptés pour généré puis présenté un document (ce que l'on a nommé cycle de vie d'un DMM) ; (ii) l'architecture globale, qui s'appuie principalement sur le modèle de document et le modèle interne de validation qui lui est associé ; ainsi que (iii) l'interface utilisateur proposée. La validation du travail s'appuie sur la réalisation d'un environnement opérationnel dont nous décrivons les principales caractéristiques dans les sections de ce chapitre.

1 CYCLE DE VIE D'UN DOCUMENT LUMIERE

Lumière prévoit d'offrir un grand nombre de services qui permettent de manipuler un document électronique depuis sa création jusqu'à sa présentation et va même plus loin en proposant des services pour sa sauvegarde et permettre ainsi une restitution ultérieure à la demande. En effet, l'auteur d'un document peut s'intéresser au contenu et à l'organisation du document, ou encore son intérêt sera fixé sur la manipulation et la disposition des éléments à visualiser sur écran ou encore sur l'enchaînement des éléments dans le temps. A partir de la présence de ces multiples facettes et afin d'éviter l'un des reproches les plus adressés à la majorité des systèmes auteurs de par l'absence de processus de développement et gestion et implicitement d'un manque de garantie de la qualité des documents qu'ils produisent, que nous proposons une méthodologie d'« authoring » qui assurera une QoS répondant aux

critères établis par les utilisateurs de Lumière. Ce processus simple, comme nous le résume la figure 2. 22, se résume en cinq étapes principales :

Etape 1- *la spécification du document en utilisant MDO2* : Ceci permet de modéliser explicitement la structure du contenu du document avec les éléments de synchronisations temporelles et spatiale constituant le squelette du document ainsi que les règles auxquelles doit obéir la présentation des objets médias assemblés. Une vérification continue de toutes les propriétés énoncées est respectée tout au long de la construction du modèle de document.

Etape 2: *la spécification temporelle formelle* : la démarche adoptée à cette étape consiste à mettre en correspondance d'une part la modélisation objets des contraintes temporelles et le modèle Rdp_MAL (Voir chapitre 5), et d'autre part la spécification objet et un modèle formel basé « matrices dynamiques » [Labeled 99, Labeled 04a]. Ces instanciations de modèle sont des entrées pour l'étape suivante dédiée à la validation et estimation de coûts des présentations multimédias.

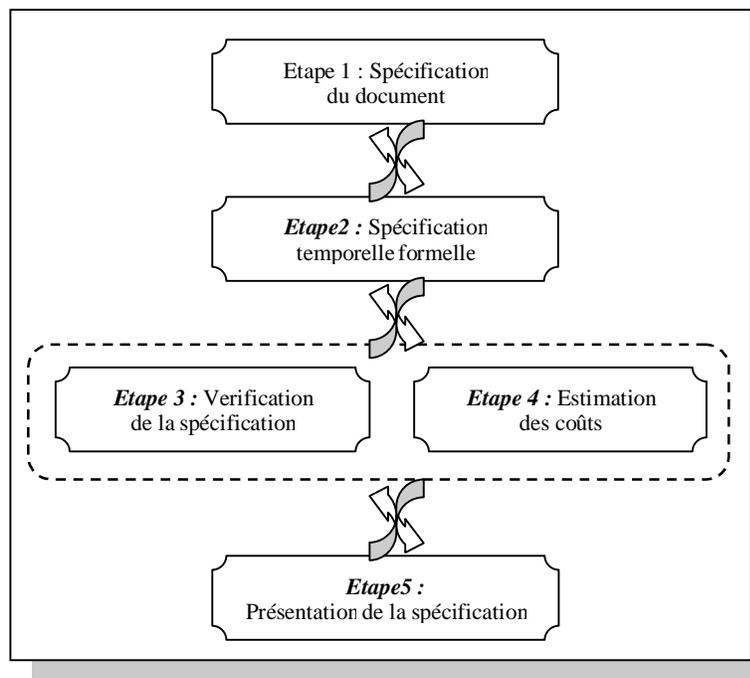


Figure 2. 22 – Etapes de l' « authoring »

Etape 3 et 4 : *la vérification de la spécification et l'estimation de ses coûts* : La dualité de la correspondance permet une application parallèle des algorithmes de validation sur les scénarios temporels et d'estimation de coûts temporels (ou autres selon les paramètres

d'exécution) nécessaires à la présentation du document selon les contraintes énoncés par l'auteur et conduit à une meilleure prise en charge de la QoS offerte à l'utilisateur dans un souci de gain de temps.

Etape 5: Présentation de la spécification : les médias orchestrés par l'auteur et vérifiés à l'aide des modèles associés sont présentés dans une interface interactive qui permet aux lecteurs des contrôles de présentation par l'ensemble des opérations d'interactions permises.

Lumière permet aux auteurs d'un DMM de visualiser, tout au long du processus de conception, le résultat supposé de l'exécution de la présentation. Ceci permet des retours arrière (à l'étape 1) afin d'effectuer des ajouts/ suppressions de médias ou contraintes sur les éléments mais aussi des modifications de propriétés dans le cas où le comportement de la présentation ne répond pas aux attentes des auteurs et/ou lecteurs des documents. Toute mise en cause d'un élément ou propriété est automatiquement répercuté sur l'ensemble des modèles sans nécessité une remise à zéro des spécifications ce qui constituerait une perte de temps. Cette propriété caractérisant le système est l'un de ses points forts qui le démarquent de ceux proposés à ce jour.

2 ARCHITECTURE DE REFERENCE DE LUMIERE

L'architecture générale de Lumière est illustrée dans la Figure 2.23. Le système se compose de trois principales parties interagissant tout au long d'une session d'édition et de présentation d'un DMM : (1) le système d'édition (Editeur) ; (2) le système de présentation (Exécuteur) ; et (3) le composant dédié à la validation (Valideur).

Les différentes fonctions d'édition, de présentation et de contrôle (qui incluent la recherche des données (multi)médias à intégrer dans le document) sont rendues accessibles à travers différentes interfaces utilisateur graphiques de simple emploi. C'est à travers ces interfaces que l'utilisateur (qu'il soit auteur ou lecteur) peut créer un document, interagir avec ses éléments, contrôler sa progression et même le modifier. Il peut aussi le charger s'il existe déjà et qu'il est sauvegardé en local ou sur la toile.

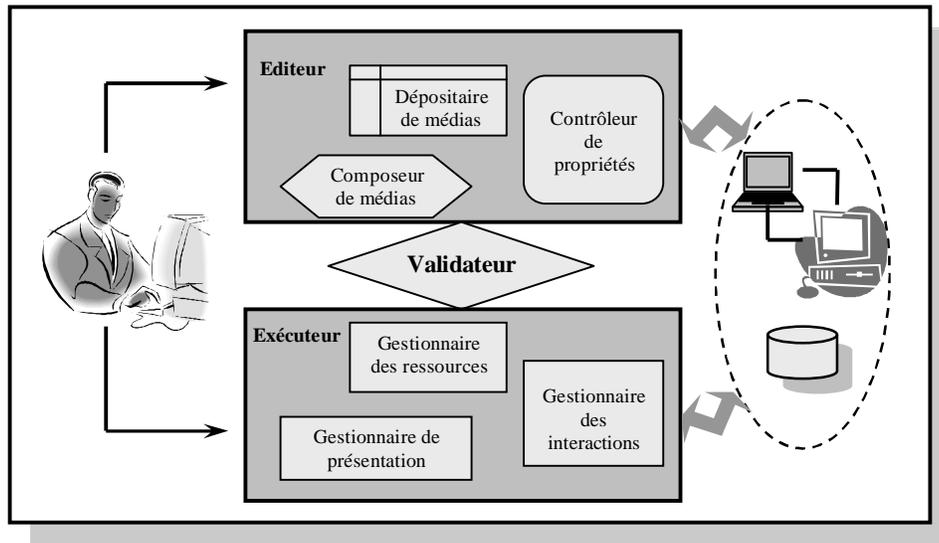


Figure 2.23 – Architecture de référence de Lumière

2.1 Editeur

Une édition rapide est facile des documents multimédias s'inscrit comme l'un des buts essentiels de notre proposition. C'est pour cette raison que Lumière offre aux utilisateurs une interface confortable permettant une édition incrémentale qui facilite au niveau du "*Composeur de médias*", la spécification des objets média auxquels sont associées des propriétés et des relations de synchronisation ainsi que les interactions utilisateurs autorisées sur chacun d'eux (étape 1 du cycle de vie). L'étape d'édition n'est en aucun cas effectuée par programmation mais seulement par :

- (i) identification de chaque objet selon son type (audio, image, etc.) et des opérations d'interactions permises sur le média ainsi reconnus ;
- (ii) saisie des propriétés (nom, source, etc.) aux éléments identifiés ;
- (iii) mise en place des relations temporelles et/ou séquentielles reliant les objets entre eux ;
- (iv) placement spatial déterminant l'apparence graphique du document par mise en place des relations de synchronisations reliant tout objet, autre qu'audio, aux autres médias de la présentation.

Le modèle de document associé à la spécification informelle est répertorié dans le

simulation en miniature du résultat obtenu selon les médias instanciés. Cette fenêtre peut être visualisée à l'écran à la demande de l'auteur et à chaque fois qu'il le demande. La figure 2.25 est un exemple obtenu suite à la spécification du texte et des deux images de la première partie de l'exemple de l'agence de voyage ci dessus.

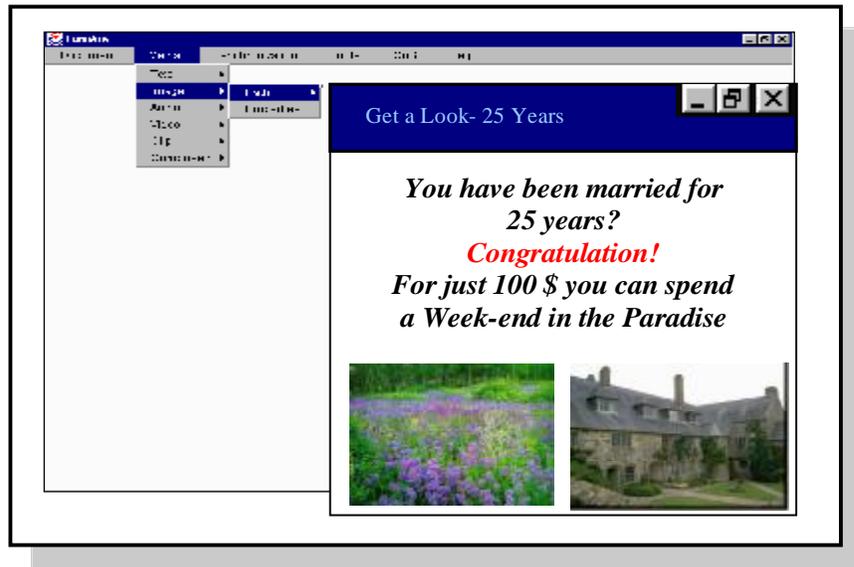


Figure 2. 25 – Vue avant exécution

2.2 Valideur

Le "*Valideur*" assure les étapes 2, 3 et 4 du cycle de vie d'un DMM. Par des règles de passage (voir chapitre 5), le composant "*Vérificateur*" génère le format interne du document modélisé par un RdP_MAL associé à la spécification temporelle du document et applique les algorithmes de validation qui permettent de prouver la cohérence des contraintes établies sur les médias. Les marquages établis lors de la validation sont gardés pour être utilisés en cas de changement ultérieur dans la spécification. Dans le cas où le système détecte une incohérence le "*Résolveur*", composant du valideur (voir figure 2.26), en s'aidant d'une interface appropriée, communique le problème à l'utilisateur qui choisi parmi les ébauches de solutions produites par le système (en utilisant les propriétés telles que le composant d'intérêt et les stratégies de relaxation) ou décide d'ignorer les propositions offertes à lui et de revenir vers la spécification pour modifier le document selon de nouvelles règles. La validation est alors reprise, selon l'historique sauvegardé, à partir du point de modification ce qui permet de ne pas retourner plus loin qu'il ne faut et ainsi de gagner du temps.

Dans le même esprit de recherche de la meilleure solution correcte, le validateur inclut un module "*Estimateur*" qui génère (par des règles de passage) un modèle en matrices dynamique (MD) permettant, par des algorithmes précis, de réaliser des estimations de coûts [Labeled 99, Labeled 04a]. Les calculs obtenus peuvent servir comme paramètre de négociation pour aider l'utilisateur dans sa quête d'une QoS selon les contraintes qu'il se pose ou qu'il rencontre.

2.3 Exécuteur

Dans d'autres systèmes auteurs, ce composant est dit "machine d'exécution". Il assure les trois fonctions principales de la cinquième étape du cycle de vie d'un DMM :

- la gestion d'accès qui se charge de récupérer les objets médias (et leurs propriétés dans le cas des médias dynamiques) locaux et/ ou d'appliquer les protocoles appropriés afin de récupérer les objets distants et de les charger dans un tampon qui se vide à la fin de l'exécution du média (exception est faite pour les médias qui sont joués plus d'une fois dans la même présentation si le tampon admet leur présence). Le responsable de la bonne marche de ces fonctions est le "*Gestionnaire d'accès*".
- la gestion des périphériques et des logiciels externes de présentation. Elle se charge :
 - d'initialiser et de contrôler par le "*Gestionnaire des ressources*" les périphériques nécessaires à l'exécution de la présentation (écrans pour les données visibles et périphérique audio pour les données audibles) ;
 - de fournir aux utilisateurs les moyens par lesquels il peut effectuer des interactions qui permettent de modifier des paramètres comme le volume audio, la taille de la fenêtre de présentation, etc. et de gérer ses interactions ("*Gestionnaire d'interactions*") ;
 - de faire appel aux logiciels de lecture d'objets approprié au type *composé* avec formats externes tels que les éditeurs PowerPoint, SMIL, etc. Dans le cas où le programme approprié n'est pas disponible en local (ni même après recherche sur le net en cas de travail en connexion), le système propose au lecteur, qui peut refuser l'offre, l'utilisation du logiciel le plus proche offrant la possibilité d'ouvrir le média (exp. VLC Media Player pour les séquences audio et vidéo). Cette propriété permet de renforcer la portabilité de Lumière sur les différentes plateformes.

- la gestion de l'exécution englobe un ensemble de fonctions permettant de jouer le document multimédia à partir des spécifications validées. Dans Lumière, le processus de présentation d'un document est réalisé par le "*Gestionnaire de présentation*" qui exécute les deux fonctions suivantes :
 - la synchronisation entre les objets média : cette fonction consiste à ordonner la présentation des objets selon le scénario spécifié ;
 - la gestion de l'indéterminisme de la présentation : cette fonction consiste à détecter les désynchronisations qui se produisent durant l'exécution d'un objet et de rétablir dès que possible la synchronisation initiale.

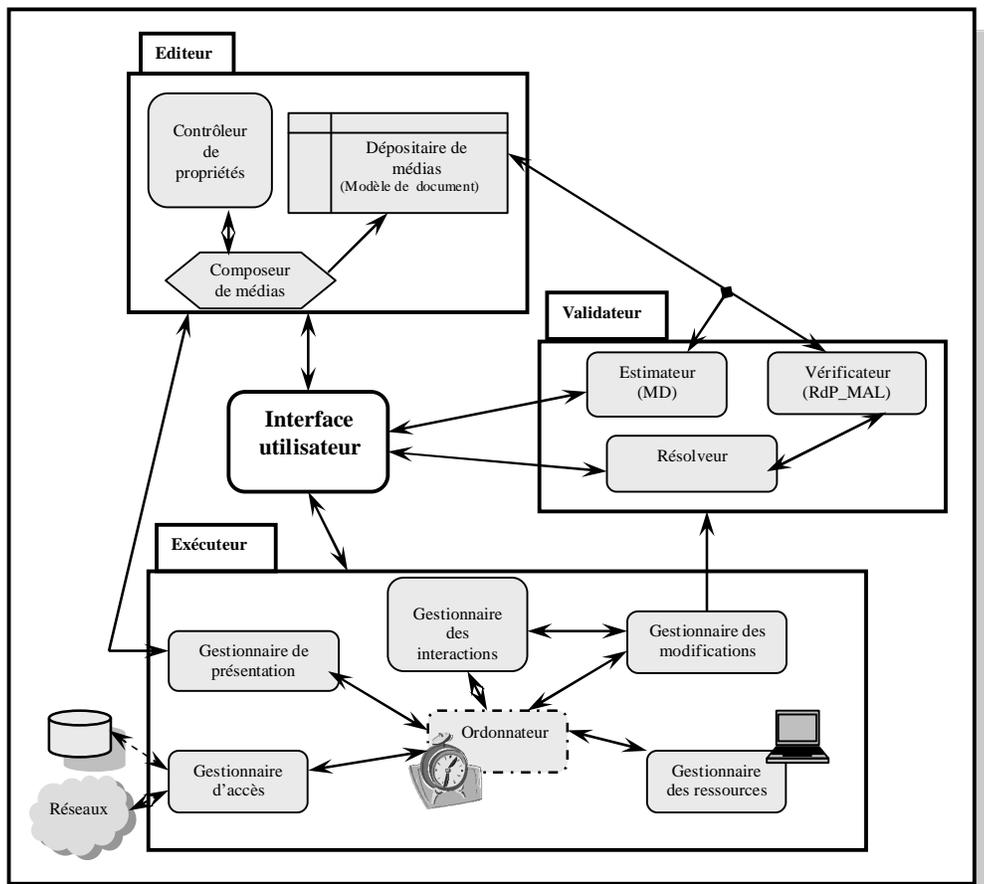


Figure 2. 26 – Architecture de référence détaillée du système Lumière

La coordination des différentes activités de présentation d'un document est assurée par le composant "*Ordonnateur*" qui assure la progression temporelle et donc la supervision de la présentation. Ce module gère les horloges associées aux médias ainsi que l'horloge principale associée au document. Il initialise à zéro l'horloge document et accorde sa progresse au (i)

temps relatif au document et aux différents objets et (ii) au temps réel (mesuré par rapport à l'horloge de la machine). Cette horloge produit (avancer l'horloge d'un média) et reçoit (fin d'une exécution, ç à d, une horloge média est arrivée à terme) des événements qui peuvent déclencher de nouvelles activités de présentation ou l'arrêt de celles qui sont en cours. Elle permet aussi de contrôler les flux de données et assurer au maximum la qualité WYSISWIG.

L'exécuteur inclut aussi un "*Gestionnaire de modification*" qui est activé suite à toute modification de la présentation en cours de l'exécution du DMM. Les modifications proposées (suppression d'une contrainte temporelle) est véhiculée vers le *Valideur* qui réalise les changements nécessaires sur ses modèles formels et reprend la validation comme présenté dans la section précédente.

La figure précédente résume les composants de l'architecture de référence détaillée de Lumière.

3 SYNTHÈSE

Dans ce chapitre, nous avons présenté le système auteur Lumière dédié à la conception et la présentation de documents multimédia. En particulier, nous avons présenté le cycle de vie d'un DMM pouvant être produit par ce système ainsi que son architecture générale et les différentes fonctions de chacun de ses composants.

Nous avons remarqué que la gestion de la synchronisation est centralisée (ordonnateur). Cette stratégie permet de gérer la synchronisation à gros grain puisque à ce niveau, la maîtrise globale du scénario est nécessaire. Par contre, la gestion distribuée est utilisée pour la synchronisation fine du fait de la localité de cette synchronisation et afin de ne charger le l'ordonnateur qu'en cas de nécessité. De plus, la gestion distribuée s'adapte bien à la structure d'objets média adoptée dans Lumière.

De plus, nous avons vu que le système, qui s'appuie sur une interface simple, facilite la réalisation de l'édition incrémentale car il permet de vérifier la cohérence des contraintes après chaque modification élémentaire du document. L'extensibilité et la portabilité du système sont aussi des points forts à ne pas négliger.

La richesse du multimédia est un avantage pour les utilisateurs qui entrevoient des systèmes tout en un avec une maîtrise plus complète de leurs créations multimédia et la faculté de bénéficier des innovations les plus récentes. Elle se révèle par contre être une difficulté quand il s'agit de concevoir des systèmes multimédias, en particulier des documents électroniques, car il faut intégrer des médias différents tout en donnant une vision claire de comment les utiliser et les gérer. L'implémentation de ces systèmes sur plusieurs plateformes informatiques est un autre aspect difficile, puisqu'il nécessite de mettre en correspondance les fonctionnalités logiques avec les équipements matériels, qui peuvent être variables d'une plateforme à l'autre.

Le contexte du multimédia étant en progression et enrichissement continu, une multitude de problèmes se pose à tous les niveaux et les recherches se poursuivent dans différentes voies (technologiques, méthodologiques, etc.). Nous avons choisi de contribuer, dans le domaine de la présentation multimédia, à réaliser un outil de spécification, de validation et de génération de documents multimédias utilisés dans un contexte distribué et interactif. La construction d'un tel environnement est soumise à des contraintes multiples. Ces contraintes sont liées d'une part à la multitude de formats de spécification de documents multimédias, et d'autre part aux différents objectifs et aux compétences des auteurs. La plupart des systèmes auteurs actuels proposent des méthodes d'édition très liées au format de document sous-jacent. De ce fait, l'édition ne propose pas, ou peu, de fonctionnalités pour aider l'auteur et automatiser certaines tâches. De plus, la plupart de ces propositions ne prennent pas en charge la détection des incohérences créées par les auteurs lors de la spécification de leurs besoins et contraintes.

1 DEMARCHE ET BILAN DU TRAVAIL

Ce travail s'est organisé en deux grandes parties, ce découpage se retrouve dans la structuration des chapitres de la thèse.

1.1 Multimédia et environnements "auteur" de documents multimédias

La première partie est axée autour des données manipulées dans les documents multimédias et leurs utilisations dans des environnements "auteur" de documents multimédias. Nous avons montré qu'une présentation multimédia est une opération complexe qui fait intervenir des traitements très variés (synchronisation, accès réseau, etc.). Ces traitements doivent respecter un ensemble de contraintes qui sont décrites dans le document multimédia (scénario temporel) ou imposés par l'environnement d'utilisation (réseau, interaction utilisateur, etc.). Parmi elles, la restitution des objets média et leur synchronisation au sein du scénario traduisent les conditions de bonne restitution. A ces contraintes s'ajoutent celles liées aux ressources du système et des caractéristiques des réseaux (débit, latences) qui peuvent affecter de façon significative la qualité de la présentation.

Dans cette perspective, nous avons étudié les caractéristiques des systèmes de présentation multimédia en général et analysé la manière avec laquelle elles sont prises en compte au sein des systèmes existants. L'étude des standards et des systèmes multimédias existants tels que HyTime, SMIL, MHEG et PREMO, a montré l'appartenance de chacun à une approche de conception (impérative, déclarative, etc.) permettant d'obtenir une structure temporelle et la structure de contrôle qui peut en être dérivée. Nous avons mis en évidence aussi bien les points forts que les inconvénients et les faiblesses de ces systèmes. Parmi les inconvénients majeurs relevés sur de ces propositions nous comme la faible portabilité des documents édités et la difficulté de leur maintenance ainsi que leur faible durée de vie.

Un premier résultat de la cette étude consiste à la définition de ce que serait les fonctions de base qui doivent être fournies dans un environnement permettant à un auteur de spécifier à la fois simplement et de manière complète son document multimédia et de le restituer à sa demande. Le deuxième résultat présenté est le système auteur Lumière qui permet aux concepteurs de ces documents électroniques de les éditer et de les restituer.

1.2 Systèmes auteur pour les documents multimédias

Notre contribution commence par une présentation du système auteur contexte de notre travail. En particulier, nous avons abordé la représentation des documents dans Lumière à travers son modèle objet MDO2 qui décrit la représentation interne de ces documents en

termes de structures de données et d'opérations d'édition et de traitements induits sur ces structures. Le modèle offre l'avantage de la richesse de ses types de données pouvant être renforcée grâce à son extensibilité. Nous nous sommes aussi intéressés au maintien de la cohérence des synchronisations temporelles établies sur les documents. Pour cela, nous avons proposé le modèle RdP_MAL comme format interne pour la vérification et validation permettant la détection et la résolution des inconsistances si elles existent.

Nous avons ensuite abordé l'architecture de référence de cette application qui définit et organise les différents traitements d'une présentation multimédia au sein d'un ensemble de modules interagissant à travers une interface utilisateur. L'architecture de référence du système qui respecte un cycle de vie pour tout DMM, est organisée autour de trois modules : un éditeur et un exécuteur qui prennent en charge les fonctions d'édition et de présentation des objets médias hétérogènes ; et un validateur conçu dans un souci de recherche de qualité de service responsable de la gestion de validité des documents édités et d'estimation des coûts associés aux présentations. Une propriété majeure de cette architecture est l'exécution en parallèle de ses modules ce qui permet un gain de temps considérable en comparaison aux multiples fonctions exécutées pour le contentement de l'utilisateur.

2 PERSPECTIVES

Dans le nouveau monde numérique qui se dessine, le multimédia apparaît comme une plaque tournante autour de laquelle viennent se greffer un ensemble de problèmes conceptuels et techniques qui appartiennent à des mondes qui jusqu'ici n'évoluaient pas dans la même direction, comme la télévision, les télécommunications ou l'industrie du jeu. Des propositions originales continuent d'émerger des systèmes multimédias et un effort d'organisation se fait aujourd'hui sentir.

Les perspectives que nous proposons ont pour objet principal d'améliorer la qualité des environnements de présentation multimédia que ce soit sur le plan de la portabilité, des performances ou de l'adaptabilité. Ainsi, les travaux qui restent à accomplir peuvent s'orienter sur les grandes directions suivantes :

- 0 Extension des modèles de représentation des documents : Le jeu des relations de synchronisation peut encore être augmenté afin de rendre d'autres aspects des présentations multimédias. Le travail futur autour des opérateurs peut s'orienter dans deux directions en particulier:
 - o la composition spatiale où il serait intéressant d'étendre et généraliser les modèles pour mieux combiner les spécifications de nature intrinsèquement déclarative comme « l'objet A est aligné à gauche avec l'objet B pendant la durée de l'objet C », de celles qui sont de nature événementielle comme « lorsque l'objet A atteint la position (x, y), alors démarrer l'objet B » ;
 - o les annotations bien que moins primordiales, cette dimension n'en est pas moins nécessaire au sein d'une présentation multimédia, afin de permettre de définir le sens abstrait des présentations multimédias, au sens où leurs créateurs l'entendent, et permettre l'organisation du contenu multimédia dans son ensemble afin de faciliter les recherches au sein du vaste contenu qui existe aujourd'hui; il est alors possible d'imaginer des méthodes de requêtes ad hoc ou inspirées de celles créées pour les bases de données multimédias [Tran_Thuong 03].
- 0 Prototypage *rapide* de documents : Un des besoins essentiels lors de la conception de documents est la possibilité de prototyper rapidement une version de manière à pouvoir avoir un aperçu du document. L'auteur, en peu de temps, devrait obtenir une maquette qui lui permet d'avoir une idée précise de ce que sera son document dans sa forme finale (comme la fenêtre "Get a look" proposée dans Lumière). L'idéal est qu'au lieu d'une seule vue, l'auteur puisse naviguer dans un espace de solutions (plusieurs versions) dépendant de différentes contraintes. Cette phase de prototypage est très utile lors de la production de documents dans un contexte collaboratif. Ces maquettes serviront alors de point de départ pour les discussions entre les concepteurs [Falkovych 06].
- 0 Ordonnancement et présentation multimédia à base d'agents mobiles : Les agents mobiles ont émergé comme un paradigme puissant en informatique répartie et fournissent déjà un environnement plus flexible que le modèle classique du client - serveur. Dans un environnement d'agents, un programme ou agent est exécuté sur le site où réside les données ; il interagit ainsi de façon plus directe et plus proche avec les données multimédias distantes. L'étude à mener consiste à effectuer l'ordonnancement et la

synchronisation d'une présentation multimédia à travers d'un ensemble d'agents déplacés depuis l'application cliente. Ces agents permettent, en reconstituant le scénario sur les sites distants, de synchroniser les données avant leur transfert, d'anticiper les transferts et d'adapter les contraintes temporelles spécifiées dans les scénarios temporels des documents présentés. De plus, ces agents peuvent être définis conformément à la structure logique (agents structurés) du document afin de prendre en charge des parties identifiées du document multimédia. Les agents peuvent aussi être employés pour déployer des politiques de filtrage des données du côté des serveurs en plus des autres activités de sécurité [Ghinea 06, Roedig 05].

- O** Spécification de documents adaptables : cette extension à considérer vient des besoins de conception de documents qui soient adaptables (1) aux différents types de terminaux (du PC au téléphone mobile), (2) à différentes classes d'utilisateurs (du néophyte au spécialiste par exemple) et (3) au contexte réseau (débit et qualité de la transmission). Il est nécessaire d'étudier quelles informations doivent être fournies au moment de la spécification et comment les exploiter au moment de la présentation. Les traitements nécessaires à l'adaptation se situent à différentes étapes de la chaîne de production des documents : de façon statique, c'est-à-dire avant la présentation (choix d'une feuille de style par exemple), ou de façon dynamique, pendant la présentation.

En conclusion, nous sommes conscient que des efforts restent à faire afin d'arriver à de meilleurs résultats mais nul doute que la créativité qui est propre au multimédia saura ouvrir de nouvelles portes et qu'elle sera un moteur d'expansion et de maturation pour l'informatique du futur.

Remerciements

Un long labeur et de longues années d'étude, de recherche, d'hypothèses et enfin ... un résultat. Tout ceci je le doit avant tout à Dieu Le Tout Puissant et ensuite à une grande équipe qui m'a soutenue et aidé. Pour cela je voudrai remercier :

Mon directeur de thèse le Professeur Mahmoud Boufaïda pour ses directives et sa patience ;

Le Professeur Sahnoun Zaidi d'avoir accepté de présider le jury de cette thèse ;

Les membres du jury : Professeur Benmohamed Mohamed et Docteur Zidani Abdelmadjid qui m'ont accordé de leur temps précieux pour se pencher sur mon humble travail ;

Le Docteur Poitrenaud Denis qui m'a fait l'honneur de se déplacer de France ainsi que pour son aide précieuse pendant mes travaux de recherche ;

Les membres du laboratoire Lire et plus particulièrement l'équipe SIBC ainsi que l'équipe SRC du laboratoire LIP6 de l'Université Pierre & Marie Curie pour leur accueil ;

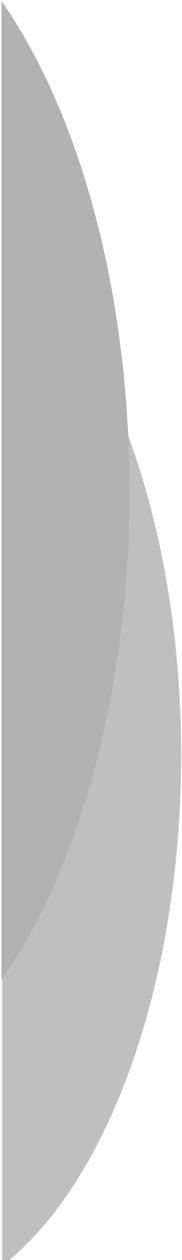
Les membres du département d'Informatique, Faculté des Sciences de l'Ingénieur de L'université Mentouri de Constantine ;

Tous mes amis, trop nombreux pour être énumérés.

A vous tous et à tous ceux qui m'ont aidé et que je n'ai pas cité

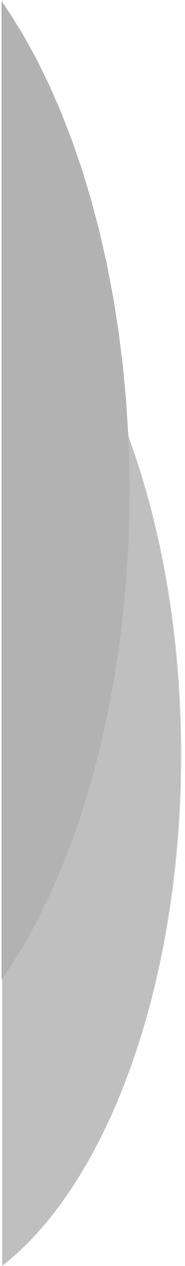
GRAND MERCI.

*A mes parents avec tout mon amour
Aux êtres chers à mon coeur
A ceux qui m'ont aimée telle que je suis.*



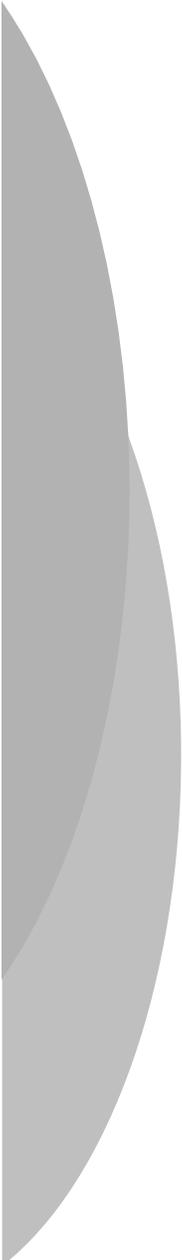
PARTIE 1 :

Les Documents multimédias



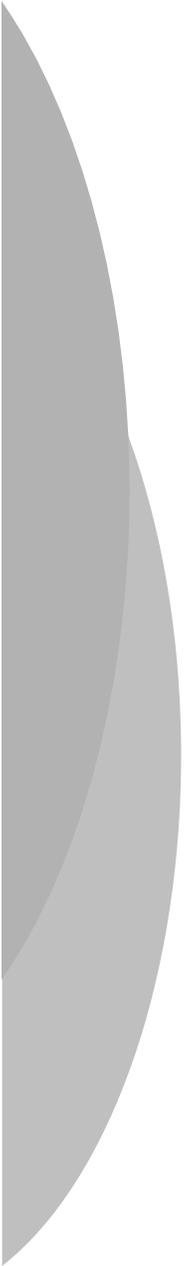
CHAPITRE 1 :

Notions De Base



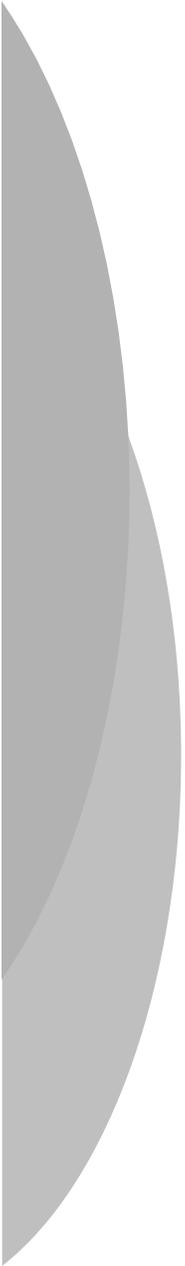
CHAPITRE 2 :

*Modélisation
De La Synchronisation
Des DMMs*



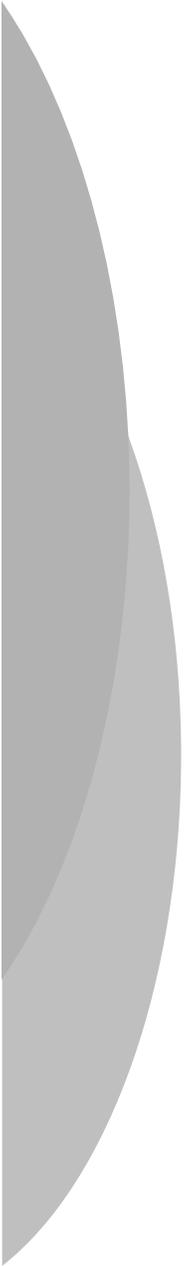
CHAPITRE 3 :

Outils Et Standards multimédias



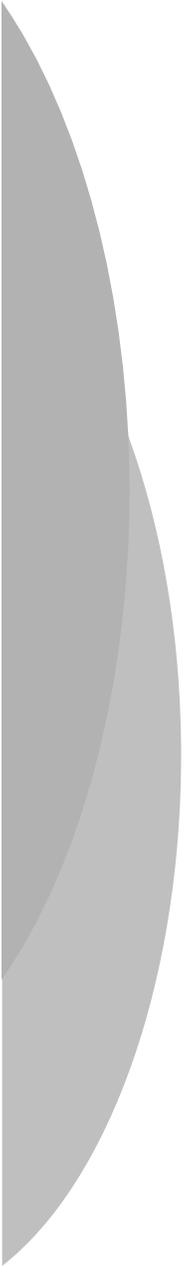
PARTIE 2 :

*Lumière, Un Système
Auteur Pour Les DMMs*



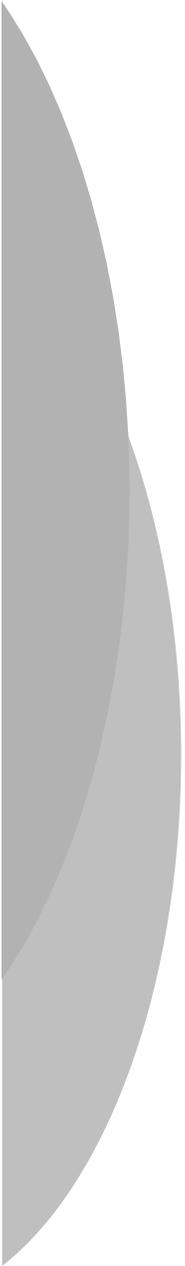
CHAPITRE 4 :

MDO2 : Un Modèle De Documents Orienté Objet



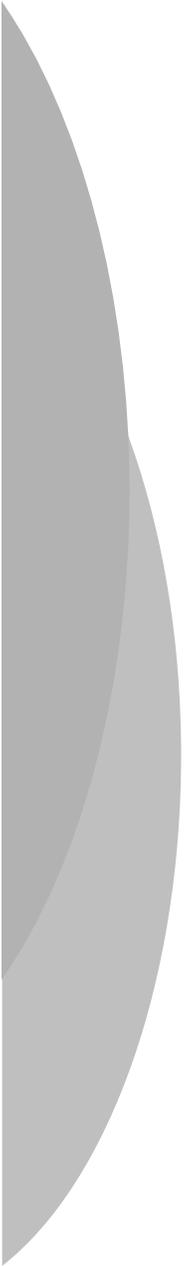
CHAPITRE 5 :

*Vérification
Des Documents multimédias*

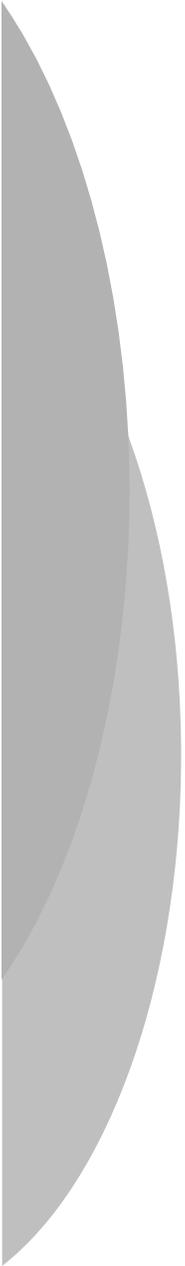


CHAPITRE 6 :

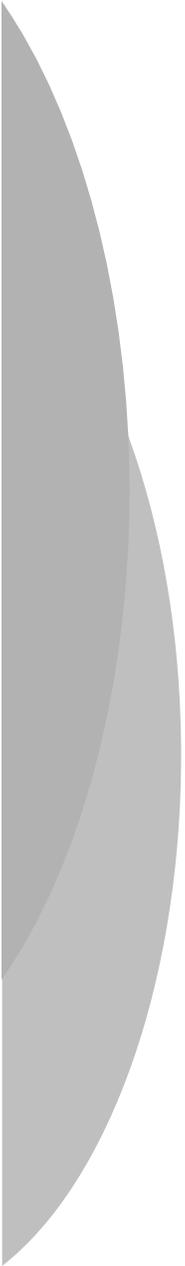
*Architecture et Interface
de Lumière*



INTRODUCTION GENERALE



CONCLUSION GENERALE



REFERENCES

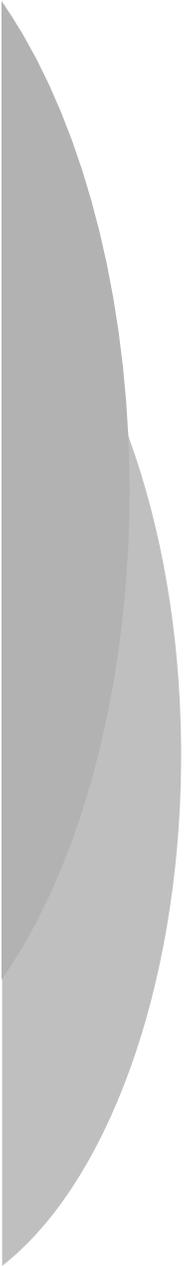


TABLE DE MATIERE

- [Adiba 96] M. Adiba. *Multimedia Database System, Design and Implementation Strategies*, Chapitre 3 : *STORM: An Object-Oriented Multimedia DBMS*, pp 47-88, Edition : Kluwer Academic Publishers, 1996.
- [Adiba 99] M. Adiba & J-L Zechinelli-Martini. *Spatio-Temporal Multimedia Presentations as Database Objects*, Proceedings : DEXA'99, 10th International Conference on Databases and Expert Systems Applications, Florence, Italie. Septembre 1999.
- [Aceto 03] L. Aceto, P. Bouyer, A. Burgueño & K. G. Larsen. *The power of reachability testing for timed automata*, Theoretical Computer Science, Vol. 1, Num. 3, pp. 411-475, 2003.
- [Allen 83] J. F. Allen. *Maintaining Knowledge about Temporal Intervals*, Communication ACM, Vol. 26, Num. 11, pp. 832-843. Novembre 1983.
- [Almeida 04] J. M. Almeida, D. L. Eager, M. K. Vernon & S. J. Wright. *Minimizing Delivery Cost in Scalable Streaming Content Distribution Systems*, IEEE Transactions on Multimedia, Vol. 6, Num. 2, pp. 356-365, Avril 2004.
- [Al-Salqan 96] Y.Y. Al-Salqan & C.K. Chang. *Temporal Relations and Synchronization Agents*, IEEE Multimedia Vol. 3, pp. 30–39, 1996.
- [ATG-FCTOOLS] <http://www.sop.inria.fr/meije/meijetools.html>.
- [Artsvideo] Système de la Société Arts Video Interactive : <http://www.artsvideo.com>
- [Ayars 01] J. Ayars, D. Bulterman, A. Cohen & al. *Synchronized Multimedia Integration Language (SMIL 2.0)*, Rapport technique: REC-smil20-20010807, World Wide Web Consortium, Août 2001: <http://www.w3.org/TR/smil20/>.
- [Bailey 01 a] B. Bailey, J. Konstan & J. Carlis. *DEMAIS: Designing Multimedia Applications with Interactive Storyboards*, Proceedings : ACM Multimedia, 2001.
- [Bailey 01 b] B. Bailey, J. Konstan, & J. Carlis. *Supporting Multimedia Designers: Towards More Effective Design Tools*, 8th International Conference Multimedia Modeling (MMM01), pp. 5-7, Amsterdam, November 2001.

- [Baldan 01] P. Baldan, A. Corradini & U. Montanari. *Contextual Petri Nets, Asymmetric Event Structures and Processes*, Information Computer, Vol. 171, Num. 1, pp. 1-49, 2001.
- [Behrmann 04] G. Behrmann. *Distributed reachability analysis in timed automata*, Software Tool for Technology Transfer (STTT), 2004.
- [Berkom 98] Berkom GLUE Project: <http://www.fokus.gmd.de/ovma/berglass/entry.html>, 1998.
- [Berthelot 94] G. Berthelot & H. Boucheneb. *Occurrence Graphs for IntervalTimed Coloured nets*, 15th International Conference on Application and Theory of Petri Nets, LNCS 815, Springer-Verlag, Zaragoza, Espagne, Juin 1994.
- [Berthomieu 91] B. Berthomieu & M. Diaz. *Modeling and Verification of Time Dependent Systems Using Time Petri Nets*, IEEE Transactions on Software Engineering, Vol. 17, Num. 3, 1991.
- [Bertino 00] E. Bertino, E. Ferrari & M. Stolf. *MPGS: An Interactive Tool for the Specification and Generation of Multimedia Presentation*, IEEE Transactions on Knowledge and Data Engineering, Vol. 12, Num. 1, pp. 102-125, 2000.
- [Bittner 02] T. Bittner. *Reasoning about Qualitative Spatio-temporal Relations at Multiple Levels of Granularity*, Proceedings : Conference ECAI'02, pp. 317-321, Lyon, France, 21-26 Juillet 2002.
- [Bitzer 96] H. W. Bitzer, C. Gran & K. Hofrichter. *MAJA: MHEG Applications Using JAVA Applets*, Proceedings: Real Time Multimedia and the World Wide Web, W3C Workshop (RTMW'96), Sophia-Antipolis, France, Octobre 1996.
- [Blair 97] G. S. Blair, L. Blair, & J. B. Stefani. *A Specification Architecture for Multimedia Systems in Open Distributed Processing*, Computer Networks and ISDN Systems, Vol. 29, Num. 4, Mars 1997.
- [Blakowski 96] G. Blakowski & R. Steinmetz. *A Media Synchronization Survey: Reference Model, Specification and Case Studie*, IEEE Journal of Selected Areas in Communications, Vol. 14, Num. 1, pp. 5-34, Janvier 1996.
- [Bollard 99] M. Bollard. *Madeus, un Environnement Auteur de Documents Multimédia*, DEA Sciences Cognitives, Université Pierre Mendès, France, Juin 1999.
- [Boucheneb 02] H. Boucheneb & G. Berthelot. *Contraction of the ITCPN State Space*, Electronic Notes in Theoretical Computer Science, Vol. 65, Num. 6, 15 pp, 2002 : <http://www.elsevier.nl/locate/entcs/volume65.html>

- [Boucheneb 06] H. Boucheneb & R. Hadjidj. *CTL* Model Checking for Time Petri Nets*, Theoretical Computer Science- Elsevier 353, pp. 208-227, 2006.
- [Bouyer 02] P. Bouyer. *Modèles et Algorithmes pour la Vérification des Systèmes Temporisés*, Thèse de Doctorat, Ecole Normale Supérieure de Cachan, Avril 2002.
- [Bouyer 06] P. Bouyer, S. Haddad & P-A. Reynier. Timed Petri Nets and Timed Automata: On the Discriminating Power of Zeno Sequences. ICALP'2006 33rd International Colloquium on Automata, Languages and Programming S. Servolo, Venise, Italie, 9 – 16 Juillet 2006.
- [Brams 83] G. W. Brams. Réseaux de Petri : théorie et pratique, volume 1 (théorie et analyse) et 2 (modélisation et applications), Edition : Masson- Paris, Mars 1983.
- [Buchanan 93] M. C. Buchanan & P. T. Zellweger. *Automatic Temporal Layout Mechanisms*, Proceedings : the First ACM International Conference on Multimedia, pp. 341-350, ACM Press, Anaheim, Californie, Août 1993.
- [Buchanan 05] M. C. Buchanan & P. T. Zellweger. *Automatic Temporal Layout Mechanisms Revisited*, ACM Transactions on Multimedia Computing Communications and Applications, Vol. 1, Num. 1, pp. 61-88, 2005.
- [Buford 94] J. F. Koegel Buford. *Multimedia Systems*, Edition : ACM Press, 1994.
- [Bulterman 91] D. C-A. Bulterman & R. Van-Liere. *Multimedia Synchronization and UNIX*. Proceedings : the 2nd International Workshop on Network and OS Support for Digital Audio/Video, LNCS 164, Heidelberg, Germany, Novembre 1991.
- [Bulterman 95] D. C-A. Bulterman. *Embedding Video in Hypermedia Documents: Supporting Integration and Adaptive Control*, ACM Transactions on Information Systems, pp. 1-30, Octobre 1995.
- [Bulterman 98] D. C-A. Bulterman, L. Hardman, J. Jansen, K. Mullender & L. Rutledge. *GRiNS: A GRAPHICAL INTERface for Creating and Playing SMIL Documents*, Proceedings : the 7th International World Wide Web Conference, pp. 519-529, Brisbane, Australie, Avril 1998.
- [Bulterman 02] D. C-A. Bulterman. *SMIL 2.0*, IEEE Multimedia, pp. 74–84, Edition : Peiya Liu, Janvier- Mars 2002.
- [Bulterman 05] D. C-A. BULTERMAN & L. HARDMAN. *Structured Multimedia Authoring*, ACM Transactions on Multimedia Computing Communications and Applications, Vol. 1, Num. 1, pp. 89-109, Février 2005.
- [Busi 99] N. Busi & G.M. Pinna. *Process Semantics for Place/Transition Nets with Inhibitor and Read Arcs*, Fundamenta Informaticae, Vol. 40, Num. 2-3, pp. 165-197, 1999.

- [CADP] CADP- CAESAR/ ALDEBARAN Development Package.
- [Campos 99] S. Campos, B. Ribeiro-Neto, A. Macedo & L. Bertini. *Formal Verification and Analysis of Multimedia Systems*, Proceedings : the Seventh ACM Multimedia Conference (MM'99), Orlando, Florida, 30 Octobre au 5 Novembre 1999.
- [Candan 96] K. S. Candan, B. Prabhakaran & V. S. Subrahmanian. *CHIMP : A Framework for Supporting Distributed Multimedia Document Authoring and Presentation* , Proceedings : the ACM Multimedia, pp. 329-339, Boston, USA, Novembre 1996.
- [Carcone 97] L. Carcone. *Formatage Spatial dans un Environnement d'Edition/Présentation de Documents Multimédia*, Mémoire CNAM-Grenoble, Décembre 1997.
- [Cardelli 00] L. Cardelli & A. D. Gordon. *Anytime, Anywhere. Modal Logics for Mobile Ambients*, Proceedings : the 27th ACM Symposium on Principles of Programming Languages, pp. 365-377, Boston, MA, USA, 19-21 Janvier 2000.
- [Celentano 02] A. Celentano & O. Gaggi. *Schema modeling for automatic generation of multimedia presentations*, SEKE'02, pp. 593-600, Italie, 2002.
- [Chaochen 91] Z. Chaochen, C.A.R. Hoare & A.P. Ravn. *A Calculus of Duration*, Information Processing Letters, Vol. 40, Num. 5, pp. 269-276, 1991.
- [Chaoui 00] J. Chaoui, K. Cyr, J-P. Giacalone & al. *OMAPTM: Enabling Multimedia Applications in Third Generation (3G) Wireless Terminals*, Rapport technique: SWPA001, Texas Instrument, Décembre 2000.
- [Courtiat 94] J-P. Courtiat & R. Cruz de Oliveira. *About Time NonDeterminism and Exception Handling in a Temporal Extension of LOTOS*, Proceedings : the 14th International IFIP Symposium on Protocol Specification, Testing and Verification (PSTV'94), pp. 33-49, Vancouver, Canada, 7-10 Juin 1994.
- [Courtiat 96a] J-P. Courtiat, L.F.R. Carmo & R.C. De Oliveira. *A General-Purpose Multimedia Synchronization Mechanism Based on Causal Relations*, IEEE Journal on Selected Areas in Communications, Vol. 14, Num. 1, pp. 185-195, Janvier 1996.
- [Courtiat 96b] J-P. Courtiat & R. C. de Oliveira. *Proving Temporal Consistency in a New Multimedia Synchronization Model*, Proceedings : the ACM International Conference on Multimedia, Boston, MA, USA, 18-22 Novembre 1996.
- [Courtiat 00] J-P. Courtiat, C.A.S. Santos, C. Lohr & O. Benaceu. *Experience with RT-LOTOS, a Temporal Extension of the LOTOS Formal Description Technique* , Computer Communications, Vol. 23, Num. 12, pp.1104-1123, Juillet 2000.

- [CWI 98] CWI, Centrum Voor Wiskunde en Informatica : <http://www.cwi.nl/>, 1998.
- [DEC 98] DEC (Digital Equipment Corporation) : <http://www.digital.com/>, 1998.
- [Dechter 91] R. Dechter, I. Meiri & J. Pearl. *Temporal Constraint Networks*, Artif Intelligence, Vol. 49, Num. 61-95, 1991.
- [Derose 94] S. J. Derose. & D. G. Durand. *Making Hypermedia Work: A User's Guide to HyTime*, Kluwer Academic Publishers, Mars 1994.
- [Diaz 94] M. Diaz & P. Sénac. *Time Stream Petri Nets, a Model for Timed Multimedia Information*, Proceedings : the 15th International Conference on Application and Theory of Petri Nets, LNCS 815, pp. 219-238, Zaragoza, Espagne, Juin 1994.
- [Diaz 98] M. Diaz, P. De Saqui Sannes, P. Senac & R. Willrich. *Modélisation et Atelier Logiciel pour Applications Multimédias*, Rapport LAAS : 98097, pp. 99, Mars 1998.
- [Diaz 01] M. Diaz (Direction) & al. *Les Réseaux de Petri : Modèles Fondamentaux*, Edition : Hermes Science Europe, 2001.
- [Dimitrova 99] N. Dimitrova, R. Koenen, H. Yu, A. Zakhor, F. Galliano & C. Bouman. *Video Portals for the Next Century*, Proceedings : the Seventh ACM Multimedia Conference (MM'99), Electronic Proceedings, Orlando, Florida, 30 Octobre au 5 Novembre 1999.
- [Donikian 93] S. Donikian & G. Hégron. *A Declarative Desing Method for 3D Scene Sketch Modeling*, Proceedings : the Eurographics'93 Conference, Vol. 12, Num. 3, Barcelona, Espagne, Septembre 1993.
- [DOTTY] <http://www.research.att.com/tools/graphviz/>
- [Drapeau 93] G. D. Drapeau. *Synchronization in the MAestro Multimedia Authoring Environment*, Proceedings : the First ACM Conference on Multimedia, ACM Press, pp.331-340, Anaheim, Californie, Août 1993.
- [Duke 97] D. J. Duke, D. A. Duce, I. Herman & G. Faconti. *Specifying the PREMO Synchronization Objects*, Rapport Technique : ERCIM-02/97-R048, ISO/IEC, Février 1997.
- [Dutertre 95] B. Dutertre. *On First Order Interval Temporal Logic*, Rapport Technique : CSD-TR-94-3, Royal Holloway, Université de Londres, 7 Février 1995.
- [Edwards 02] J. Edwards & L. Dailey Paulson. *Smart Graphics: A New Approach to Meeting User Needs*, Computer, Vol. 35, Num. 5, pp. 18-21, Mai 2002.

- [Erfle 93] R. Erfle. *Specification of Temporal Constraints in Multimedia Documents Using HyTime*, Publication Electronique, Vol. 6, Num. 4, pp. 397-411, Décembre 1993.
- [Euzenat 95] J. Euzenat. *An Algebraic Approach for Granularity in Qualitative Space and Time Relations*, Proceedings : the Conference IJCAI-95, pp. 894-900, 1995.
- [Falkovych 06] K. Falkovych & F. Nack. *Context Aware Guidance for Multimedia Authoring: Harmonizing Domain and Discourse Knowledge*, Multimedia Systems Springer, Vol. 11, Num. 3, pp. 226-235, Mars 2006.
- [Flanck 02] S. Flanck. *Multimedia Technology in Context*, IEEE Multimedia, Juillet - Septembre 2002.
- [François 97] P. François. *Etude et Conception d'une Structure d'Accueil de Documents Hypertextés Structurés Selon les Normes SGML et HyTime : Application à la Documentation Aéronautique*, Thèse de Doctorat, Université Paul Sabatier - Toulouse III (Sciences), Octobre 1997.
- [Gaffiot 02] F. Gaffiot. *Le Grand Gaffio, Dictionnaire Français-Latin*. Hachette, 2000.
- [Gerevini 02] A. Gerevini & N. Bernhard. *Qualitative Spatio-Temporal Reasoning with RCC-8 and Allen's Interval Calculus: Computational Complexity*, Proceedings : the Conference ECAI'02, pp. 312-316, Lyon, France, 21-26 Juillet 2002.
- [Geyer 97] L. Geyer, M. Baentsch, L. Baum, G. Molter, S. Rothkugel & P. Sturm. *MHEG in Java- Integrating a Multimedia Standard into the Web*, Session Poster : the 6th International World Wide Web Conference, California, USA, Avril 1997.
- [Ghafoor 95] A. Ghafoor & Y. F. Day. *Object-Oriented Modeling and Querying of Multimedia Data*, Proceedings : First International. Workshop on Multimedia Information Systems, Arlington, VA, pp. 111-119, Septembre 1995.
- [Ghinea 06] G. Ghinea & S. Y. Chen. *Perceived Quality of Multimedia Educational Content: A Cognitive Style Approach*, Multimedia Systems Springer, Vol. 11, Num. 3, pp. 271-279, Mars 2006.
- [Golaup 06] A. Golaup & H. Aghvami. *A Multimedia Traffic Modeling Framework for Simulation-Based Performance Evaluation Studies*, Elsevier Computer Networks Num. 50, pp. 2071-2087, 2006.
- [Goldfarb 90] C. Goldfarb. *The SGML Handbook*, Edition : Oxford University Press, Oxford, 1990.

- [Grigoras 01] R. Grigoras, V. Charvillat & M. Douze. *Contenu Multimédia à Mémoire*, Proceedings : 7emes Journées d'Etudes et d'Echanges Compression et REprésentation des Signaux Audiovisuels (CORESA), Dijon, France, Novembre 2001.
- [Guan 98] S. U. Guan, H. Y. Yu & J. S. Yang. *A Prioritized Petri net Model and its Application in Distributed Multimedia Systems*, IEEE Transactions on Computers, Vol. 47, Num. 4, pp. 477-481, 1998.
- [Guan 00] S. U. Guan & S. S. Limm. *Modeling Interactive Distributed Multimedia Application by a New Class of Reconfigurable Petri Nets*, Proceedings: ISA'2000, International ICSC Congress on Intelligent Systems and Applications, University of Wollongong, Autriche, 12- 15 Décembre 2000.
- [Guan 02] S. U. Guan & S. S. Limm. *Modeling with Enhanced Prioritized Petri Nets: EP-nets*, Elsevier Science Computer Communication, Vol. 25, pp. 812-824, 2002.
- [Halasz 90] F. Halasz & M. Schwarz. *The Dexter Hypertext Reference Model*, NIST hypertext Standardization Workshop, Gaithersburg, MD, 16-18 Janvier 1990.
- [Hamblin 72] C. L. Hamblin. *Instants and Intervals*. Proceedings : the 1st Conference of the International Society for the Study of Time, pp. 324-331, New York, 1972.
- [Hanappe 99] P. Hanappe. *Design and Implementation of an Integrated Environment for Music Composition and Synthesis*, Thèse de Doctorat, Université Paris 6, Avril 1999.
- [Hardman 93a] L. Hardman, D. C-A. Bulterman & G. Van Rossum. *The Amsterdam Hypermedia Model: Extending Hypertext to Support Real Multimedia*, Hypermedia Journal, Vol. 5, Num. 1, pp. 47-69, Juillet 1993.
- [Hardman 93b] L. Hardman, G. Van Rossum & D. C-A. Bulterman. *Structured Multimedia Authoring*, Proceedings : the ACM International Multimedia Conference, pp. 283-289, Anaheim, CA, USA, 1993.
- [Hardman 93c] L. Hardman, D. C-A. Bulterman & G. Rossum. *Links in Hypermedia: the Requirement for Context*, Hypertext '93, pp. 183-191, Novembre 1993.
- [Henzinger 98] T. A. Henzinger, J-F. Raskin & P-Y. Schobbens. *The Regular Real-Time Languages*, Automata, Languages and Programming, 25th International Colloquium, ICALP'98, volume 1443 of Lecture Notes in Computer Science, Aalborg, Denmark, 13-17 Juillet 1998.
- [Herman 94] I. Herman, C.S. Carson, J. Davy & al. *PREMO: An ISO Standard for a Presentation Environment for Multimedia Objects*, Proceedings : the ACM International Conference on Multimedia, San Francisco, CA, 15-20 Octobre 1994.

- [Hirzalla 95] N. Hirzalla, B. Falchuk & A. Karmouch. *A Temporal Model for Interactive Multimedia Scenarios*, IEEE Multimedia, Vol. 2, Num. 3, Fall 1995.
- [Huang 99] C. M. Huang, C. H. Lin & C. Wang. *Specifying and Executing Interactive Multimedia Presentations using the Formal Approach*, Proceedings : National Science Council, Vol. 23, Num. 4, pp. 495-510, 1999.
- [HyTime 97] HyTime User's Group, *A Reader's Guide to the HyTime Standard* : <http://www.hytime.org/papers/htguide.html>, 1997.
- [ISO 96a] ISO/IEC JTC1/SC24, *Presentation Environments for Multimedia Objects (PREMO); Part 1: Fundamentals of PREMO*, Num. 14478-1:199x(E), ISO/IEC, Septembre 1996.
- [ISO 96b] ISO/IEC JTC1/SC24, *Presentation Environments for Multimedia Objects (PREMO); Part 2: Foundation Component*, Num. 14478-2:199x(E), ISO/IEC, Septembre 1996.
- [ISO 96c] ISO/IEC JTC1/SC24, *Presentation Environments for Multimedia Objects (PREMO); Part 3: Multimedia Systems Services*, Num. 14478-3:199x(E), ISO/IEC, Septembre 1996.
- [ISO 96d] ISO/IEC JTC1/SC24, *Presentation Environments for Multimedia Objects (PREMO); Part 4: Modelling, Rendering, and Interaction Component*, Num. 14478-4:199x(E), ISO/IEC, Septembre 1996.
- [ISO 97] ISO/IEC JTC1/SC18/WG8 N1920, *Information Technology: Hypermedia Time-based Structuring Language (HyTime)*, deuxième édition, ISO/IEC, Août 1997 : <http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html>.
- [Janicki 95] R. Janicki, M. Koutny. *Semantics of Inhibitor Nets*, Information Computer, Vol. 123, Num. 1, pp. 1-16, 1995.
- [Jin 04] J. Jin & K. Nahrstedt. *QoS Specification Languages for Distributed Multimedia Applications: A Survey and Taxonomy*, IEEE MultiMedia, Feature Article, IEEE Computer Society, Juillet–September 2004.
- [Jourdan 98] M. Jourdan, N. Layaïda, L. Sabry-Ismaïl, C. Roisin & L. Tardif. *MADEUS: an Authoring Environment for Interactive Multimedia Documents*, Proceedings : the 6th ACM International Multimedia Conference, pp. 267-272, Bristol, Angleterre, 12-16 Septembre 1998.
- [Keramane 96] C. Keramane & A. Duda. *Interval Expressions – a Functional Model for Interactive Dynamic Multimedia Presentations*, Proceedings : the IEEE International

Conference on Multimedia Computing and Systems, pp. 283–286, Hiroshima, Japan, Juin 1996.

[Kim 95] M. Y. Kim. *Multimedia Documents With Elastic Time Proc*, 3^{ème} Conférence sur le Multimédia, pp. 143-154, San Francisco, Novembre 95.

[Koenen 01] R. Koenen. *Overview of the MPEG-4 Standard*, ISO/IEC JTC1/SC29/WG 11 N4030, Mars 2001.

[Kumar 99] G. K. Kumar, J. S. Lipscomb, A. Ramchandra & al. *The HotMedia Architecture: Progressive and Interactive Rich Media for the Internet*, Rapport technique : RC21519(97069) 6JUL1999, IBM Internet Media Group, Juillet 1999.

[Kwon 99] Y. M. Kwon, E. Ferrari & E. Bertino. *Modeling Spatial-Temporal Constraints for Multimedia Objects*, Data Knowledge Engineering, Vol. 30, pp. 217-238, 1999.

[Labeled 99] I. Labeled I & P. Barril. Towards a model supporting synchronization of multimedia data and cost estimation, ISPS'99 Alger, pp. 42- 51, 18- 20 Octobre 1999.

[Labeled 04a] I. Labeled & M. Boufaïda. *A temporal Synchronization Model for Cost Estimation in Distributed Multimedia Documents*, Proceedings : MediaNet'04, pp. 255-260, Tozeur-Tunis, 25- 28 Novembre 2004.

[Labeled 04b] I. Labeled & M. Boufaïda. *O2DM: An Object Oriented Model for Multimedia Authoring Systems*, Proceedings: ACIT'04, pp. 667, Constantine- Algérie, 12- 15 Décembre 2004.

[Labeled 05] I. Labeled & M. Boufaïda. *Un Modèle de Documents Multimédias Distribués*. Proceedings : CHIA'05, pp: 167- 173, Bordj Bouarreridj, Algérie, 19- 21 Novembre 2005.

[Labeled 07] I. Labeled & M. Boufaïda. *A Specification and Generation System for Interactive Multimedia Documents*. Information Technology Journal, Vol. 6, Num. 7, pp. 1013- 1020, 2007.

[Laird 02] J. E. Laird. *Research in Human-Level AI Using Computer Games*. Communications ACM, Vol. 45, Num. 1, pp. 32-35, Janvier 2002.

[Layaïda 97] N. Layaïda. *Madeus : Système d'Édition et de Présentation de Documents Structurés Multimédia*. Thèse de Doctorat, Université Joseph Fourier – Grenoble 1. Juin 1997.

[Layaïda 98] N. Layaïda. *Synchronized Multimedia Integration Language (SMIL) :* <http://www.inrialpes.fr/opera/people/Nabil.Layaïda/smil/smil.html> 1998.

- [Lecomte 00] D. Lecomte, D. Cohen, P. De Bellefonds & J. Barda. *Les Normes et les Standards du Multimédia*, Edition : Dunod (2eme), 2000.
- [Li 96] J. Z. Li, M. T. Ozsú & D. Szafron. *Modeling Video Spatial Relationships in an Object Model*, Rapport technique : TR 96-06, Department of Computing Science, University of Alberta, Mars 1996.
- [LimSee] LimSee : *Un Editeur Temporel pour les Documents au Format SMIL*, Projet Opéra/Wam : <http://opera.inrialpes.fr/LimSee.html>.
- [Little 90] T. D. C. Little & A. Ghaffor. *Synchronization and Storage Models for Multimedia Objects*. IEEE Journal on Selected Areas in Communications, Vol. 8, Num. 3, Avril 1990.
- [Little 93] T. D. C. Little & A. Ghaffor. *Interval-Based Conceptual Models for Time-Dependent Multimedia Data*, IEEE Transactions on Knowledge and Data Engineering (Special Issue : Multimedia Information Systems), Vol. 5, Num. 4, pp. 551-563, Août 1993.
- [Lo Presti 02] S. Lo Presti. *Langage de Spécification et de Description de Présentations Multimédias*. Thèse de Doctorat, Institut National Polytechnique de Grenoble, 27 Novembre 2002.
- [Lu 96] G. Lu. *Communication and Computing for Distributed Multimedia Systems*, Edition : Artech House Publishers, 1996.
- [Ma 04] H. Ma & K. G. Shin. *Checking Consistency in Multimedia Synchronization Constraints*. IEEE Transactions On Multimedia, Vol. 6, Num. 4, pp. 565-574, Août 2004.
- [Macromedia] *Macromedia Director*, <http://www.macromedia.com/>.
- [Macromedia 01] *Macromedia Flash file format (SWF) Software Development Kit (SDK)*, 2001 : <http://www.macromedia.com/software/flash/open/licensing/fileformat/>.
- [Manjunath 03] B. S. Manjunath, P. Salembier & T. Sikora. *Introduction to MPEG-7 Multimedia Content Description Interface*, Edition : John Wiley & Sons LTD, Mars 2003.
- [Martin 97] H. Martin, M. Adiba. R. Lozano & F. Mocellin. *Management of Multimedia Data Using an Object-Oriented Database System*, Proceedings : the DEXA'97 workshop, Toulouse, France, 2-5 Septembre 1997.
- [Maubourguet 91] P. Maubourguet. *Grand Larousse Universel*, Edition : Larousse, 1991.
- [Merlin 74] P. Merlin. *A study of the Recoverability of Communication Protocols*, Thèse de Doctorat, 1974.

- [Mirble 00] I. Mirbell, B. Pernici, T. Sellis & al. *Checking the Temporal Integrity of Interactive Multimedia Documents*, VLDB Journal, Springer Verlag, Num 9, pp. 111-130, 2000.
- [Mocellin 97] F. Mocellin. *Gestion de Données et de Présentations Multimédias par un SGBD à Objets*, Thèse de Doctorat, Université Joseph Fourier, Grenoble, Décembre 1997.
- [Montanari 95] U. Montanari & F. Rossi. *Contextual nets*. Acta Inf., 32(6):545–596, 1995.
- [Motif 92] Motif. *Programmer's Guide*, Open Software Foundation, Adresse: 11 Cambridge Center, Cambridge, MA 02142, 1992.
- [Mourlas 02] C. Mourlas. *Specification and Verification of Quality Requirements in Distributed Multimedia Presentations*, Proceedings : the 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02), IEEE, 2002.
- [Muchalut-Saade 01] D. C. Muchalut-Saade & L-F. G. Soares. *Hypermedia Spatio-Temporal Synchronization Relations also Deserve First-Class Status*, Proceedings : the 8th International Conference on Multimedia Modeling (MMM'2001), CWI, Amsterdam, Pays-Bas, 5-7 Novembre 2001.
- [Muller 98] P. Muller. *A Qualitative Theory of Motion Based on Spatio-Temporal Primitives*, Proceedings : the Principles of Knowledge Representation and Reasoning Conference (KR'98), pp. 179-187, San Francisco, CA, USA, Juin 1998.
- [Ohba 98] K. Ohba, T. Tsukada, T. Kotoku & K. Tanie. *Facial Expression Space for Smooth Tele-Communications*. Proceedings : the Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98), pages 378–383, Nara, Japan, 14-16, Avril 1998.
- [Owen 98] C. B. Owen. *Multiple Media Correlation: Theory and Applications*. Rapport Technique : PCS-TR98-335, Dartmouth College, Computer Science, Hanover, NH, Juin 1998.
- [Pazandak 95a] P. Pazandak & J. Srivastava. *Experience with the DAMSEL Multimedia Specification Language*, Proceedings : the Society for Photonics and Electronic (SPIE) East, Conference Multimedia, Philadelphia, PA, USA, Septembre 1995.
- [Pazandak 95b] P. Pazandak, J. Srivastava & J. V. Carlis. *The Temporal Component of DAMSEL*, Proceedings : the Second Workshop on Protocols for Multimedia Systems (PROMS'95), Salzburg, Autriche, 1995.
- [Pérez-Luque 96] M. Pérez-Luque & T. D. C. Little. *A Temporal Reference Framework for Multimedia Synchronization*, IEEE Journal on Selected Areas in Communications (Special Issue: Synchronization Issues in Multimedia Communication), Vol. 14, Num.1, pp. 26-51, Janvier 1996.

- [Persidsky 97] A. Persidsky. *Director 5 Windows*, Edition : Eyrolles, 1997.
- [Prabhakaran 94] B. Prabhakaran & S. V. Raghavan. *Synchronization Models for Multimedia Presentation With User Participation*, ACM / Springer-Verlag Journal of Multimedia System Vol. 2, Num. 2, pp. 53-62, 1994.
- [Quentin-Baxter 98] M. Quentin-Baxter. *Hypermedia Learning Environments Limit Access to Information*, Proceedings : the Seventh International World Wide Web Conference (WWW7), Computer Networks and ISDN Systems, Brisbane, Australie, 1998.
- [Ramchandani 74] C. Ramchandani. *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, Project MAC: TR 120, MIT, 1974.
- [Real 98a] Real G2. *RealNetworks Announcing RealSystem G2* : <http://www.realaudio.com/g2/index.html>, 1998.
- [Real 98b] RealNetworks. *Formerly Progressive Networks*, 1998 : <http://www.real.com/>.
- [Rivière 02] N. Rivière, B. Bradin-Chezaviel & R. Valette. *Propagation de Contraintes et Ordonnancement de Documents Multimédias*, 5^{ème} Journée Doctorale Informatique et Réseaux (JDIR'2002), pp. 211-218, Toulouse, France, 4-6 Mars 2002.
- [Roedig 05] U. Roedig & J. Schmitt. *Multimedia and Firewalls: a Performance Perspective*, Multimedia Systems Springer, Vol. 11, Num. 1, pp. 19-33, November 2005.
- [Rogge 04] B. Rogge, J. Bekaert & R. Van de Walle. *Timing Issues in Multimedia Formats: Review of the Principles and Comparison of Existing Formats*, IEEE Transactions On Multimedia, Vol. 6, Num. 6, Décembre 2004.
- [Roisin 99] C. Roisin. *Documents Structurés Multimédia*, Habilitation à diriger les recherches, Institut national polytechnique de Grenoble, le 22 Septembre 1999.
- [Roşu 06] G. Roşu & S. Bensalem. *Allen Linear (Interval) Temporal Logic – Translation to LTL and MonitorSynthesis*, CAV 2006, LNCS 4144, pp. 263-277, 2006.
- [Rousseau 98] F. Rousseau & A. Duda. *An Execution Architecture for Synchronized Multimedia Presentations*, Proceedings : the Third European Conference on Multimedia Applications, Services and Techniques (ECMAST'98), pp. 42-55, Berlin, Allemagne, 1998.
- [Rousseau 99] F. Rousseau. *Présentations Multimédia Synchronisées pour le WWW*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, Janvier 1999.
- [Sabry-Ismaïl 98] L. Sabry-Ismaïl & R. Guetari. *L'Objet : Les Représentations par Objets en Conception*, Chapitre 2 : *Le modèle objet Madeus*, Edition : Hermes, 1998.

- [Sabry-Ismaïl 99] L. Sabry-Ismaïl. *Schéma d'Exécution pour les Documents Multimédia Distribués*, Thèse de Doctorat, Université Joseph Fourier, Janvier 1999.
- [Sampaio 00a] P. N. M. Sampaio, C. A. S. Santos & J-P. Courtiat. *About the Semantic Verification of SMIL Documents*, Proceedings : the IEEE International Conference on Multimedia (ICME 2000), pp. 1675-1678, NY, USA, 30 Juillet-2 Août 2000.
- [Sampaio 00b] P. N. M. Sampaio, C. A. S. Santos & J-P. Courtiat. *Using a Formal Method to Verify the Temporal Semantics of SMIL Documents*, Proceedings : the Conference on Software: Theory and Practice, IFIP World Congress, pp. 451-458, Beijing, Chine, 21-25 Août 2000.
- [Sampaio 03] P.N.M. Sampaio. *Conception Formelle de Documents Multimédias Interactifs: Une approche s'Appuyant sur RT-LOTOS*, Thèse de Dctorat, Université Paul Sabatier, Toulouse, France, 2003.
- [Santoso 93] H. Santoso, L. Dairaine, S. Fdida & E. Horlait. *Preserving Temporal Signature: a Way to Convey Time Constrained Flows*, Proceedings : the IEEE GLOBECOM Conference (GLOBal teleCOMmunications), Houston, USA, Novembre 1993.
- [Schnepf 96] J. Schnepf, J. A. Konstan, D. Doing. *FLIPS : Flexible Interactive Presentation Synchronization*, Distributed Multimedia Center, Department of Computer Sciences, University of Minnesota, 1996: <ftp://ftp.cs.umn.edu/users/du/papers/flips.ps>.
- [Schobbens 02] P-Y. Schobbens, J-F. Raskin & T. A. Henzinger. *Axioms for Real-Time Logics*, Theoretical Computer Science, Vol. 274, Num. 1-2, pp. 151–182, 2002.
- [Sifakis 77] J. Sifakis. *Use of Petri Nets for Performance Evaluation, Measuring, Modeling and Evaluating*, Computer Systems, Hollande, 1977.
- [SMDL 98] SMDL. *Standard Music Description Language*, 1998 : <http://www.sub.su.se/henrik/tomasf/smdl.htm>.
- [SMIL 01] SMIL: *Synchronized Multimedia Integration Language 2.0*, 2001: <http://www.w3.org/Tr/smil20>.
- [Steinmetz 90] R. Steinmetz. *Synchronisation Properties in Multimedia Systems*, IEEE Journal on Selected Areas in Communication, Vol. 8, Num. 3, pp. 401-412, 1990.
- [Steinmetz 95] R. Steinmetz & K. Nahrstedt. *Multimedia: Computing, Communications & Applications*, Edition : Prentice Hall, 1995.
- [Steinmetz 96] R. Steinmetz. *Human Perception of Jitter and Media Synchronisation*, IEEE Journal on Selected Areas in Communications, Vol. 14, Num. 1, pp. 61-72, Janvier 1996.

- [Subrahmanian 97] V. S. Subrahmanian. *Creating Distributed Multimedia Presentations. Principles of Multimedia Database Systems*. Edition : Morgan Kaufman, 1997.
- [Tardif 00] L. Tardif. *Kaomi : Réalisation d'une Boîte à Outils pour la Construction d'Environnements d'Édition de Documents Multimédias*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, Décembre 2000.
- [Tran_Thuong 01] T. Tran_Thuong & C. Roisin. *Structured Media for Multimedia Document Authoring*, International Workshop on Web Document Analysis (WDA'2001), Seattle, Washington, USA, 8 Septembre 2001.
- [Tran_Thuong 03] T. Tran_Thuong. *Modélisation et Traitement du Contenu des Médias pour l'Édition et la Présentation de Documents Multimédias*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, le 03/02/2003.
- [Tooley 95] M. Tooley. *Causation and Temporal Relations*, The Newsletter of the Philosophy of Science Society, Japan, Avril 1995 : <http://wwwsoc.nacsis.ac.jp/pssj/NL/99425data/MICHAELTOOLEY.html>.
- [UPPAAL] <http://www.uppaal.com>
- [Vazirgiannis 98] M. Vazirgiannis 98, M. Trafalis, Y. Stamati & M. Hatzopoulos. *Interactive Multimedia Scenario: Modeling and Rendering*. Proceedings : the Multimedia Track of ACM-SAC'98 Conference : <http://www.dbnet.ece.ntua.gr/~michalis/publications.html>.
- [Vieira 97] M.T.P. Vieira & M.T.P. Santos. *Content-Based Search on a MHEG-5 Standard-based Multimedia Database*, Proceedings : the QPMIDS DEXA 97 Workshop, pp. 341-350, Toulouse, France, Septembre 1997.
- [Vilain 86] M. Vilain & H. Kautz. *Constraint Propagation Algorithms for Temporal Reasoning*, Proceedings : the Fifth National Conference on Artificial Intelligence, pp. 377-382, Philadelphia, PA, USA, 1986.
- [Villard 02] L. Villard. *Modèles de Documents pour l'Édition et l'Adaptation de Présentations Multimédias*, Thèse de Doctorat, Institut National Polytechnique de Grenoble, Décembre 2002.
- [Vogler 02] W. Vogler. *Partial Order Semantics and Read Arcs*. Theoretical Computer Sciences Vol. 286, Num. 1, pp. 33-63, 2002.
- [Vuong 95] S. Vuong, K. Cooper & M. Ito. *Petri Net Models for Describing Multimedia Synchronization Requirements*, IEEE Transactions, pp. 260- 267, 1995.
- [W3C 97] W3C. *Extensible Markup Language (XML), 1997* : <http://www.w3.org/XML>.

[W3C 98a] W3C. *XML Linking Language (XLink)*, 1998 : <http://www.w3.org/TR/1998/WD-xlink-19980303>.

[W3C 98b] W3C. *Working Draft Specification of SMIL (Synchronized Multimedia Integration Language)*, Avril 1998 : <http://www.w3.org/TR/>.

[Wahl 94] T. Wahl & K. Rothermel. *Representing Time in Multimedia Systems*, Proceedings : the IEEE International Conference on Multimedia Computing and Systems, pp. 538-543, Boston, MA, USA, Mai 1994.

[Willrich 01] R. Willrich, P. De Saqui Sannes, P. Senac & M. Diaz. *HTSPN: An Experiment in Formal Modeling of Multimedia Applications Coded in MHEG or Java*, pp. 380-411. Idea Group Publishing, 2001.

[Willrich 02] R. Willrich, P. De Saqui Sannes, P. Senac & M. Diaz. *Multimedia Authoring with Hierarchical Timed Stream Petri Nets and Java*. Multimedia Tools and Applications, Vol. 16, Num. 1, pp. 7-27, Janvier – Février 2002.

[Wirag 97] S. Wirag. *Modeling of Adaptable Multimedia Documents, Intercatives Distributed Multimedia Systems and Telecommunication Services*, 4^{em} Workshop International IDMS'97, pp. 420- 429, Darmstadt, Septembre 1997.

[Wolter 00] F. Wolter & M. Zakharyashev. *Spatio-Temporal Representation and Reasoning Based on RCC-8*, Proceedings : the Seventh International Conference Principles of Knowledge Representation and Reasoning (KR 2000), pp. 3-14, Breckenridge, Colorado, USA, 11-15 Avril 2000.

[Yoon 98] K. Yoon & P. B. Berra. *TOCPN: Interactive Temporal Model for Interactive Multimedia Documents*, International Workshop on Multimedia Database Management Systems (IW-MMDBMS'98), Dayton, Ohio, USA, 1998.

[Yu 97] J. Yu & Y. Xiang. *Hypermedia Presentation and Authoring System*, Proceedings : the 6th International World Wide Web Conference, pp. 153-164, Avril 1997 : <http://gatekeeper.dec.com/pub/DEC/SRC/technical-notes/SRC-1997-003-html/>.

RESUME

Cette thèse s'inscrit dans le volet édition / présentation des documents multimédias. Le travail consiste à modéliser le contenu des médias simples (texte, image) et complexes (vidéo, audio) regroupés dans des scénarios temporels et spatiaux de façon à obtenir des présentations multimédias riches et flexibles. Outre l'objectif de couvrir les besoins de description des auteurs pour réaliser des synchronisations aussi bien synthétiques que naturelles entre médias, le système "Lumiere" proposé permet de s'assurer que ces scénarios restent cohérents par rapport aux compositions mises en place par les auteurs des documents et pouvant être troublées par les interactions des lecteurs. La modélisation de ses scénarios par une extension des réseaux de Petri permet d'exécuter des contrôles de vérification. L'objectif de cette thèse est donc de fournir un support, une méthodologie de conception et un modèle théorique qui permettent de représenter une application multimédia, et plus particulièrement, à la fois son comportement, son architecture, la gestion de ses données ainsi que ses aspects temporels et spatiaux tout en respectant la qualité des documents à présenter.

Mots Clés : Multimédia, Synchronisation, Modèle de document, Vérification, Qualité de Service.

Abstract

This work proposes a new way to edit/present easily multimedia documents. It consists in modeling the contents of simple (text, image) and complex media (video, audio). These elements when grouped in temporal and spatial scenarios enable rich and flexible multimedia presentations thanks to natural and synthetic synchronization between media. The proposed system "Lumiere" also insures that each scenario remains consistent with the interaction of the document readers thanks to verifications executed upon a Petri nets modeling. The objective of this work is thus to provide a support, a design methodology and a theoretical model which make it possible to represent a multimedia application, and more particularly, its behavior, its architecture, the management of its data and all the associated temporal and space aspects while respecting the quality of the documents to be presented.

Key words: Multimedia, Synchronization, Document Models, Verification, Quality of Service.

INTRODUCTION GENERALE

Introduction	I
Motivation et Objectifs	II
Plan de la thèse	IV

PARTIE 1 - LES DOCUMENTS MULTIMEDIAS (DMMs)

Chapitre 1 - NOTIONS DE BASE

1 MEDIAS	1
<i>1.1 Définition d'un média</i>	1
<i>1.2 Principaux médias</i>	2
<i>1.2.1 Texte</i>	2
<i>1.2.2 Image</i>	3
<i>1.2.3 Son</i>	5
<i>1.2.4 Vidéo</i>	5
<i>1.2.5 Programme</i>	6
<i>1.3 Typologie des médias</i>	8
2 DOCUMENTS MULTIMEDIAS	10
<i>2.1 Définition</i>	10
<i>2.2 Processus de production</i>	11
<i>2.3 Dimensions des documents multimédias</i>	13
<i>2.4 Qualité de service et DMMs</i>	17
3 CONCLUSION	19

Chapitre 2 - Modélisation de la synchronisation des DMMs

1 SYNCHRONISATION TEMPORELLE	20
<i>1.1 Caractéristiques de la synchronisation</i>	21
<i>1.1.1 Les formes des compositions temporelles</i>	21
1.1.1.1 Synchronisation intra- objet	21
1.1.1.2 Synchronisation inter- objets	22

1.1.2	<i>L'indéterminisme</i>	24
1.1.2.1	Instants de début et de fin indéterminés	24
1.1.2.2	Délais systèmes	24
1.2	Modèles temporels pour les DMMs	25
1.2.1	<i>Modèles à base d'instants</i>	28
	Timeline	28
	Réseaux de points temporels	29
1.2.2	<i>Modèles à événements</i>	29
	Langages de script	30
1.2.3	<i>Modèles à intervalles</i>	31
	Logique d'intervalle	31
	Relations d'Allen	32
	Relations de Wahl- Rothermel	33
	Modèles de réseaux de Petri	34
1.2.4	<i>Les modèles hybrides</i>	36
	Multimedia Re-presentation Graph	36
	Connecteurs hypermédias	37
	DAMSEL	37
	Algèbre de processus	38
1.3	Contrôle du temps dans les DMMs	39
1.3.1	<i>Qu'est ce qu'un document consistant ?</i>	39
1.3.2	<i>Classification</i>	40
1.3.2.1	Consistance qualitative	40
1.3.2.2	Consistance quantitative	40
1.3.2.3	Consistance générique	41
1.3.2.4	Consistance spécifique à l'application	42
1.3.3	<i>Modèles et techniques de contrôle d'intégrité</i>	42
1.3.4	<i>Synthèse</i>	44
2	SYNCHRONISATION SPATIALE	45
2.1	<i>Modèles absolus</i>	45
2.2	<i>Modèles relatifs</i>	46
3	CONCLUSION	47

Chapitre 3 - Outils et standards Multimédias

1 HYTIME	48
2 PREMO	50
3 MHEG	53
4 MADEUS	56
5 MPGS	58
6 SMIL	61
7 DISCUSSION	64

PARTIE 2 - LUMIERE : UN SYSTEME AUTEUR POUR LES DMMs

Chapitre 4 - MDO2 : Un modèle de documents orienté Objet

1 STRUCTURE DU CONTENU	68
<i>1.1 Types des composants</i>	69
<i>1.2 Attributs des composants</i>	70
<i>1.2.1 Attributs temporels</i>	71
<i>1.2.2 Attributs spatiaux</i>	73
<i>1.2.3 Autres</i>	74
a Composants d'intérêt	74
b Déplacement de composants	75
2 STRUCTURE SYNCHRONISATION	76
<i>2.1 Synchronisation temporelle</i>	76
<i>2.1.1 Principe de la gestion du temps</i>	76
<i>2.1.2 Mise en œuvre de la synchronisation</i>	78
<i>2.2 Composition spatiale</i>	82
3 STRUCTURE DE PRESENTATION	83
<i>3.1 Contrôle de la présentation</i>	84
<i>3.2 Contrôle de l'environnement</i>	86
4 ILLUSTRATION DE LA SPECIFICATION D'UN DMM DANS MDO2	87
<i>4.1 Spécification textuelle et graphique</i>	87
<i>4.2 Modélisation en MDO2</i>	89
<i>4.3 Interactions admises</i>	91

5 CONCLUSION	91
Chapitre 5 - Vérification des documents multimédias	
1 RESEAUX DE PETRI TEMPORISES AVEC ARC DE LECTURE	94
<i>1.1 Présentation</i>	95
<i>1.2 Propriétés et particularités</i>	96
<i>1.3 Exemple de RdP_AL</i>	98
2 DOCUMENTS MULTIMEDIAS ET RDP_MAL	98
2.1 Traduction d'un document multimédia vers un RdP-MAL	99
<i>2.1.1 Correspondance et règles de construction</i>	99
2.1.1.1 Les places	99
2.1.1.2 Les transitions et les arcs	101
2.1.1.3 Les représentations des relations de synchronisation	101
2.1.1.4 La définition formelle	104
<i>2.1.2 Exemple</i>	105
<i>2.1.3 Gestion des interactions</i>	106
2.2 Analyse des scénarios temporels	108
<i>2.2.1 Propriétés vérifiées</i>	108
<i>2.2.3 Processus de et techniques de vérification</i>	109
2.3 Gestion de l'indéterminisme	110
3 SYNTHÈSE	111
Chapitre 6 - Architecture et Interface de Lumière	
1 CYCLE DE VIE D'UN DOCUMENT LUMIERE	112
2 ARCHITECTURE DE REFERENCE DE LUMIERE	114
2.1 Editeur	115
2.2 Valideur	117
2.3 Exécuteur	118
3 SYNTHÈSE	120

CONCLUSION GENERALE

1 DEMARCHE ET BILAN DU TRAVAIL	121
<i>1.1 Multimédia et environnements "auteur" de documents multimédias</i>	122
<i>1.2 Systèmes auteur pour les documents multimédias</i>	122
2 PERSPECTIVES	123
REFERENCES	126