

# Sommaire

<b>Chapitre I: Introduction Générale</b> .....	1
I.1 Contexte d'étude.....	1
I.2 Problématique.....	2
I.3 Plan du mémoire .....	2
<b>Chapitre II : Les Documents Multimédias (DMM)</b> .....	4
II.1 Introduction.....	4
II.2 MEDIAS .....	6
II.2.1 Définition d'un média .....	6
II.2.2 Principaux médias.....	6
II.2.2.1 Texte.....	6
II.2.2.2 Image.....	6
II.2.2.3 Son .....	7
II.2.2.4 Vidéo .....	7
II.2.3 Typologie des médias .....	7
II.3 DOCUMENTS MULTIMEDIAS .....	8
II.3.1 Définition .....	8
II.3.2 Processus de création d'un DMM.....	8
II.3.2.1 L'édition.....	9
II.3.2.2 Le contrôle.....	9
II.3.2.2.1 Cohérence qualitative .....	10
II.3.2.2.2 Cohérence quantitative .....	10
II.3.2.3 La présentation .....	11
II.3.3 Dimensions des documents multimédias .....	11
II.3.3.1 Dimension spatiale.....	11
II.3.3.2 Dimension temporelle .....	12
II.3.3.3 Dimension hypermédia .....	12
II.3.3.4 Dimension logique .....	13
II.4 Synchronisation temporelle des DMM .....	13
II.4.1 Synchronisation intra-media.....	14
II.4.2 Synchronisation inter-média.....	14
II.4.3 Synchronisation des lèvres .....	15
II.5 Synchronisation spatiale des DMM.....	15
II.5.1 Modèles absolus .....	16
II.5.2 Modèles relatifs .....	16
II.6 Modèles temporels pour les DMM.....	17

II.6.1 Modèles à base d'instants.....	19
II.6.1.1 TimeLine .....	19
II.6.1.2 Réseaux de point temporels .....	20
II.6.2 Modèles à événements .....	20
II.6.2.1 Langages de script .....	21
II.6.3 Modèles à intervalles .....	21
II.6.3.1 Relations d'Allen .....	21
II.6.3.2 Relations de Wahl-Rothermel.....	22
II.7 Les standards de structuration des DMM .....	23
II.7.1 SGML .....	23
II.7.2 HyTime .....	24
II.7.3. XML.....	24
II.7.4 ODA.....	25
II.7.5 MHEG.....	26
II.7.6 SMIL.....	26
II.8 Conclusion .....	27
<b>Chapitre III : Le langage SMIL : Etat de l'Art des approches de validation .....</b>	<b>28</b>
III.1 Introduction.....	28
III.2 Définition de SMIL .....	29
III.3 Les versions de SMIL.....	29
III.4 Les avantages de SMIL .....	30
III.5 La modularisation et le profilage.....	31
III.5.1 Structure d'un document SMIL.....	32
III.5.2 Module Méta-Information.....	34
III.5.3 Module Layout .....	35
III.5.4 module Media Objects .....	36
III.5.5 Module de temporisation et de Synchronisation.....	38
III.5.5.1 Les attributs de temporisation .....	38
III.5.5.2 Les éléments de synchronisation .....	39
III.6 Le module Linking .....	41
III.7 Le module Content Control.....	43
III.8 Le module Animation .....	44
III.9 Le module Transitions .....	44
III.10 Analyse des document SMIL .....	45
III.10.1 Structuration .....	45
III.10.2 Navigation .....	45
III.10.3 Placement temporel et synchronisation.....	46
III.11 Cohérence des documents SMIL.....	47
III.11.1 Conflit temporel intra-élément .....	47
III.11.2 Conflit temporel inter-éléments.....	48

III.12	Inconsistances des documents SMIL.....	48
III.13	Etat de l'art des approches de validation des documents SMIL.....	50
III.14	Etude comparative des différentes approches de validation de SMIL.....	52
III.15	Conclusion.....	54
<b>Chapitre IV : La logique de Hoare.....</b>		<b>55</b>
IV.1	Introduction.....	55
IV.2	Logique de Hoare.....	56
IV.3	Les triplets de Hoare.....	56
IV.4	Correction d'un triplet de Hoare.....	57
IV.4.1	Correction partielle.....	57
IV.4.2	Correction totale.....	57
IV.5	Les assertions.....	57
IV.6	La règle d'inférence.....	58
IV.7	Arbre de déduction.....	59
IV.7.1	Arbre de déduction complet.....	59
IV.8	Jugement prouvable.....	59
IV.9	Les règles de Hoare.....	59
IV.9.1	Affectation.....	60
IV.9.2	Séquence.....	60
IV.9.3	Conséquence.....	60
IV.9.4	Conditionnelle.....	61
IV.9.5	While.....	61
<b>IV.10</b>	<b>Conclusion.....</b>	<b>62</b>
<b>Chapitre V : Conception et Réalisation d'un Outil de Modélisation et de</b>		<b>63</b>
<b>Validation des Documents SMIL.....</b>		<b>63</b>
V.1	Introduction.....	63
V.2	Notre proposition.....	63
V.3	Définition d'une lecture abstraite.....	64
V.4	Définition d'état d'un lecteur.....	64
V.5	Les type d'événements.....	65
V.5.1	Les événements interactifs.....	65
V.5.2	Les événements non interactifs.....	66
V.6	Le langage d'assertion pour SMIL.....	66
V.7	Conventions de notation.....	67
V.8	Les règles d'inférence pour SMIL.....	68
V.8.1	Règles de preuve pour les objets médias.....	69
V.8.2	Règle de preuve pour la composition parallèle 'PAR'.....	71
V.8.3	Règle de preuve pour la composition séquentielle 'SEQ'.....	73
V.8.4	Règle de preuve pour la composition exclusive 'EXCL'.....	74

---

V.8.5 Règle de preuve pour une composition générale d'éléments lorsque l'attribut Dur est Défini .....	76
V.8.6 Règles de preuve pour les éléments SMIL avec un événement interactif dans 'Begin' ou 'end' .....	76
V.8.7 Preuve règle pour de multiples exécutions d'un même élément .....	78
V.8.8 Règle de Preuve pour les éléments qui contiennent des événements interactifs à la fois dans Begin et End .....	79
V.8.9 Gestion des événements non interactifs .....	80
V.8.10 Règle de preuve pour la composition parallèle lorsque l'attribut « EndSync » est égal à 'First' .....	81
V.8.11 Règle de preuve pour l'attribut « MIN » .....	82
V.8.12 Règle de preuve pour l'attribut « MAX » .....	82
V.8.13 Règle de preuve pour l'attribut « RepeatDur » .....	82
V.9 Implémentation .....	83
V.10 Environnement de développement JBuilder .....	83
V.11 Architecture générale de l'outil .....	84
V.12 Conception UML .....	85
V.13 Captures d'écrans de l'application .....	88
<b>Conclusion Générale &amp; Perspectives.....</b>	<b>90</b>
<b>Bibliographie.....</b>	<b>92</b>
<b>Annexe A.....</b>	<b>96</b>

## Liste des Figures

### Chapitre II

Figure II.1 : Définition d'un DMM .....	.....
Figure II.2 : Information temporelle attachée aux objets .....	.....
Figure II.3 : Le Time Line d'un DMM .....	.....
Figure II.4 : Les réseaux de points temporels .....	.....
Figure II.5 : Le modèle des relations de Wahl and Rothermel.....	.....
Figure II.6 : Trois relations spatiales directionnelles entre deux points .....	.....
Figure II.7 : Les quatre dimensions d'un DMM.....	.....
Figure II.8 : Dimension spatiale .....	.....
Figure II.9 : Dimension temporelle d'un DMM.....	.....
Figure II.10 : Dimension hypermédia d'un DMM .....	.....
Figure II.11 : Dimension logique d'un DMM.....	.....
Figure II.12 : Exemple d'une synchronisation intra-média .....	.....
Figure II.13 : Exemple d'une synchronisation inter-média.....	.....
Figure II.14 : Exemple d'une synchronisation des lèvres ( <i>lip-synchronization</i> ).....	.....
Figure II.15 : Exemple d'un scénario valide .....	.....
Figure II.16 : Exemple d'un scénario invalide.....	.....
Figure II.17 : Exemple d'un scénario valide .....	.....
Figure II.18 : Exemple d'un scénario invalide.....	.....

### Chapitre III

Figure III.1 : Signification du langage SMIL.....	.....
Figure III.2 : Exemple d'un document SMIL.....	.....
Figure III.3 : Structure d'un document SMIL .....	.....
Figure III.4 : Exemple d'éléments joués en parallèle .....	.....
Figure III.5 : Exemple d'éléments joué en séquence .....	.....
Figure III.6 :Exemple d'éléments jouées en parallèle et en séquence .....	.....
Figure III.7 :Exemple de synchronisation en enchainement .....	.....
Figure III.8 : Exemple d'une partie temporelle cliquable .....	.....
Figure III.9 : Conflit temporel intra-élément.....	.....
Figure III.10 : Conflit temporel inter-éléments .....	.....

**Chapitre V**

Figure V.1 : Les Règles (Static+Begin, Cont+Begin, Madia+Begin+End+Dur).....	
Figure V.2 : La règle PAR+Begin+End.....	
Figure V.3 : La règle SEQ+Begin+End.....	
Figure V.4 : La règle EXCL+Begin+End.....	
Figure V.5 : La règle CMD+Begin+End+Dur.....	
Figure V.6 : la règle Begin+User-Interaction et End+ User-Interaction.....	
Figure V.7 : La règle Begin+Multiple-Playbacks.....	
Figure V.8 : La règle Begin+End+Multiple-Playbacks.....	
Figure V.9 : La règle PAR+EndSync = First.....	
Figure V.10 : La règle CMD+MIN.....	
Figure V.11 : La règle CMD+MAX.....	
Figure V.12 : La règle CMD+RepeatDur.....	
Figure V.13: Environnement JBulider 2006.....	
Figure V.14: Architecture générale de l'outil.....	
Figure V.15 Diagramme des objets.....	
Figure V.16: La class SMOB.....	
Figure V.17 : La class ATT (Begin ou End).....	
Figure V.18 : La class Condition.....	
Figure V.19 : La class M (les règles d'inférence).....	
Figure V.20 : La class MT (Le moteur d inference).....	
Figure V.21 : L' outil SMIL_Validator.....	

---

## Liste des Tableaux

### Chapitre II

Tableau II.1 – Les Relations d’Allen.....

### Chapitre III

Tableau III.1. Fonction et extensions des éléments médias .....

Tableau III.2. Les attributs Media Objects.....

Tableau III.3. Les valeurs acceptées par l’attribut fill .....

Tableau III.4 Les valeurs acceptées par l’attribut endsync .....

Tableau III.5 Les attributs de test de <switch/> .....

Tableau III.6 Calcul de la durée active et situations de conflit intra-élément.....

Tableau III.7 Comparaison de différentes approches.....

Tableau III.8 Comparaison entre l’approche de [A. Bossi, O. Gaggi,] et notre approche.....

### Chapitre V

Tableau V.1 les éléments SMIL, attributs, abréviation utilisée .....

Tableau V.2 liste des fonctions et notation utilisé pour définit les règles d’inférence. ....

Tableau V.3 liste des prédicats et ensembles utilisé pour définit les règles d’inférence. ....

## Chapitre I: Introduction Générale

### I.1 Contexte d'étude

La manipulation digitale, de graphique, de son et d'images animées sur des stations de travail ou des ordinateurs personnels a changé la nature d'un grand nombre d'applications. En particulier, les applications de traitement de documents électroniques, habituellement dédiées à la création et à la présentation de données textuelles et graphiques, trouvent dans le multimédia des possibilités nouvelles. L'information qu'elle manipule est plus riche, puisqu'elle intègre dans ces documents du son et des images animées. Ces nouveaux types de documents électroniques sont communément appelés documents multimédia (DMM en abrégé).

Lorsque nous parlons d'un document, nous pensons généralement à un texte imprimé sur papier. Mais tout ensemble d'informations constitue un document: un article de journal, une émission de télévision, une chanson ou un jeu vidéo interactif. Grâce à l'enregistrement numérique, les documents sont faciles à trouver, à mémoriser, à charger et à présenter. Cependant, malgré ces progrès, les DMM sont très peu organisés au niveau temporel, alors que des documents par exemple éducatifs (encyclopédie, atlas, cours...) nécessitent des scénarios très planifiés prenant en compte les interactions avec l'utilisateur.

Ces nouveaux documents électroniques ne contiennent pas seulement des représentations statiques comme le texte (titre, paragraphe, sections), les images, les interactions (hyperliens), mais contiennent aussi de nouvelles représentations basées sur la synchronisation temporelle des différents ingrédients (image, audio, vidéo, texte) composant les documents. Nous passons ainsi d'un simple document statique, où les relations entre les ingrédients sont spatiales, vers un document dynamique où les relations entre les ingrédients intègrent une dimension temporelle. Les ingrédients sont ainsi projetés dans le temps donnant naissance à une suite d'actions (ex. afficher un texte pendant 10 secondes, jouer des séquences audio et vidéo pendant 50 secondes suivis d'un commentaire textuel, etc.).

Depuis plusieurs années, des standards tels que SGML [ISO-86], ODA [ISO-89], HyTime [ISO92-b, ISO-97], MHEG [Meye-95] et HyperODA [Ccit-92] sont apparus. Il s'agit de langages de spécification de présentations multimédia qui sont associés à des environnements de compilation et d'exécution. Ces standards ne suffisent plus pour représenter les documents avec leur dimension temporelle. De ce fait, un nouveau standard comme SMIL [W3C-00, Bult-02] a émergé pour les compléter. Malgré ses nombreux avantages, beaucoup de choses restent à faire, particulièrement dans le domaine de la formalisation, l'expérimentation et la persistance dans les bases de données.



L'apparition du facteur temporel avec les données multimédia a imposé de nouvelles contraintes et de nouvelles approches dans la réalisation des DMM. Celles-ci vont donc naturellement se répercuter à un niveau d'abstraction plus élevé avec la définition des présentations multimédia. La spécification de scénarios temporels de présentation multimédia nécessite de nouveaux supports formels.

## **I.2 Problématique**

La création et la conception d'une présentation multimédia est une activité complexe et source d'erreurs. En effet, la présentation des objets multimédia peut être totalement modifiée par l'utilisateur quand il s'agit d'un lien de navigation hypermédia ou d'un clic d'une image.

De nombreux chercheurs ont proposé différentes solutions pour ce problème en utilisant les relations temporelles (par exemple, [Allen, J.F.], AHM [Hardman, L., Bulterman,]), des outils de modélisation (par exemple, [Bulterman, D.C.A., Hardman, L.]), ou des langues de synchronisation (par exemple, SMIL [Bulterman, D.C.A., Rutledge, L.W.]). Malgré toutes ces propositions, la création d'une présentation multimédia semble être difficile et mérite donc un intérêt considérable de la part du concepteur des DMM.

L'apparition du langage de présentation et de synchronisation SMIL en sa première version [W3C 1998] a fortement contribué dans la création et la présentation des DMM permettant ainsi la modélisation, la synchronisation et la présentation des média de nature différente (images, son, texte, vidéo, animation, ...) sur le Web. SMIL est basé sur le langage XML (eXtensible Markup Language), méta-langage permettant la définition de langage à base de balises. Il permet de créer des présentations multimédia interactives grâce à l'usage de balises et d'attributs définis dans des modules.

L'auteur d'un DMM est censé développer un code SMIL syntaxiquement correcte mais n'admet pas la possibilité de le valider manuellement du point de vue sémantique. En effet, un DMM peut comporter des incohérences temporelles ou des inconstances qui ne peuvent être vérifiées que lors de son exécution. Pour cela, il faudra disposer d'outils formels de validation.

Ce mémoire a pour but de contribuer à la modélisation et à la validation d'un document SMIL en utilisant la logique de Hoare [Hoa69]. Notre travail consiste donc à concevoir et à réaliser un outil d'aide à la vérification de cohérence temporelle intra et inter-média et de consistance d'un document SMIL.

### I.3 Plan du mémoire

Notre mémoire est organisé comme suit :

**Chapitre I :** présente le contexte d'étude auquel nous nous sommes intéressés et explique la problématique en question.

**Chapitre II :** est une introduction au multimédia et aux documents multimédia.

**Chapitre III :** concerne l'étude du langage SMIL et un état de l'art des différentes approches de sa validation.

**Chapitre IV :** introduit la logique de Hoare pour concevoir notre outil de modélisation et de validation d'un document SMIL.

**Chapitre V :** concerne la conception et la réalisation de notre outil de modélisation et de validation des documents SMIL.

**Annexe A :** résumé l'expérimentation de notre outil en utilisant une étude de cas du document SMIL de publicité de jeux vidéo sur PS4 (METAL GEAR SOLID 4).

---

## Chapitre II : Les Documents Multimédias (DMM)

### II.1 Introduction

De nos jours, la notion de document multimédia est devenue de plus en plus répandue et utilisée dans différents domaines comme : le téléenseignement, la visioconférence, la publicité, la création des sites web, l'infographie, etc. Le **tableau II.2** récapitule les types d'application que l'on peut trouver selon leurs disponibilités en ligne ou hors ligne.

Les différentes sources d'information sur les ventes de produits et de services multimédias indiquent toutes un niveau de croissance des revenus très important depuis le début des années 90, croissance qui devrait se poursuivre dans la prochaine décennie. Les données de l'étude internationale de Frost et Sullivan, *World multimedia application markets*, indiquent une augmentation importante du secteur (voir le **tableau II.1**).

Ces documents sont caractérisés par l'intégration et la synchronisation de plusieurs médias (son, vidéo, image,...), grâce à un objet fédérateur, l'ordinateur, on peut envisager des définitions beaucoup plus larges qui soulignent le phénomène de la transformation des médias. En effet la révolution multimédia est la résultante de l'évolution de trois familles de médias :

**L'informatique** : grâce à des langages évolués (comme Hypertalk, Lingo, Java, etc.), le producteur multimédia n'a plus besoin d'être un informaticien pour réaliser des documents qui traitent synthétiquement à la fois le texte, le son et l'image sur un CD-ROM, un CD-I, un DVD ou Internet.

**La vidéo numérique** : elle oblige la télévision à évoluer vers la TVI (la télévision interactive) et la vidéo à la demande dans les réseaux de câblodistribution.

**Les réseaux** : ils accueillent le trafic d'Internet et des grandes inforoutes de l'avenir où circulent déjà toutes sortes de services interactifs comme le courrier électronique, la téléphonie vocale et la vidéoconférence, le commerce électronique et bientôt peut-être la télévision à la demande.

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.

	Amérique du Nord	Europe	Pacifique	Total
1995	4,1	3,3	1,7	9,8
2000	6,1	7,4	5,5	22,2
Croissance annuelle	8,3 %	17,5 %	26,5 %	17,8 %

Source: Frost et Sullivan, 1994.

**Tableau II.1** les ventes des produits multimédia en milliard de dollars.

TYPES D'APPLICATIONS	HORS LIGNE	EN LIGNE
<b>Grand public – Loisirs</b>		
Film, animation	Vidéo, CD ou DVD	Vidéo à la demande
Jeux	Jeux vidéo, CD-ROM et CD-I	Jeux à la demande
Musée, tourisme	Collection sur CD-ROM Visite guidée virtuelle	Consultation d'œuvres à distance Information, réservation, paiement
Livres	Livre interactif	Édition et commande de livres à la demande
<b>Grand public – Information</b>		
Presse	Compilation de journaux sur CD-ROM	Information à la demande
Bibliothèque	Ouvrages, livres sur CD-ROM	Consultation à distance
Reference	Encyclopédies, dictionnaires sur CD-ROM	Interrogation à distance
<b>Grand public – Commercial</b>		
Achat	Catalogues sur CD-ROM	Commande, paiement à distance
Publicité	Matériel promotionnel sur CD-ROM	Messages publicitaires interactifs en ligne
Banque	n. d.	Gestion du compte et transactions
<b>Entreprise</b>		
Communication	Présentation corporative sur CD-ROM	Videophone
Formation	CD-ROM de formation	Vidéoconférence
Information	Banques de données sur CD-ROM	Banque de données en ligne
<b>Éducation</b>		
Enseignement	CD-ROM et CD-I d'apprentissage	Télé-éducation, téléclasse, éducation à distance
Reference	Encyclopédies, dictionnaires sur CD-ROM	Interrogation à distance
Ludo-éducatif	Jeux éducatifs sur CD-ROM	Jeux éducatifs en ligne

**Tableau II.2** les types d'application multimédia.

## II.2 MEDIAS

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

### II.2.1 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].

### II.2.2 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.

#### II.2.2.1 Texte

C'est le média artificiel le plus ancien au monde et celui qui a fait l'objet des premiers développements en informatique. 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 œuvre littéraire comme un article ou un livre.

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.

#### II.2.2.2 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). 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*.

### II.2.2.3 Son

Le son est l'un des deux principaux médias utilisés aujourd'hui, avec la vidéo. L'audio inclut les données de type vocal : paroles digitalisées, musique digitalisée, synthèse de paroles, etc. 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. 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.

### II.2.2.4 Vidéo

La vidéo est constituée d'une succession d'images animées (ou trame d'images). La manipulation, la consultation et la recherche des informations peuvent être réalisées sur chacune des images qui constituent cette vidéo. Les données de type vidéo sont de grandes consommatrices de ressources systèmes telles que l'espace sur support de stockage secondaire et sont représentées par le format de codage MPEG. A titre d'exemple, une heure de vidéo compressée au format MPEG-1 nécessite 675 Giga-octets pour une qualité VHS standard.

## II.2.3 Typologie des médias

Les objets multimédia peuvent être classés en deux catégories de par leur mode de présentation dans le temps [Laya 97] [Rois 99] :

- **Les objets discrets** : Ce sont des médias qui n'ont aucune dépendance temporelle, et leur contenu est délivré de façon instantanée, comme le texte et les images statiques.
- **Les objets continus** : Ce sont des médias qui ont une dépendance temporelle, et leur contenu est délivré de façon progressive comme les vidéos, les sons ou les animations.

Ils peuvent aussi être caractérisés par leur mode de perception:

- **Les objets visibles**, c'est-à-dire que l'on peut afficher : texte, image, vidéo, animation.
- **Les objets audibles** comme les sons.

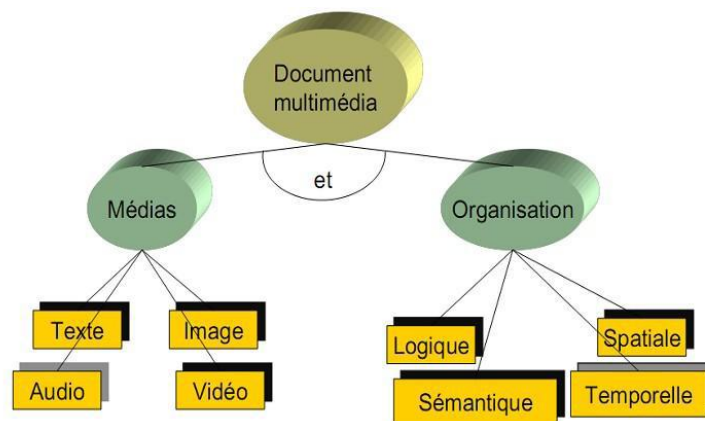
Pour chaque catégorie de médias, différents formats de codage existent (ASCII pour le texte, JPEG, GIF, TIFF, BMP pour l'image, AU, MP3, WAV pour l'audio, MPEG, AVI, ... pour la vidéo, etc.).

### II.3 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.

#### II.3.1 Définition

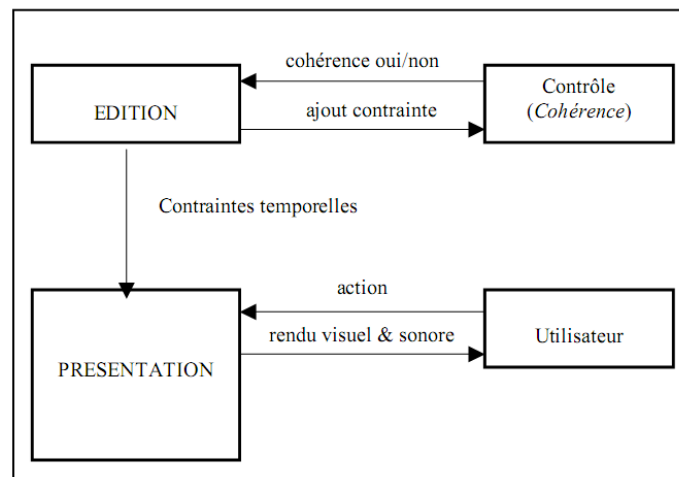
Un document est qualifié de multimédia s'il supporte la présentation intégrée de plusieurs médias dont au moins un est de nature temporisée [Blakowski 96]. ou bien, un document multimédia est toujours décrit comme étant un assemblage, selon un scénario spécifié, de médias traditionnels dits statiques (texte, graphique), tout aussi bien que des médias continus (animation, audio et vidéo), des médias structurés (documents HTML, SMIL, SVG) [Bulterman 05]. Un DMM est une collection de médias (texte, audio, image, vidéo ou animation), il est composé de quatre dimensions: la dimension temporelle qui spécifie la synchronisation des médias dans le temps, la dimension spatiale qui définit la disposition spatiale des médias, la dimension logique qui permet de définir une structure ou un regroupement de certains médias sous une même entité et la dimension hypermédia (sémantique) qui définit les mécanismes de navigation dans le document voire (**figure II.1**).



**Figure II.1** : Définition d'un DMM [laya 97]

#### II.3.2 Processus de création d'un DMM

Une présentation multimédia est aussi un assemblage de données multimédia créées afin d'être présentée selon la sémantique spatio-temporelle définie par cet assemblage. Elle fournit la plupart du temps une abstraction des mécanismes de la mise en œuvre de sa sémantique et peut donc être interprétée par une variété d'applications différentes. Elle est à la fois pilotée par les données qui la composent (média), par les enchaînements temporels et par les flux de contrôle (interaction, navigation). La production de présentations multimédia s'organise autour d'un cycle qui peut être diversifiée mais dans lequel on retrouve néanmoins toujours un certain nombre de phases caractéristiques que nous décrivons dans la **Figure II.2**.



**Figure II.2:** Cycle de vie d'un DMM

### II.3.2.1 L'édition

Dans cette phase, l'auteur utilise un outil multimédia, application ou langage, afin de définir l'assemblage des données multimédia dont il dispose. Ces outils facilitent le travail de spécification de l'auteur en lui fournissant des moyens techniques ou conceptuels permettant de la définir.

Les applications d'édition fournissent une certaine rapidité de développement grâce à l'usage d'interface graphique, alors que les langages exhibent des notions pertinentes dans lesquelles l'auteur pourra transcrire ses idées. Selon l'approche suivie, l'auteur aura ou non le choix du format final de sa présentation multimédia.

### II.3.2.2 Le contrôle

Lors de l'élaboration du scénario, c'est-à-dire au moment d'énoncer les relations temporelles et de fixer les ensembles de tolérance des durées des objets, il faut vérifier en temps-réel que l'auteur ne produit pas un système de contraintes irréalisable, et donc vérifier la cohérence d'un scénario. Il est également important d'essayer d'indiquer la ou les contraintes qui entraînent l'incohérence du scénario.

La création du scénario d'un document est un processus incrémental : l'auteur crée un premier scénario et doit être sûr de suite qu'il est cohérent, il le pré visualisé, revient au scénario, ajoute/supprime une relation. Il est donc nécessaire à la fois de vérifier en temps-réel la cohérence temporelle et spatiale du document (on parle aussi de sa validation).

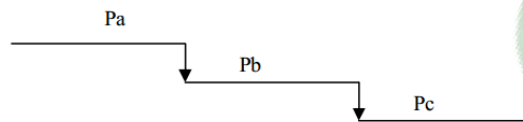
En effet, la présentation d'un DMM dépend des contraintes de synchronisation temporelle établies entre les objets multimédia de ce document à la phase d'édition. Ces contraintes spécifiées via des relations temporelles et causales, peuvent s'avérer incompatibles les unes avec les autres conduisant ainsi à des situations de dysfonctionnement sinon de blocage dites d'incohérence ou d'inconsistance. Alors l'analyse de l'état du scénario temporel en cours de composition est très importante. La cohérence temporelle qualitative ou quantitative doit être vérifiée après le formatage du DMM.



### II.3.2.2.1 Cohérence qualitative

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

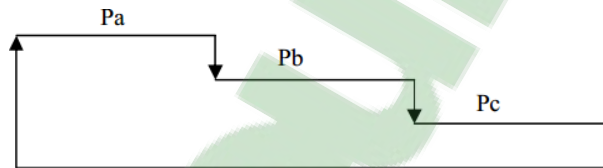
Exemple : soient trois médias 'a', 'b' et 'c', reconnus par leurs durées de présentations respectives:  $P_a$ ,  $P_b$  et  $P_c$ , et reliés par les relations *meets* (a, b) et *meets* (b, c) comme c'est schématisé dans la **Figure II.3** suivante :



**Figure II.3:** Exemple d'un scénario valide

Il est à remarquer qu'aucune incohérence n'est relevée sur le scénario de la **Figure II.3**.

**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', comme montre la **Figure II.4**.



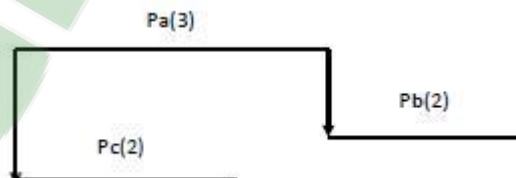
**Figure II.4 :** Exemple d'un scénario invalide

Dans la **Figure II.4**, la fin du scénario n'est jamais atteinte c'est donc un cas d'incohérence qualitative.

### II.3.2.2.2 Cohérence quantitative

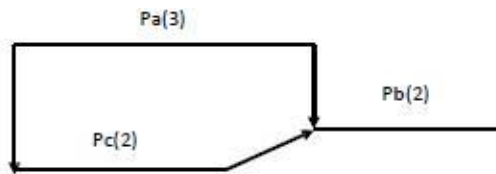
Cette consistance concerne la validité des combinaisons des relations temporelles parmi les objets médias de la présentation par rapport à leurs durées d'exécution.

Exemple: soient toujours nos trois médias 'a', 'b' et 'c', reliés par les relations *starts* (a, c) et *meets* (a, b) avec les durées respectives 30s, 20s et 20s comme c'est schématisé dans la **Figure II.5** suivante :



**Figure II.5:** Exemple d'un scénario valide

**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 le montre la **Figure II.6** suivante :



**Figure II.6:** Exemple d'un scénario invalide

L'objet 'b' ne peut pas être déclenché par 'a' et 'c', en même temps, car le premier termine après le deuxième. Donc c'est un cas d'incohérence quantitative.

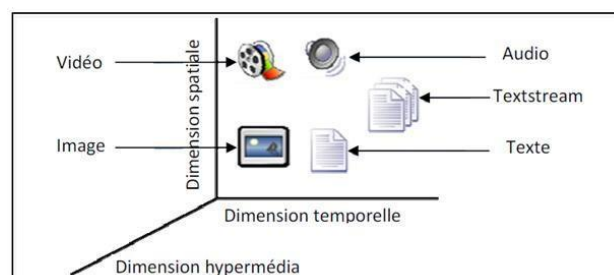
### II.3.2.3 La présentation

Finalement, un utilisateur va restituer, ou encore exécuter, la présentation 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 nécessite un système d'exécution particulier selon son format. L'exécution peut nécessiter le transfert de données, voire d'applications, via un réseau et l'interrogation d'une base de données.

La présentation offre à l'auteur de voir la présentation finale pour vérifier que le comportement de son DMM correspond bien au scénario désiré.

### II.3.3 Dimensions des documents multimédias

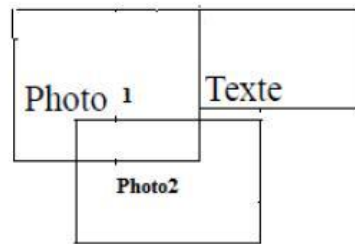
La présentation des documents électroniques à l'utilisateur a jusqu'ici été effectuée principalement en fonction de leur structure spatiale et structure temporelle et structure logique et structure hypermédias, Il est donc possible d'analyser un DMM selon quatre dimensions [Rois 99] voire (**Figure II.7**):



**Figure II.7:** Les quatre dimensions d'un DMM

#### II.3.3.1 Dimension spatiale

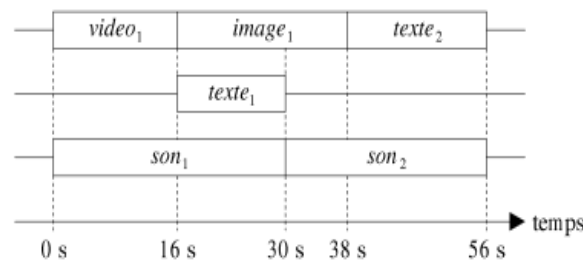
La spécification de l'organisation spatiale définit, quant à elle, le placement, sur écran, des objets médias les uns par rapport les autres pendant la présentation [Carcone 97]. En réalité, un média dans un scénario occupe un espace (fenêtre), spécifié par les coordonnées du coin supérieur gauche et de la taille de cette fenêtre (hauteur, largeur). Il peut, ou non, partagé cet espace avec un (ou plusieurs) média(s) selon l'espace d'intersection entre sa fenêtre et celle(s) de son (ses) voisin(s), Cette dimension est illustrée dans **la figure II.8**.



**Figure II.8 :** Dimension spatiale d'une présentation multimédia.

### II.3.3.2 Dimension temporelle

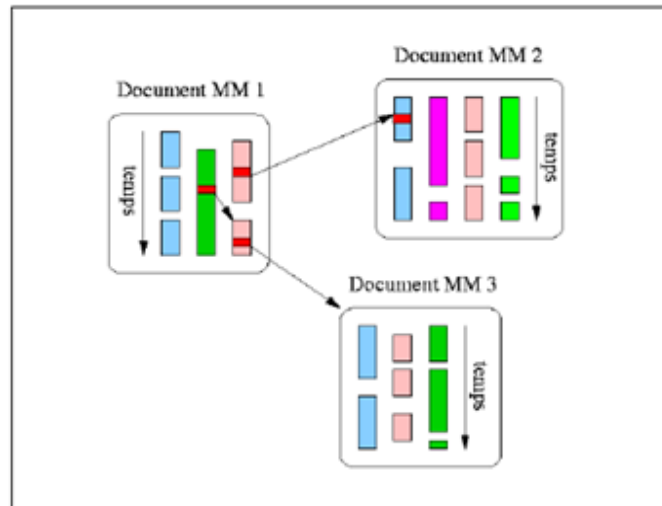
La structure temporelle qui décrit l'enchaînement des éléments dans le temps dont certains ont eux-mêmes une dimension temporelle (vidéo, audio). La spécification de l'organisation temporelle définit l'ordonnement temporel de la présentation des différents composants afin de construire le scénario souhaité. 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 Cette dimension est illustrée dans **la figure II.9**.



**Figure II.9:** Dimension temporelle d'un DMM

### II.3.3.3 Dimension hypermédia

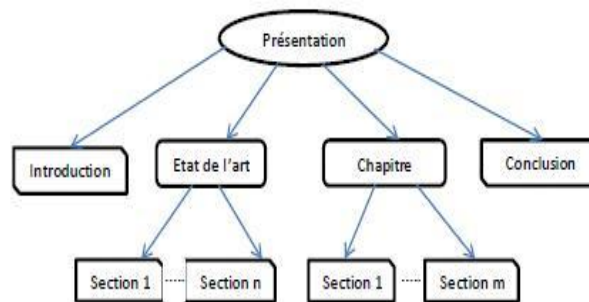
La structure hypermédia correspond à l'organisation des données dans un réseau de nœuds connectés par des liens qui définissent des relations sémantique entre les nœuds. L'ancrage, notion introduite par le modèle de référence hypertexte [Halasz 90], permet de spécifier la partie de l'information qui 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 Cette dimension est illustrée dans **la figure II.10**.



**Figure II.10:** Dimension hypermédia d'un DMM

### II.3.3.4 Dimension logique

Dans un document multimédia, il est possible de distinguer des parties du document qui sont sémantiquement liées et qui peuvent être regroupées par le biais de relation logique. Regrouper les éléments sous une même entité revient à définir une structure logique du document, c'est-à-dire une structure hiérarchique des éléments multimédias. La **figure II.11** présente une structure logique d'un DMM.



**Figure II.11:** Dimension logique d'un DMM.

### II.4 Synchronisation temporelle des DMM

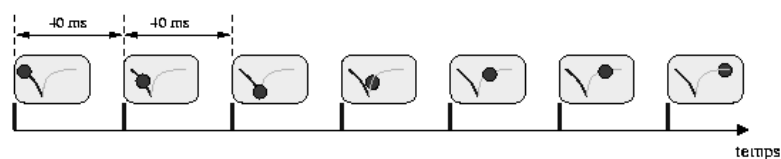
Les données médias sont plongées dans le temps, même si leur comportement n'en dépend pas toujours, et confèrent aux présentations multimédias la même propriété intrinsèque. Parmi les multiples facteurs à prendre en compte dans une présentation multimédia, le temps est le plus important car c'est une 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.

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.

#### II.4.1 Synchronisation intra-média

La synchronisation intra-média désigne les relations temporelles d'unités d'information de base d'un même média. 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. D'après [Blak 96] Ce type de synchronisation est associé à l'objet média continu, vidéo par exemple, qui est considéré comme une suite ordonnée d'unités de présentation ayant des relations temporelles implicites entre elles. Par exemple pour la synchronisation entre les images successives d'une séquence vidéo d'une vitesse de présentation de 25 images par seconde, chaque image de la séquence doit être affichée à l'utilisateur pour une durée de 40 millisecondes (**Figure II.12**).



**Figure II.12:** Exemple d'une synchronisation intra-média

#### II.4.2 Synchronisation inter-média

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].

La synchronisation inter-média peut être synthétique (à gros grain) ou naturelle (fine). La granularité des éléments composés ainsi que les relations temporelles entre eux représentent les critères de classification.

La synchronisation synthétique inter-média 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-média 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. Donc, elle s'applique pour enchaîner les différents médias du document. Par exemple, la présentation simultanée d'un élément vidéo et d'un audio suivie par un élément de type image (Figure II.13).

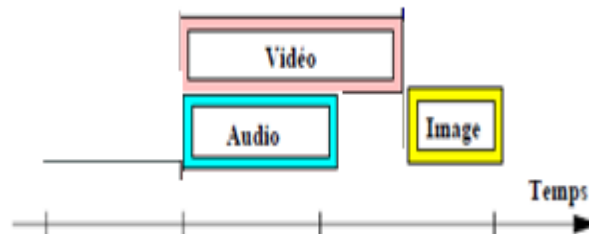


Figure II.13 : Exemple d'une synchronisation inter-média

### II.4.3 Synchronisation des lèvres

Par contre, la synchronisation naturelle représente une combinaison des deux dernières. A noter que ce type de synchronisation est appelé également synchronisation de lèvres (*lip-synchronization*).

La synchronisation des lèvres impose un couplage temporel fort entre la progression temporelle de deux ou plusieurs objets multimédia (leurs flux). Ce couplage est généralement exprimé en terme de décalage temporel admissible entre les flux (*skew*).

L'exemple typique de cette forme de synchronisation correspond à la présentation simultanée d'un discours audio et de la séquence vidéo associée. Ces contraintes permettent d'exprimer les conditions de maintien de la voix en accord avec le mouvement des lèvres de la personne. Cette forme de synchronisation n'est cependant pas réservée au couplage des médias de type audio et vidéo. On peut la retrouver dans d'autres situations, comme celle illustrée dans la Figure II.14.

Dans cet exemple, un élément graphique (carré) est utilisé pour annoter un élément en mouvement dans une séquence vidéo (Avion) [Chen 95].

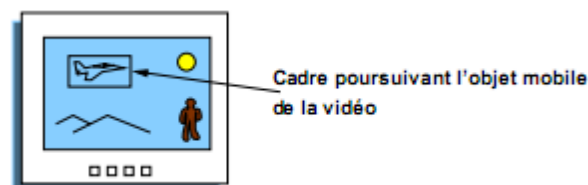


Figure II.14 : Exemple d'une synchronisation des lèvres (*lip-synchronization*)

### II.5 Synchronisation spatiale des DMM

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 (ç 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.

### II.5.1 Modèles absolus

Ce type de spécification spatiale repose sur une description de l'espace en terme de référentiel, ç 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.

### II.5.2 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 spatiotemporel 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- disjointives 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,  $\zeta$  à  $d$ , sans recouvrements, entre les régions dénommées granularités.


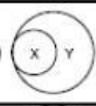






DC (disconnected)		TPP (tangential proper part)	
PO (partial overlap)		TPP <sup>-1</sup> (TPP inverse)	
EC (externally connected)		NTPP (non-tangential proper part)	
EQ (equal)		NTPP <sup>-1</sup> (NTPP inverse)	

Figure II. 15 – 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.

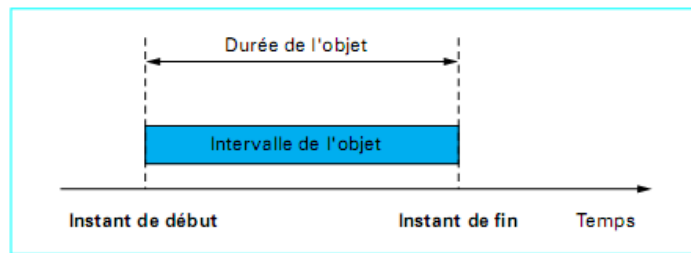
**Les modèles basés sur la distance :** se basent sur la définition d'une distance entre objets.

## II.6 Modèles temporels pour les DMM

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 (**Figure II.16**):



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



**Figure II.16 :** Information temporelle attachée aux objets.

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.

### II.6.1 Modèles à base d'instantants

Dans les relations à base d'**instantants** [Hamb 72], les unités temporelles considérées dans les relations sont les instants de début et de fin des objets. Étant donné deux instants dans un scénario, trois relations peuvent exister entre eux. Un instant peut en précéder un autre ( $<$ ), lui succéder ( $>$ ) ou lui être égal, c'est-à-dire coïncider temporellement ( $=$ ).

Avec ce modèle, on peut exprimer la position relative de deux éléments A et B selon trois relations : *A avant B* ( $A < B$ ), *A égale à B* ( $a = b$ ) et *A après B* ( $A > B$ ). On peut citer comme modèles de temps utilisés :

#### II.6.1.1 TimeLine

L'axe de temps est utilisé pour représenter la dimension temporelle d'un DMM. On place les instants sur un axe de temps et on leur affecte des dates fixes. En plus, chaque média est représenté par une bande qui est manipulé indépendamment des autres objets médias et leurs événements, tels que le début ou la fin d'un segment, sont totalement ordonnés sur l'axe de temps (**Figure II.17**). De même, nous pouvons trouver ce modèle dans le domaine des réseaux sous forme d'estampille [Sant 93][Bult 05].

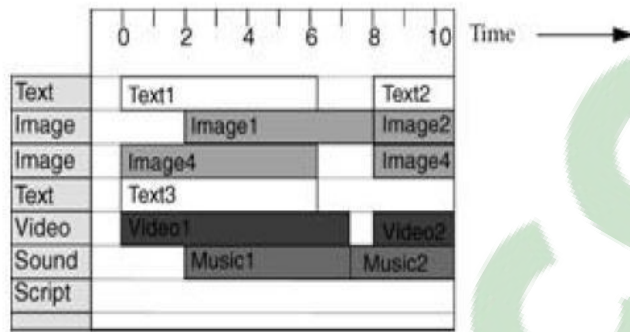


Figure II.17 : Le TimeLine d'un DMM

### II.6.1.2 Réseaux de point temporels

Selon Buchanan [Buch 93] [Buch 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 (figure II.18). Les points peuvent être en début ou en fin de données médias. L'étendue entre le début et la fin est un arc qui peut être contraint à travers trois valeurs : minimale, maximale et optimale.

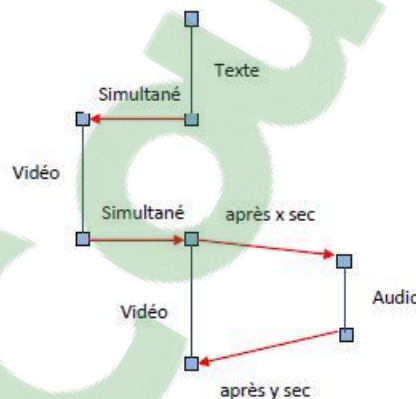


Figure II.18 : Les réseaux de points temporels

### II.6.2 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.

### II.6.2.1 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 donnée, 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 II.19 – Description d'un document au moyen de scripts (Système Lingo)

### II.6.3 Modèles à intervalles

Un intervalle est défini par deux instants, un de début et un de fin, et possède une durée non nulle lorsque ces deux instants sont distincts [Bult 95]. L'approche par intervalles se prête bien à la conception multimédia car elle permet de représenter les différents éléments de façon purement temporelle en faisant abstraction de leur nature réelle. L'intervalle représente alors la durée de présentation de l'élément en question.

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

#### II.6.3.1 Relations d'Allen

Le modèle le plus général, proposé par Allen [Alle 83], dresse la liste exhaustive de toutes ces relations. La liste des combinaisons possibles entre les éléments multimédias comporte ainsi 13 relations

consistant en 7 relations de base (**Tableau II.3**) et leurs relations inverses, l'égalité étant elle-même son inverse. Ces treize relations se répartissent en deux classes: celle des relations de séquentialité notée **seq**, et celle des relations introduisant le parallélisme de présentation, notée **par**.

Dans l'avant-dernière colonne du **Tableau II.3**, nous présentons la traduction de chaque relation sous la forme d'une suite de relations à base d'instant. Les variables x et y représentent des intervalles et les notations x- et x+ correspondent respectivement aux instants de début et de fin de l'intervalle x.

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

Tableau II.3 – Les Relations d'Allen

**II.6.3.2 Relations de Wahl-Rothermel**

A partir d'une analyse systématique de l'approche par instants et par intervalles, Wahl et Rothermel [Wahl 94] ont proposé un modèle reposant sur les intervalles. L'idée est de supprimer l'ambiguïté inhérente à certaines des relations d'Allen. Ils ont définis trois valeurs possibles pour le délai (0 pour un délai nul, + pour un délai positif et \* pour un délai positif ou nul) et en exploitant les régularités entre les relations d'intervalles, les auteurs ont ramené de 29 à 10 les relations d'intervalles utiles pour le multimédia (**Figure II.20**). Ils ont ainsi augmenté l'expressivité des 29 relations tout en simplifiant l'édition d'un scénario.

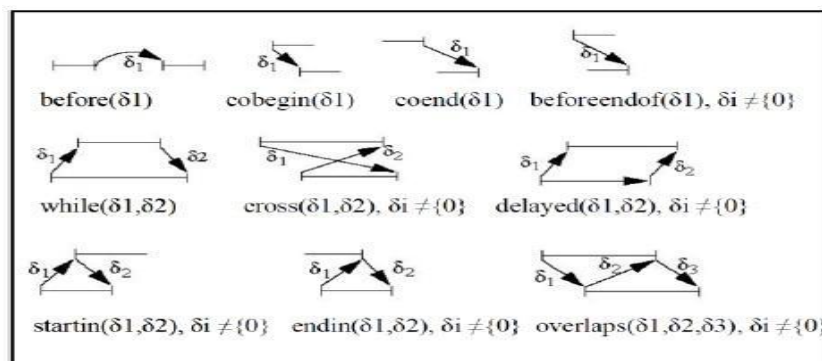


Figure II.20 : Le modèle des relations de Wahl and Rothermel

## II.7 Les standards de structuration des DMM

Les normes et les standards de structuration ont pour objet d'homogénéiser la présentation des documents et d'augmenter leur lisibilité afin de faciliter leur consultation et leur échange. Ces derniers doivent incorporer les différents média ainsi que leur composition, et sont indispensables pour assurer le fonctionnement des projets sur de multiples plates-formes et une durée de vie suffisante pour garantir les retours sur investissements. Dans la cette section, nous décrivons les principales normes et standards (SGML, HYTIME, HTML et XML, XHTML, SMIL) qui ont marqué et marquent à ce jour l'histoire des DMM.

### II.7.1 SGML

SGML (Standard Generalized Markup Language) [ISO 86] autorise une structuration de l'information à l'aide de balises. Une balise désigne une marque particulière ajoutée à un texte afin d'en déduire sa structure ou le format dans lequel il sera édité [Gold 81]. SGML a été adopté officiellement en octobre 1986 comme standard international [ISO 86]. La structure globale d'un document SGML est la suivante :

- (1) un ensemble de déclarations où sont précisées les caractéristiques SGML utilisées telles que la version, le jeu de caractères utilisé, etc. ;
- (2) une Définition de Type de Document (DTD) qui décrit la structure logique et exprime la manière dont les différents éléments d'information sont organisés et articulés entre eux ;
- (3) un contenu (ses données qui seront rédigées dans des instances).

Le standard SGML permet de définir des classes de documents, c'est à dire des documents ayant la même structure logique, et ce, indépendamment de leurs formats d'édition. Cette structure logique est définie dans la DTD sous la forme d'une arborescence qui indique tous les éléments que peut contenir une classe de documents SGML, ainsi que les contraintes d'organisation. La DTD doit décrire les balises qui limitent les différents composants, ainsi que les règles d'utilisation des balises (l'organisation hiérarchique des différents éléments d'une DTD). L'interprétation d'un document SGML nécessite donc la connaissance de la DTD. Les auteurs de documents utilisent ensuite cette DTD pour rédiger les documents, au moyen d'éditeurs de texte. Une fois le document créé, un analyseur syntaxique le transforme en un document dont le balisage est complet.

En effet, cet analyseur restitue les balises volontairement omises (par souci de simplification) lors de la saisie et vérifie si tous les éléments utilisés ont bien un modèle de contenu conforme à la DTD. Un des intérêts du standard SGML est que la structure logique est indépendante de la présentation de l'information et donc des moyens de restitution (séparation entre la structure logique et la structure physique). Ainsi, il est possible, à partir du même fichier, de réaliser des présentations visuelles différentes.

De plus, le fait de pouvoir spécifier des règles d'utilisation des balises dans une DTD, est un aspect très important de SGML. En effet, ceci permet d'imposer une uniformité aux documents d'un même type rendant plus aisé l'exploitation de l'information contenue dans les documents d'une même classe ou DTD. Bien que précurseur dans le domaine de la structuration normalisée de documents, l'usage de SGML tend à disparaître au profit de HTML pour le Web et surtout de XML qui en reprend les principaux avantages.

### II.7.2 HyTime

Le standard HyTime (HYpermédia/Time-based Structuring Language) [ISO 92] est une extension du noyau SGML qui permet l'intégration de structures hypertextes par l'utilisation de balises de renvoi. Elle se base sur la mise en place d'une méta-DTD (document de base) qui relie des DTD élémentaires entre elles (documents hyperliés).

HyTime est le premier standard qui représente les dépendances temporelles et qui cherche à exprimer l'ensemble des liens hypertextuels, et permet de décrire la structure de document, qui peut inclure tout type de données (texte, images fixes, audio, vidéo, ..) sans format de codage spécifique. La description des modules HyTime [Babe 93] est comme suit :

**Module de base:** fournit les facilités requises par les autres modules. Il inclut un modèle lexical pour décrire le contenu des éléments, il offre la possibilité de définir des entités capables de supporter de nombreuses données. Ce module permet la gestion élémentaire des documents.

**Module de mesure:** un objet appartenant à un document HyTime doit pouvoir être localisé dans le temps et/ou dans l'espace, ou dans tout autre domaine défini par la DTD.

**Module d'adressage:** en plus des propriétés d'adressage fournies par SGML, ce module permet de référencer indirectement les données externes, et les éléments non référencés classiquement par SGML.

**Module d'Hyperlink:** deux types de liens sont définis d'après leur forme architecturale: le lien contextuel et le lien indépendant. Le premier a deux ancres, dont l'une est incluse dans le document, et l'autre est référencée comme la cible. Le second a un nombre indéterminé d'ancres qui ne sont pas forcément spécifiées dans le document.

### II.7.3. XML

XML (EXtensible Markup Language) [6] est un langage de description et d'échange de données semi-structurées. Il permet de décrire la structure logique des documents à l'aide d'un système de balisage. XML permet de marquer les éléments qui composent la structure et les relations entre ces éléments. La structure d'un document XML est formée par :

- (1) un prologue qui est un ensemble de déclarations dont la présence est facultative, mais conseillée;
- (2) un arbre d'éléments qui forme le contenu proprement dit du document ;

(3) des commentaires et des instructions de traitement dont la présence est facultative et qui peuvent être soit dans le prologue, soit dans l'arbre d'éléments.

XML utilise des balises et des attributs comme SGML mais laisse à l'utilisateur l'entière possibilité de définir son propre jeu de balises dans le but de personnaliser la structure des documents. Le standard XML a été conçu pour être utilisé de deux manières distinctes [Mich 98] :

- d'une part, nous pouvons utiliser une DTD, comme pour les documents SGML, qui spécifie la structure logique d'une classe de documents et définit les balises à utiliser pour identifier les entités de cette structure. Le document XML faisant appel à cette DTD est dit "document valide" ;

- d'autre part, un document XML peut être écrit sans DTD, il est alors dit "document bien formé". Dans ce cas, le document doit respecter la syntaxe du standard XML. Ainsi, il ne peut comporter aucune ambiguïté dans le balisage: tous les éléments doivent posséder une balise ouvrante et fermante, les attributs doivent être entre guillemets, etc.

Dans XML, la séparation de la description structurelle des documents et de la description de leurs présentations physiques offre d'énormes avantages en terme de facilités d'échange et de production coopérative de documents et surtout une possibilité accrue en terme de traitement automatisé de documents. XML apporte également une souplesse au niveau des liens hypertextes dont l'utilisation est plus aisée que celle de HYTIME. XML est devenu le standard d'échange d'informations sur Internet.

Pour assurer une meilleure présentation des DMM et pour gérer leur stockage ainsi que leur diffusion, XML est complété par de nouveaux langages comme XSL pour la gestion des styles et XLINK pour la description des liens. XSL (eXtensible Styling Language) est le langage utilisé pour la définition des feuilles de styles qui sont associées aux documents XML. C'est le fichier XSL qui permet de définir les propriétés typographiques et graphiques du document (un élément XML doit être affiché avec telle police de caractères, telle couleur, etc.). Une feuille de styles XSL se compose des règles de construction décrivant comment les éléments du fichier XML doivent être transformés vers le format de sortie (un document HTML, un texte, etc.). XLink et le langage utilisé pour la création à la fois de liens unidirectionnels simples et de structures de liaison plus complexes. Il permet d'associer des métadonnées à un lien et d'exprimer des liens qui résident dans une localisation séparée des ressources reliées.

#### II.7.4 ODA

ODA (Office Document Architecture) [Dani 91] se réfère aux notions de structure logique et de structure physique, et qui est aussi un moyen pour garantir des formats d'échange entre différents environnements. Cette norme est fondée sur le principe de coexistence des "vues" logiques et physiques au sein d'un même document. Ces deux vues ou structures sont indépendantes et complémentaires, mais il ressort de cette norme que la structure logique ne joue pas un rôle dominant. ODA permet de créer



des documents qui se réfèrent à des classes prédéfinies de documents ou d'objets. La structure logique est constituée de composants logiques à structure répétitive (chapitre, sous- chapitre, ...) et la structure physique est constituée de composants physiques à structure répétitive (page, notes de bas de page, ...).

### II.7.5 MHEG

Group MHEG (Multimedia and Hypermedia Information coding Expert) [MHEG 92] est issu d'un comité de l'Organisation Internationale de Normalisation ISO (ISO/IEC JTC1/SC29 WG12) pour résoudre les problèmes de présentation et de codage des applications multimédias et hypermédias. Le groupe MHEG est créé en 1989 à l'investigation de l'organisme national français de normalisation, l'Afnor. Cette norme MHEG est l'élément clé des systèmes de communication multimédia. C'est une interface de programmation de haut niveau.

MHEG doit fournir un format standard pour l'échange de données provenant de divers types de médias: images fixes, audio, vidéo. Les unités d'échange sont manipulées par un moteur MHEG sous le contrôle d'une application indépendante. De plus, MHEG permet de structurer les différents médias à l'intérieur d'une même unité d'échange (d'où le nom de multimédia). La norme MHEG définit la représentation codée des entités d'information finies, qui sont des éléments de l'information multimédia/hypermédia qui peuvent être utilisées et échangées les applications en temps réel. Les objets multimédia définis par la norme MHEG sont destinés à être utilisés dans les recommandations CCITT [Ccitt 92], ISO ainsi que d'autres normes et architectures et des applications définies par l'utilisateur.

### II.7.6 SMIL

SMIL (*Synchronized Multimedia Integration Language*) est un standard du W3C [1], et c'est aussi un langage d'édition et de présentation des DMM qui permet de décrire des présentations multimédia synchronisées ou des scénarios multimédia mettant en œuvre les média de base (audio, vidéo, image et texte) afin de les consulter en temps réel et de façon interactive. SMIL est un format d'intégration, c'est-à-dire qu'il ne décrit pas le contenu des objets média faisant partie d'une présentation multimédia, mais plutôt leur composition temporelle et spatiale ainsi que les hyperliens entre ces objets. Son principe consiste à construire des compositions séquentielles ou parallèles de média de base. SMIL est basé sur XML. L'auteur d'un document SMIL peut ainsi créer et éditer facilement des présentations multimédia, en ayant la possibilité de :

- décrire le comportement temporel d'une présentation. SMIL utilise les balises <seq> et <par> pour spécifier qu'un ensemble d'objets est joué respectivement en séquence ou en parallèle. La durée d'un objet peut être spécifiée par un délai par rapport à la date de début ou de fin d'un autre objet ;
- construire des média complexes par des combinaisons de compositions séquentielles et/ou parallèles (respectivement des combinaisons de <seq> et/ou <par>). Les contraintes de temps sont soit implicites, c'est-à-dire calculées à partir de celles des composants, soit spécifiées dans les balises ;

- décrire le placement des objets média sur l'écran pendant la présentation. Sur la fenêtre principale de présentation, SMIL spécifie des régions dont la position et la taille sont exprimées soit en valeur absolue, soit en pourcentage de la taille de la fenêtre principale ;
- associer des hyperliens aux objets média. La désignation de la destination d'un lien est effectuée en terme d'adresse URI (Uniform Resource Identifier).

### **II.8 Conclusion**

Les DMM 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.

Ces DMM est caractérisé par leur structure spatiale (présentation graphique et mise en page), de leur structure logique (organisation en chapitres, sections, paragraphes, etc.) et de leur structure sémantique (hypertexte). 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, interaction de l'utilisateur...). Les documents étudiés ici sont donc multimédia, temporisés et interactifs.

L'introduction de la dimension temporelle et spatiale dans les documents conduit nécessairement à de nouveaux standards pour la conception et l'édition des DMM : comme XML, MHEG, ODA .....SMIL, ....etc. Le standard SMIL est le plus utilisé à l'heure actuelle pour la synchronisation et la présentation des DMM complexes, mais ne garantit pas leur qualité de présentation.

La complexité de la synchronisation temporelle et spatiale des documents est telle qu'il est nécessaire de développer des techniques de modélisation et de validation formelle pour garantir la validité d'un scénario.

## Chapitre III : Le langage SMIL : Etat de l'Art des approches de validation

### III.1 Introduction

Alors que le Web a augmenté dans l'utilisation de dizaines millions à des centaines de millions au cours des dernières années, les fournisseurs de contenu et les audiences web ont appris à apprécier et valoriser les multimédia d'impact peuvent avoir sur la création d'expériences riches. La norme de langage de balisage qui définit le Web, HTML, son origine dans permettant la composition et la présentation des médias statiques (texte, images).

Le W3C a cherché à travailler avec les membres de la communauté Internet à définir un langage de balisage qui serait spécifiquement conçu pour répondre aux exigences uniques de dynamique des médias, et ouvrir la voie à la prochaine vague de contenu Web dynamique.

SMIL est une norme proposée qui déverrouille le potentiel du Web pour synchroniser une large gamme de multimédia basés sur le temps et les combiner en une seule présentation.

Synchronized Multimedia Integration Language (SMIL) [W3C.SMIL] est un langage déclaratif défini et recommandé par le W3C depuis 1998 qui a pour but de permettre l'intégration de contenus multimédias diversifiés (images, sons, textes, vidéo, animations, hypertexte) en les synchronisant afin de permettre la création de présentations multimédias.

La structure d'un document SMIL décrit le déroulement temporel et spatial des différents composants intégrés dans une présentation multimédia, sous forme de documents XML. Il permet donc d'indiquer le moment où un contenu sera affiché, pendant combien de temps et dans quelle partie de la fenêtre d'affichage. Pour faciliter l'utilisation de ce standard dans les contextes très variés (desktop, Télévision, téléphone portable,...), deux mécanismes ont été mis en place :

**La modularisation** est une approche dans laquelle la fonctionnalité de balisage est spécifiée par un ensemble de modules qui regroupent chacun des sous-ensembles sémantiques par d'éléments XML de SMIL, d'attributs, et de leurs valeurs.

**Le profilage** est la création d'un langage basé sur XML en combinant des modules différents afin de fournir les fonctionnalités pour les applications particulières.

Dans SMIL 3.0, les modules sont regroupés dans 4 profils : Langage profile, Mobile profile, Basic profile, et Extended Mobile profile.

Il existe des lecteurs capables de lire SMIL (QuickTime, Ambulant, RealPlayer). Quelques navigateurs ont aussi implémenté certains modules de SMIL permettant de lire parti de ce langage au sein d'autres langages. Par exemple, Firefox et Opera peuvent jouer des animations de SMIL. En plus le profil MMS (Multimedia Messaging Service) est un profile de SMIL 3.0 destiné aux utilisateurs de téléphones mobiles par offrir un service de messagerie multimédia et est supporté par la plupart de réseaux téléphoniques et des téléphones portables.

### III.2 Définition de SMIL

SMIL (Synchronized Multimedia Integration Language), est un langage de spécification créé en 1998 par un groupe de travail appelé SYMM (SYnchronised MultiMedia), de la W3C (World Wide Web Consortium), qui permet de décrire l'organisation spatiale et temporelle d'une présentation multimédia. Il permet de spécifier des documents qui incluent divers éléments multimédia locaux ou résidant sur des serveurs distants et de les synchroniser pour créer des documents interactifs tout en s'adaptant à plusieurs paramètres.

SMIL est un langage de balisage basé sur le standard XML (eXtensible Markup Language). La structure du document SMIL décrit l'organisation temporelle d'objets multimédias (aspect temporel), spécifie leur disposition spatiale (aspect spatial), et définit des liens hypermédias (aspect hypermédia), sous forme de documents XML.

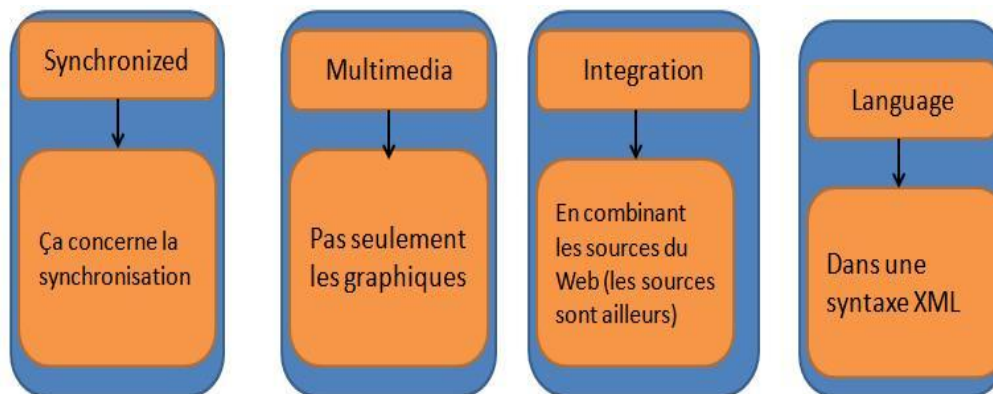


Figure III. 1 Signification du langage SMIL

### III.3 Les versions de SMIL

SMIL 1.0, depuis son apparition dans sa première version 1.0 le 15 juin 1998 [1] a connu une évolution constante grâce au groupe de travail SYMM (Synchronized Multimedia Working Group) du W3C. Au début, le groupe a instauré les bases du langage permettant une spécification de DMM pour:

- permettre la synchronisation des objets multimédia de manière plus ou moins flexible (aspect temporel).
- permettre le placement visuel des objets multimédias de la présentation (aspect spatial).
- permettre le référencement des objets multimédias (aspect hypermédia).

Le 7 août 2001, une deuxième version SMIL 2.0 [2] a été publiée avec de nouveaux modules qui viennent s'ajouter offrant plus de flexibilité et de fonctionnalités par exemple : le module méta-information et le module animation.

SMIL 2.1, la troisième version publiée en 2005[3], a amené beaucoup de changement par l'ajout

## Chapitre III

des modules d'effets et de définition des profils du langage en DTD (Document Type Definition), à exécuter sur des plateformes mobiles (comme SMIL basic profil).

En fin 2008, la quatrième version SMIL 3.0 [4] est élaborée, celle-ci définit de nouveaux modules et profils très puissants. Cette version a été défini pour pouvoir utiliser une plus grande variété de terminaux dans le réseau, non seulement les ordinateurs mais aussi les appareils portatifs de poche tels que les téléphones cellulaires, les Blackberries et même des machines du jeu. Ainsi cette version permet la lecture audio de textes, ce qui est très utile pour les non-voyants et les utilisateurs de livres électroniques. Dans notre mémoire nous nous intéressons aux documents SMIL 3.0.

### III.4 Les avantages de SMIL

Le langage SMIL grâce à ces différentes versions a investi une grande partie du marché du multimédia, surtout celui du mobile, et de plus en plus d'applications utilisent ce standard :

Les MMS (Multimedia Messaging Service) ce format largement utilisé sur les téléphones mobiles, utilise un sous-ensemble de modules du langage SMIL.

HTML-TIME est un langage permettant l'ajout de la dimension temporelle aux pages HTML, ceci est réalisé par l'inclusion des éléments temporels provenant du langage SMIL.

SVG (Scalable Vector Graphics), est un format de documents basé sur XML pour la description des ensembles de graphiques vectoriels, dont sa partie animation est basé sur le module d'animation du langage SMIL.

DAISY-DTB (Digital Accessible Information System-Digital Talking Book) est basé sur le standard XHTML et SMIL, permettant la lecture d'un format de livre électronique par synthèse vocale.

En raison de la large utilisation de SMIL, de nombreux outils d'édition et de lecture multimédia exécutant SMIL ont été développés, comme :

#### Les éditeurs

- LimSee 2.0 [7] pour SMIL 2.0 et LimSee 3.0 [8] pour SMIL 2.1 et SMIL 3.0
- GRINS [9] pour SMIL 2.0 ;
- Adobe GoLive [10] ;

#### Les lecteurs

- QuickTime [12] pour SMIL 1.0 ;
- RealOne Player [13] pour SMIL 2.0 ;

## Chapitre III

---

- Ambulant 2.0.2 [14] pour SMIL 2.1 et SMIL 3.0.
- Internet Explorer [15] pour HTML-TIME.
- PoketSMIL 2.0 [16] exécutant des documents SMIL 2.0 sur PDA.
- Rubic [17] exécutant des MMS basées sur SMIL.

Tous ces outils d'édition et de lecture, peuvent être utilisés pour créer plusieurs types de présentations, tout en exploitant les diverses fonctionnalités qu'offre SMIL, dont on cite quelques-unes :

- SMIL peut être utilisé pour créer des présentations Internet ou Intranet ;
- Il peut être utilisé pour créer des présentations slide-show ;
- Il est décrit comme l'image Internet de PowerPoint ;
- Les présentations SMIL peuvent afficher de multiples types de fichiers (texte, vidéo, audio.);
- Elles peuvent afficher de multiples fichiers en même temps ;
- Elles peuvent afficher de multiples fichiers à partir de multiple serveurs web, grâce aux URL ;
- Elles peuvent contenir des liens à d'autres présentations SMIL ;
- Elles peuvent contenir des boutons de contrôle (stop, start, next, ...) ;
- SMIL a des fonctions pour définir les séquences et la durée des éléments ;
- Il a des fonctions pour définir la position et la visibilité des éléments ;
- Etc.

### III.5 La modularisation et le profilage

La modularisation est une approche dans laquelle la fonctionnalité de balisage est spécifiée comme un ensemble de modules sémantiquement liés aux éléments, attributs et valeurs d'attribut XML. La modularisation permet aux concepteurs du langage de spécifier le balisage dédié destiné à l'intégration avec d'autres profils existants. Des exemples de telles spécifications destinées à l'intégration sont : MathML [18], et XForms [19].

Le profilage est la création d'un langage basé sur XML par le biais de la combinaison de ces modules, afin de fournir la fonctionnalité requise par une application particulière. Le profilage introduit la possibilité d'adapter un langage basé sur XML (dans notre cas c'est le langage SMIL) à

## Chapitre III

des besoins spécifiques, par exemple, pour optimiser la présentation et l'interaction aux capacités du client. Le profilage ajoute également la capacité d'intégration d'une fonctionnalité à partir d'autres langages de balisage, Il prévoit en outre la cohérence de balisage par l'utilisation du même modèle à intégrer. Le profilage en plus de la facilité de création de présentations offre la possibilité de réutilisation de code. Par exemple, un code en SMIL qui contient des fonctionnalités de synchronisation peut être utilisé pour des documents codés en XHTML + SMIL, et des documents codés en SVG.

La modularisation et le profilage utilisent des propriétés d'extensibilité de XML, et des technologies connexes comme les espaces de nommages XML (XML-NS) et XML Schéma (XSCHEMA).

### III.5.1 Structure d'un document SMIL

L'élément racine d'un document SMIL est l'élément *Smil*. Cet élément contient deux éléments fils *body* et *head*. L'élément *head* ne contient pas des informations temporelles mais renseigne sur l'auteur et la présentation spatiale. L'élément *body* contient des informations sur les média et leurs relations temporelles. Un document SMIL est structuré sous forme de conteneurs temporels (*time containers en anglais*), appelés également éléments composites ou opérateurs. Un opérateur comprend une sémantique temporelle particulière qui permet de définir le placement des objets média. Ces opérateurs sont les éléments *seq*, *par* et *excl*.

Le conteneur temporel *seq* (séquence) définit une présentation en séquence des ressources média. Le conteneur temporel *par* (parallèle) permet de jouer les ressources en parallèle. Finalement, l'opérateur *excl* est basé sur *par* mais en ajoutant la contrainte que seul un objet enfant soit joué à un moment donné.

L'exemple de la **Figure III.2** présente les principaux éléments d'une présentation multimédia SMIL :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 3.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL30.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL30/Language">

<head>
  <layout type="text/smil-basic-layout">
    <root-layout id="rootLayout" width="640" height="480"/>
```

```
<region id="region1"left="-24"top="30"width="252" height="174"/>
<region id="region2"left="282"top="6"width="222" height="164"/>
</layout>
</head>
<body >
  <par>
    <seq begin="0s" id="seq_1">
      
      <text src="text" region="region2"/>
    </seq>
    <par id="par1" endsync="first" dur="10s">
      <a href="http://www.w3.org/ns/smil">
        <video src="video1" region="region2"/>
      </a>
      <text src="text1" region="region2"/>
    </par>
    <seq id="seq2" end="6s">
      </seq>
    </par>
  </body>
</smil>
```

Figure III.2 : Exemple d'un document SMIL

Le module *Structure* est un module obligatoire pour les profils de langage conformes au langage hôte SMIL et fournit les éléments de base pour structurer un document SMIL. Il est composé des éléments : **smil**, **head** et **body** comme suit :

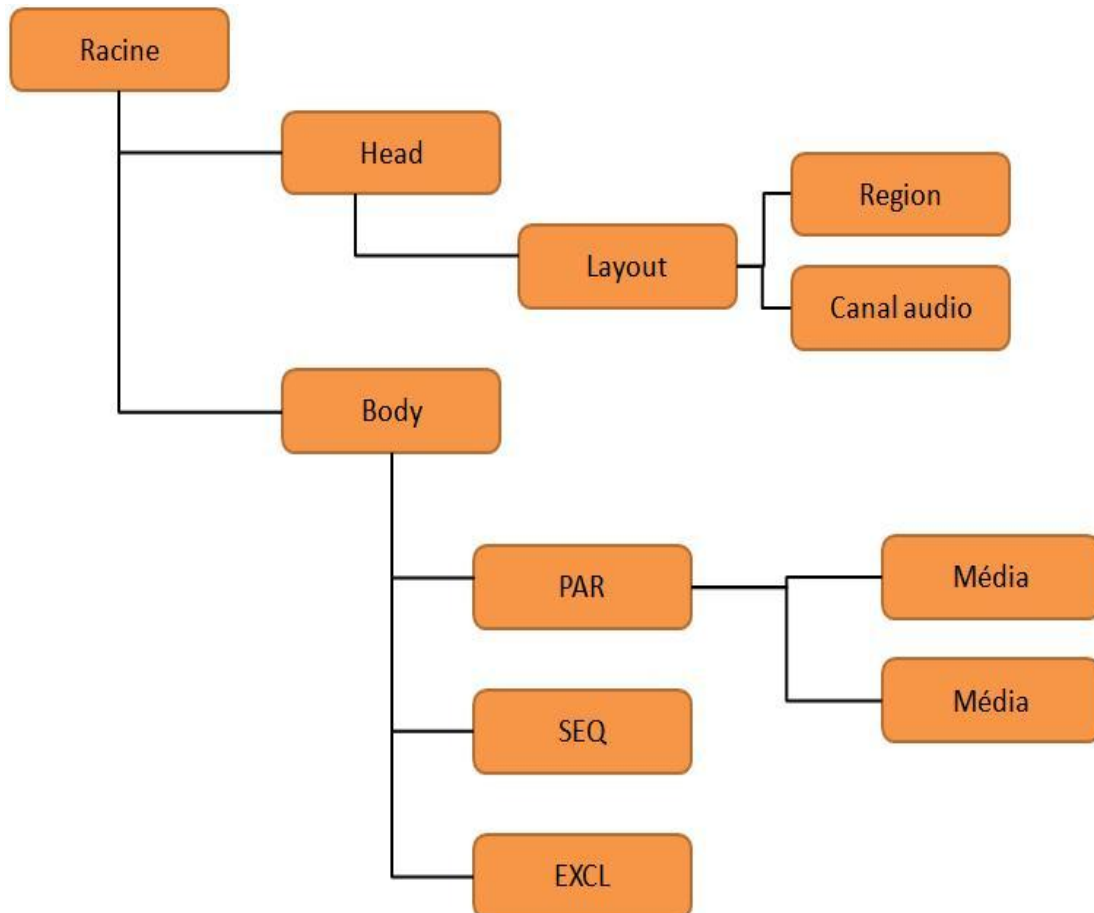
- **L'élément <smil/>** : agit comme l'élément racine, il marque le début et la fin de chaque document SMIL.
- **L'élément <head/>** : appelé aussi l'entête du document SMIL, contient les informations concernant l'aspect spatial de la présentation. Pour visualiser les objets multimédias, on doit définir une partie **<meta>** qui contient des informations descriptives, comme l'auteur ou la date de conception de la présentation, et une partie **<layout>** qui définit le positionnement et la mise en page de la fenêtre dans laquelle va se jouer la présentation.



## Chapitre III

- L'élément `<body/>` : ou le corps de la présentation, cet élément contient les informations sur l'aspect temporel. L'élément `<body/>` contient tous les conteneurs de temps et les liens existant dans une présentation SMIL.

La structure d'un document SMIL peut être schématisée par un arbre qui contient une racine et des branches, et des sous branches comme le montre la **Figure III.3** suivante :



**Figure III. 3** Structure d'un document SMIL

### III.5.2 Module Méta-Information

Le module **Méta-Information** (ou *métadonnées*), contient les éléments et les attributs qui permettent la description de documents SMIL. Les métadonnées sont des « *données sur des données* » ou, spécifiquement dans le contexte de cette spécification, des « *données décrivant des ressources web* » ; qui sont utilisés pour décrire les documents **SMIL** publiés sur le Web pour permettre une bonne lisibilité.

La spécification SMIL 3.0 permettait aux auteurs de décrire des documents avec un vocabulaire très basique en utilisant l'élément **meta**. Le module **Méta-Information** de SMIL 3.0,

## Chapitre III

gère pleinement l'utilisation de l'élément **meta** de SMIL 1.0, mais il introduit aussi de nouvelles fonctionnalités pour décrire des métadonnées en utilisant la syntaxe et le modèle du cadre de description de ressource **RDF** (*Resource Description Framework*), un langage de méta-information puissant pour fournir des informations sur des ressources.

### III.5.3 Module Layout

Le module de disposition de SMIL ou le module **Layout** est composé d'un module **BasicLayout** et de trois modules avec des fonctionnalités supplémentaires qui sont basées sur le module **BasicLayout**. Ces modules sont :

- Le module **AudioLayout** qui gère le contrôle du volume des médias auditifs grâce à l'attribut **soundLevel**,
- Le module **MultiWindowLayout** qui permet de définir plusieurs fenêtres de haut niveau superposées d'une façon hiérarchique par le biais de l'attribut **topLayout**,
- Et en fin le module **HierarchicalLayout** pour affiner la position du contenu dans une région par ces deux attributs **regPoint** et **regAlign**. La fonctionnalité dans ce module est essentiellement identique à la fonctionnalité de disposition dans SMIL 3.0.

Le module **BasicLayout** inclut un modèle de disposition pour organiser les **éléments médias** dans des régions sur la surface de rendu visuel. L'élément **layout** est utilisé dans l'en-tête **head** du document pour déclarer un **ensemble de régions** sur lesquelles les éléments médias sont rendus. L'élément **layout** admet deux fils :

- **Root-Layout** : détermine les valeurs des propriétés de disposition (dimensions + couleur de fond) de l'élément racine, qui détermine la taille de la fenêtre dans laquelle la présentation SMIL est rendue. Cet élément accepte les attributs : **id**, **height**, **width** et **backgroundColor**.
- **Region** : décrit les dimensions et la position de chaque zone où sera inséré l'élément média. Chaque région possède un ensemble de propriétés comme :

**Id** (identifiant), **top** et **left** pour la position par rapport au coin supérieur gauche de la fenêtre globale, **width** et **height** pour la largeur et hauteur (en pixels ou en % par rapport au root-layout), **z-index**, **fit** et **backgroundColor** (couleur de la fenêtre).

**Remarque** : Si un élément média n'a pas d'attributs explicites de région, alors une mise en page par défaut est attribuée. Si aucune mise en page **<root-layout>** n'est spécifiée, alors la taille de la fenêtre principale est calculée automatiquement de façon que cette dernière aura la taille du média le plus grand.

L'attribut **fit** spécifie le comportement du lecteur lorsque la définition de la région ne correspond pas à celle du média affiché. Cet attribut prend les valeurs suivantes :

- **Fill** : ajuste les dimensions de l'objet indépendamment de son contenu de telle sorte qu'il

## Chapitre III

remplisse tout l'espace de la région.

- **Hidden** : Si les dimensions de l'objet sont inférieures à celles de la région alors le média est affiché et le reste de l'espace inoccupé est rempli par la couleur de fond. Autrement, le média n'est pas affiché ou est tronqué. C'est la valeur par défaut.
- **Meet** : ajuste le média en préservant le rapport largeur-hauteur sans tronquer le contenu.
- **Slice** : ajuste le média en préservant le rapport largeur-hauteur en tronquant si nécessaire.
- **Scroll** : création d'ascenseurs en cas de dépassement.

L'attribut *z-index* détermine le niveau de l'empilement de la région dans son contexte d'empilement (l'indice le plus élevé sur le dessus).

Si deux éléments A et B ont la même profondeur alors:

- Si B débute après A alors B est placé au-dessus de A
- Si B et A débute en même temps et si l'élément B est après l'élément A dans l'arbre XML alors B est placé au-dessus de A.

### III.5.4 module Media Objects

Le module des objets médias définit la **syntaxe** des éléments et attributs nécessaires à la description des objets média que sont : *Animation, Audio, Img, Ref, Text, Textstream et Video*. Les éléments du module d'objet média permettent l'insertion d'objets média dans une présentation **SMIL**, ces objets média sont insérés par référence (via une URL). On distingue deux types d'objets média (cf. **Tableau III.1**) :

- **Média continu** : Fichier audio, fichier vidéo ou autre média pour lequel il existe une durée mesurable intrinsèque clairement établie.
- **Média discret** : Fichier image, fichier texte ou autre média n'ayant pas de durée intrinsèque.

Pour chaque élément multimédia, on indique la source du fichier et la région où il doit apparaître à l'aide des deux attributs "**src**" et "**region**", ainsi que tous les attributs de temporisation comme l'attribut "**dur**" qui détermine la durée d'activité d'un élément (cf. **Tableau III.2**).

L'attribut **src** permet de spécifier l'emplacement du média, qu'il se trouve dans le répertoire courant ou sur un serveur web distant.

L'attribut **region** permet de définir l'emplacement dans lequel le média va se jouer (cet emplacement doit avoir été créé préalablement).

**Tableau III.1.** Fonction et extensions des éléments médias

<b>Élément media</b>	<b>Fonction et extensions</b>
<animation />	Clips d'animation tel que les fichiers flashplayer (.swf)
< audio/>	Fichier son (.rm, .wav, .mov, .mp3, .mp4)
<img />	Image (.jpg, .gif, .png)
<ref />	Une référence sur un objet média (.rp)
< text/>	Référence textuelle (.txt)
<textstream />	Flux de texte (.rt)
<video />	Clips vidéo ou real vidéo (.rm, .avi, .mov, .mpeg)

**Tableau III.2.** Les attributs Media Objects

<b>Attribut</b>	<b>Fonction</b>	<b>Type de valeur</b>
Abstract	Décrit brièvement le contenu de l'élément.	Chaîne de caractère
Alt	Contient un texte de remplacement pour les médias qui ne peuvent pas être visualisés.	Chaîne de caractère
Author	Indique le nom de l'auteur de l'objet multimédia.	Chaîne de caractère
Copyright	Pour les droits de l'auteur.	Chaîne de caractère
Longdesc	Spécifier un lien (URI) vers une description détaillée de l'objet média.	Une adresse URI
Region	Pour indiquer l'identifiant de la surface d'affichage associé à l'objet	Un identifiant de region
Id	Il est unique pour chaque objet	Chaîne de caractère
Source	Donne la source URI de l'objet Multimédia	
Title	Donne le titre de média	Chaîne de caractère
Type	Indique le type MIME du média.	Chaîne de caractère
Begin	Définit l'événement de début de la présentation de l'objet.	Un délai ou un événement
End	Définit l'instant fin d'un événement	Un délai ou un événement
Dur	Indique la durée d'un objet.	Un délai ou la chaîne « indéfinie »
Repeat	Il est utilisé pour donner le nombre de fois qu'on veut jouer un objet.	Une valeur entière ou la chaîne « indéfinie »
Clip – begin & Clip-end	Utilisés pour le découpage temporel des objets continus, ils indiquent un décalage depuis l'instant de début et de fin d'objet.	Un délai
Fill	Utilisé pour la persistance d'un objet multimédia sur l'écran, il peut prendre les valeurs : <ul style="list-style-type: none"> <li>▪ <b>Freeze</b> : garder la dernière image sur l'écran après terminaison.</li> <li>▪ <b>Remove</b> : effacer l'objet de l'écran dès sa terminaison.</li> </ul>	Freeze ou remove
System-bitrate	Les attributs de test peuvent accompagner	
System-caption	Tout élément de synchronisation et qui évaluent les capacités et les paramétrages des systèmes.	
System-CPU		
System-language	Ils effectuent des tests booléens. Quand l'un des attributs de test spécifiés sur un élément est évalué à une valeur de « <b>false</b> », alors l'élément	

### III.5.5 Module de temporisation et de Synchronisation

La temporisation de SMIL fournit également des attributs permettant de spécifier le comportement d'un élément au cours du temps. Les éléments ont un début et une *durée simple*. La *durée simple* définit la durée de présentation de base d'un élément. Les éléments peuvent être définis pour répéter une durée simple, un certain nombre de fois ou encore pendant un certain temps. La durée simple et les effets de répétition sont combinés pour donner la *durée active*. Lorsque la durée active d'un élément est écoulée, celui-ci peut être soit retiré de la présentation, soit *gelé* (conservé dans son dernier état), par exemple pour combler un espace dans la présentation.

Le but principal des concepteurs du langage SMIL était de résoudre les problèmes de **synchronisation** des médias et définir un mécanisme puissant pour **ordonner** les contenus multimédias, ceci a été réalisé grâce à la première version qui offre un ensemble d'éléments et d'attributs.

- **Temporisation déterministe** : C'est la **temporisation** où les valeurs des attributs de temps (*begin*, *end*, *dur* ...), associés à un élément sont définis comme suit :
  - Soit en heures ("*xh*"), minutes ("*xm*"), secondes ("*xs*") ou encore en valeur d'horloge ("*hh:mm:ss*").
  - Soit par un évènement associé à un autre élément, tel que **id(Idf)(begin)**, **id(Idf)(end)** ou **id(Idf)(begin)+x** où *x* désigne une valeurs parmi celles décrites précédemment.
- **Temporisation non déterministe** : C'est une temporisation où les attributs de temps associés à un élément ne sont pas définis mais la valeur de ces attributs est déterminée par une certaine **activation extérieure**. L'activation peut reposer sur un évènement ou un hyperlien.

#### III.5.5.1 Les attributs de temporisation

SMIL 3.0 offre l'ensemble d'**attributs** existant dans SMIL2.0 et l'enrichit avec d'autres attributs. La temporisation de base pour un élément se fait grâce aux attributs : *begin* et *dur*, tandis que les attributs *end*, gèrent le contrôle de la durée active.

**L'attribut begin** : Cet attribut sert à lancer l'objet multimédia à un moment précis de l'axe du temps de la présentation. Par conséquent, si l'attribut *begin* est utilisé dans un groupe, le temps qui lui sera affecté au début du groupe. La valeur de cet attribut peut être déterministe ou non et peut être positive, négative ou nulle.

**L'attribut dur** : Il spécifie la durée simple d'un élément qui définit la durée de présentation

## Chapitre III

de base d'un élément. Combiner avec ses effets de répétitions, elle nous donne une durée **active**. Si un élément n'a pas d'attribut dur (valide), la durée simple de cet élément est définie comme étant sa durée implicite. La durée implicite dépend du type de l'élément.

**L'attribut end** : Permet à l'auteur de contraindre la durée active en spécifiant une valeur de fin via un décalage simple. La valeur de cette durée peut être exprimée relativement à un temps donné, par rapport à un autre élément ou un événement précis.

**L'attribut repeatDur** : Indique la durée totale de la répétition d'un élément média ou conteneur. Cet attribut fournit une sémantique qui correspond mieux aux cas d'utilisation typiques et offrent un meilleur contrôle de la durée des répétitions.

**L'attribut Min** : permette à l'auteur de spécifie la valeur minimum de la durée active.

**L'attribut Max** : permette à l'auteur de spécifie la valeur maximum de la durée active.

### III.5.5.2 Les éléments de synchronisation

Appelés aussi conteneurs de temps, ils acceptent comme fils les éléments média. Ces éléments sont :

L'élément **<seq>** joue les éléments enfants les uns à la suite des autres en séquence.

L'élément **<excl>** joue les éléments enfants un à la fois mais sans imposer l'ordre.

L'élément **<par>** joue les éléments enfants comme un groupe permettant une lecture en parallèle.

L'élément **<par/>** : Un conteneur de temps **<par/>**, abréviation de « parallèle », permet de lancer plusieurs éléments en même temps. Le début des éléments fils d'un conteneur **<par/>** est par défaut son début. La durée implicite de **<par/>** se termine avec la dernière fin active des éléments enfants. Cette durée peut être contrôlée par l'attribut **endsync**.

Exemple : L'exemple de la **Figure III.4** permet à trois images de s'afficher en parallèle.

```
<par>
  
  
  
</par>
```

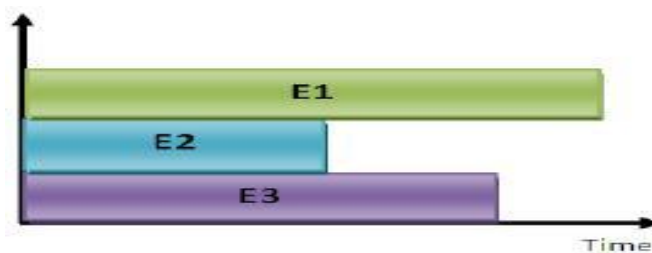


Figure III. 4 Exemple d'éléments joués en parallèle

L'attribut **endsync** : L'attribut **endsync** contrôle la durée implicite des conteneurs de temps en fonction des enfants. Les valeurs acceptées par l'attribut **endsync** sont présentées dans le **tableau III.3** suivant :

Tableau III.3: Les valeurs acceptées par l'attribut **endsync**

Valeur	Fonction
First	La durée implicite du conteneur ou de l'élément se termine avec la fin active la plus courte d'un élément enfant.
Last	La durée implicite du conteneur ou de l'élément média se termine en même temps que la dernière fin active d'un enfant.
All	La durée implicite du conteneur ou de l'élément se termine quand tous les éléments enfants ont terminé leurs durées actives respectives.
Media	La durée implicite du conteneur de temps se termine quand la durée intrinsèque du média se termine.
Id (valeur)	La durée implicite du conteneur ou de l'élément se termine quand l'enfant spécifié par <b>id</b> termine sa première durée active.

L'élément **<seq/>** : Un conteneur **<seq/>** définit une séquence d'éléments dans laquelle les éléments sont joués les uns après les autres. L'élément **<seq/>** lui-même gère tous les éléments de temporisation sauf **endsync**. La durée implicite d'un élément **<seq/>** se termine avec la fin active du dernier enfant du conteneur **<seq/>**.

L'exemple de la **Figure III.5** suivant permet d'afficher séquentiellement trois éléments média.

```

<seq>
  <text id="E1" src="txt.rt"/>
  <video id="E2" src="video1.mpeg"/>
  
</seq>

```



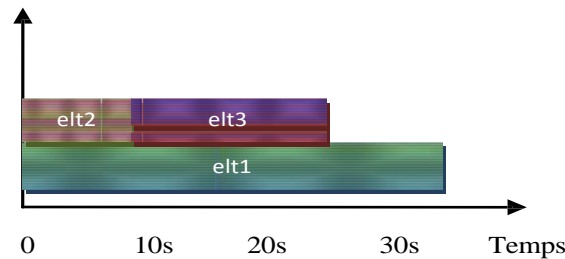
Figure III. 5 Exemple d'éléments joués en séquence

**Remarque :** Si au moins un enfant d'un **<seq/>** a une durée active indéfinie, alors la durée implicite du **<seq/>** est aussi indéfinie. Il est possible d'imbriquer les modes de lecture séquentielle et parallèle, ce qui nous ramène à une lecture combinée. L'exemple de la **Figure III.6** lance parallèlement la lecture de deux groupes d'éléments multimédia. Le premier groupe ne comprend

## Chapitre III

qu'un élément, alors que le deuxième en comprend deux qui sont lus séquentiellement.

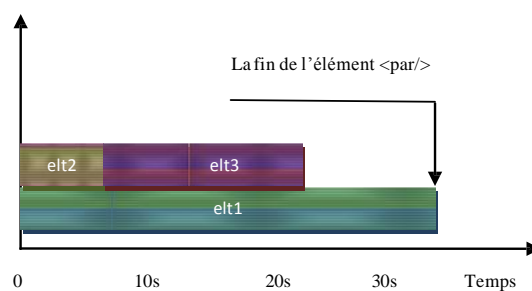
```
<par>
<audio id="elt1" src="chanson.mp3"/>
  <seq>
    <video id="elt2" src="video1.rm" region="r1"/>
    <video id="elt3" src="video2.rm" region="r1"/>
  </seq>
</par>
```



**Figure III.6** Exemple d'éléments jouées en parallèle et en séquence

Dans l'exemple de la **Figure III.7**, la fin du conteneur "par" se termine au moment où le média "elt1" termine sa présentation.

```
<par endsync="elt1">
  <audio id="elt1" src="chanson.mp3"/>
  <seq id="sequence">
    <video id="elt2" src="video1.rm" region="r1"/>
    <video id="elt3" src="video2.rm" region="r1"/>
  </seq>
</par>
```



**Figure III.7** Exemple de synchronisation en enchaînement

### III.6 Le module Linking

Le module de lien, définit des éléments et attributs du document **SMIL 3.0** pour la **navigation hyperlien**. Ce sont des liens qui peuvent être déclenchés par interaction de l'utilisateur ou par d'autres événements déclencheurs, tels que des **événements temporels**. On distingue deux éléments :

**A- L'élément <a/>** : La fonctionnalité de l'élément <a/> est similaire à celle de l'élément <a/>



## Chapitre III

dans HTML. Deux attributs principaux :

- **href** : donne l'URL où se trouve le fichier correspondant à l'objet média (la cible du lien)
- **show** : qui contrôle le comportement du document source contenant le lien une fois que l'interaction se produit. Il peut prendre l'une des valeurs suivantes :
  - ✓ **"replace"**: La présentation courante est suspendue et remplacée par la cible. C'est la valeur par défaut.
  - ✓ **"new"**: La présentation cible commence dans un nouveau contexte sans affecter la présentation source.
  - ✓ **"pause"**: La présentation source est mise en pause à son état courant, et la cible commence dans un nouveau contexte. Lorsque la présentation cible termine, la source est reprise à partir de l'état de sa mise en pause.

**Remarque** : Les liens peuvent se faire sur tous les éléments multimédia ainsi que sur les éléments `<seq>`, `<par>` et `<excl >`.

**Exemple** : Lien d'un élément vers une présentation externe.

```
<a href=http://www.cwi.nl/video.smi show="replace">
<video src="movie1.mov" region="main"/>
</a>
```

**B- L'élément `<area/>`** : La sémantique de l'élément `<area/>` dans SMIL 3.0 est la même que celle dans HTML du fait qu'elle peut spécifier qu'une partie spatiale d'un objet visuel peut être choisie pour déclencher l'apparition de la destination du lien. C'est-à-dire, elle rend la partie spatiale ou temporelle de ces objets cliquable.

Pour déterminer spatialement quelle partie de l'objet média est un lien, on utilise l'attribut *coords* qui permet de définir la zone rectangulaire cliquable à l'aide de quatre valeurs (en pixels ou en pourcentage) séparées par des virgules. Les deux premières valeurs donnent les coordonnées du coin supérieur gauche du rectangle, les deux dernières valeurs donnent les coordonnées du coin inférieur droit du rectangle.

Les zones cliquables d'un objet média peuvent être déterminées également temporellement.

Il est également possible de combiner une détermination temporelle et spatiale d'une partie cliquable. L'exemple de la **Figure III.8** ci-dessous montre une partie cliquable (après 4s et avant 8s)

```
<video src="video1.rm" region="R1" >
< area id="anc" href="fleur.smil" begin="4s" end="8s"
  coords = "left-x , top-y ,right-x ,bottom-y" />
</video>
```

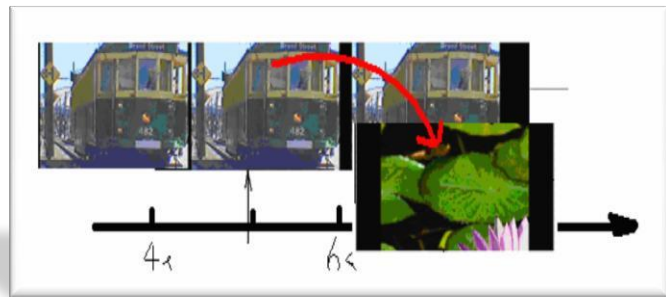
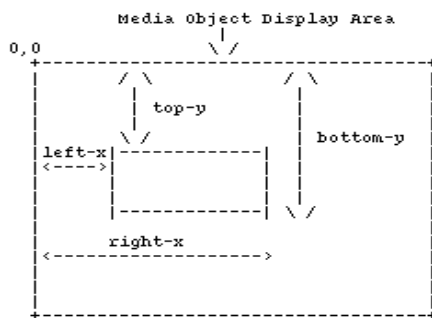


Figure III.8 Exemple d'une partie temporelle cliquable

### III.7 Le module Content Control

Le module de contrôle de contenu de SMIL 3.0 contient les éléments et les attributs qui offrent des choix de contenu à l'exécution et une livraison de contenu optimisée. La fonctionnalité de contrôle de contenu de SMIL3.0 se partage en quatre modules :

- **Le module BasicContentControl** : Contient les **attributs de test** du système définis dans SMIL2.0. SMIL3.0 inclut la fonctionnalité de l'attribut de test de SMIL 3.0 et l'étend en reconnaissant de nouveaux attributs de test qui définissent des caractéristiques additionnelles à l'environnement système tels que : systemAudioDesc et systemCPU qui peuvent être associés à un média dans l'élément **<switch/>**.
- **L'élément <switch/>** : Permet à un auteur de spécifier un ensemble d'éléments alternatifs dont seul le premier élément acceptable est choisi. Les attributs de test de l'élément **<switch/>** sont présentés dans le **Tableau III.4** suivant :

Tableau III.4 Les attributs de test de <switch/>

Attribut	Valeur	Fonction
SystemLanguage	Liste de noms de langues (Fr,	Effectue un test sur la langue configurée par
SystemBitrate	Bites/seconds	Choix du type de connexion, contrôle le débit de transmission.
SystemCaptions	On ou off	Ignoré si l'utilisateur ne veut pas de sous-titre.
SystemRequired	Nom de l'extension	Reconnaissance ou non d'une extension.
SystemScreenSize	Taille: hauteur x largeur (en	Spécifie la taille minimale de l'écran.
SystemScreenDepth	Nombre de bits de codage 1, 4, 8, 16,	Spécifie la résolution minimale.
SystemOverdupOrCaption	<b>Caption</b> ou <b>overdub</b>	Sous-titrage ou non.

## Chapitre III

L'exemple suivant nous montre une illustration d'utilisation de l'élément <switch/>.

**Exemple** : choisir entre des ressources audio dans différentes langues.

```
<switch>  <audio src="audio-french"      systemLanguage="fr"/>
          <audio src="audio-english"    systemLanguage="en"/>
          <audio src="audio-arabic"/>
</switch>
```

Si l'élément **switch** est utilisé comme enfant direct ou indirect d'un élément **body**, il peut contenir n'importe quel objet média ou conteneur de structure de temporisation (seq, par,), ou il peut contenir des éléments **switch** imbriqués. Tous ces éléments peuvent apparaître plusieurs fois au sein du **switch**. Si le **switch** est utilisé comme enfant direct ou indirect d'un élément **head**, il peut contenir un ou plusieurs éléments **layout** .

### III.8 Le module Animation

Les modules d'animation de SMIL 3.0, sont composés d'un module *BasicAnimation* et d'un module *SplineAnimation*. Ces modules contiennent des éléments et des attributs pour intégrer une animation sur un plan de montage chronologique, et un mécanisme pour composer les effets de plusieurs animations. Puisque ces éléments et ces attributs sont définis dans des modules, les concepteurs d'autres langages balisés peuvent choisir d'inclure ou non cette fonctionnalité dans leurs langages. Les concepteurs de langage qui intègrent d'autres modules SMIL n'ont pas besoin d'inclure les modules d'animation si la fonctionnalité de l'animation n'est pas utile.

### III.9 Le module Transitions

Les modules de transition de SMIL 3.0 contiennent les éléments et les attributs qui offrent la fonctionnalité de transition à SMIL 3.0. Cette catégorie est donnée à travers quatre modules :

- le module *BasicTransitions*, qui définit la méthode de sténographie style-like de spécification des transitions,
- le module *FullScreen*, qui définit des mécanismes pour les transitions qui anime l'écran entier,
- le module *InlineTransitions*, ce module permet un niveau beaucoup plus fin de contrôle des transitions, et en fin
- le module *TransitionModifiers*, qui fournit le contrôle additionnel de l'aspect visuel d'une transition.

## Chapitre III

### III.10 Analyse des document SMIL

De nos jours, la notion de DMM interactifs est devenue de plus en plus répandue dans différents domaines d'applications tels que : l'éducation, la médecine, etc. du fait que ces documents peuvent être distribués et accessibles via le Web. Dans ce contexte, le langage SMIL du W3C [1] a été proposé pour la présentation des DMM interactifs sur le Web.

Ces documents sont caractérisés par l'intégration et la synchronisation de plusieurs médias (son, vidéo, image,...). De nombreux travaux se sont penchés sur la synchronisation multimédia, et ont abouti à plusieurs standards internationaux de multimédia. La plupart de ces standards considèrent plutôt la synchronisation basse niveau des flux multimédia. En revanche, SMIL qui a été retenu comme standard par le W3C permet de spécifier des synchronisations entre objets multimédia plus élaborées.

Actuellement, plusieurs versions du langage existent dont SMIL 1.0 [1], SMIL 2.0 [2] et SMIL 3.0 [4].

Notre travail concerne le langage SMIL 3.0 [4] qui est une amélioration de SMIL 2.0 de telle sorte que la présentation devienne plus dynamique et flexible (par exemple, la présentation d'animations, la gestion d'évènements, la gestion de désynchronisation parmi les objets média, ...). Les fonctionnalités de SMIL 3.0 ont été définies en utilisant des modules qui permettent à un utilisateur de décrire son application selon le niveau de complexité voulu. Les modules SMIL 3.0 sont : Structure, Media, Content Control, Linking, Timing and Synchronization, Animation, Time Manipulations, Transition Effects et Metainformation.

#### III.10.1 Structuration

La structuration d'un DMM est importante car celui-ci n'est pas uniquement une suite de slides avec quelques animations. Il est en fait composé de différents média (textes, sons, images, vidéo, animation...) présentés dans des régions, chacun de ces média pouvant, à un moment ou à un autre avoir une interaction plus ou moins grande avec toute autre partie du document. Cette interaction se concrétise par des id's et des références à ces id's. Cela implique une nécessaire structuration du document permettant d'éviter les redondances de code. Le document SMIL est donc composé de plusieurs parties, l'entête comprenant notamment la description des régions et des animations, le corps du document regroupant le scénario proprement dit. La richesse d'un document SMIL dépend aussi de la capacité du langage, comme la plupart des langages dérivés d'XML d'inclure la notion d'espaces de nom (namespaces) permettant d'inclure, dans un document SMIL d'autres langages (SVG, formules mathématiques, documents html).

#### III.10.2 Navigation

A la navigation interne à un DMM et aux différents détails qui le composent s'ajoutent des

## Chapitre III

besoins de navigation entre documents sur des thématiques particulières. Le langage doit ainsi prendre en compte non seulement la notion hypertexte liée au lien proprement dit, mais aussi la donnée temporelle, tant à la source (le lien peut n'exister que pendant un temps défini dans une vidéo par exemple), qu'à la destination (la cible du lien est un point précis temporel du document de destination avec l'ensemble des autres média synchronisés sur ce point). SMIL3.0 permet aussi de prendre en compte la notion d'évènement, notamment pour les attributs begin et end.

### III.10.3 Placement temporel et synchronisation

La gestion des différents média temporels (animations, vidéos, sons) nécessite une réflexion approfondie, particulièrement sur la granularité de leur synchronisation. En effet, s'il est relativement simple de faire apparaître un média dans un document, il est beaucoup plus complexe de le synchroniser finement avec les autres éléments du document. On distinguera 3 types de synchronisation :

- La synchronisation « gros grain » gérée par les éléments <par> et <seq> et <excl> permettant de définir le placement dans le temps des différents média.
- La synchronisation liée à l'utilisation des attributs de synchronisation begin ou end. Ceux-ci permettent d'exprimer le début d'un élément de plusieurs façons (temps d'horloge ou temps où se produit un évènement).
- Ce type de synchronisation peut encore s'affiner par l'utilisation par exemple :
  - de l'attribut clipBegin qui permet de commencer un média, non pas au début, mais à un instant décalé par rapport à son début.
  - de l'élément area permettant de séparer un objet en sous-parties temporelles, en utilisant des attributs comme begin et end. Les valeurs des attributs begin et end étant relatives au début de l'objet média conteneur. L'élément area permet de faire d'une sous-partie de l'objet média la destination d'un lien, en utilisant ces attributs temporels et l'attribut id.

La manipulation directe du langage SMIL est trop complexe pour la plupart d'utilisateurs car ces tâches demandent des bonnes connaissances de la sémantique du langage. En effet, un auteur doit comprendre la structure temporelle et les sémantiques d'évènement pour créer des relations temporelles entre des objets médias.

Le standard SMIL, tel qu'il a été défini par le W3C [1], est un langage de description de DMM synchronisés. L'aspect temporel est donc un point crucial pour de telles présentations. Ce langage permet aussi d'exprimer d'autres informations: des informations spatiales, des informations hypermédias, et des Meta informations. Dans le cadre de l'hypermédia, l'apport majeur du langage SMIL est l'introduction de la notion de lien temporel

## Chapitre III

La flexibilité du modèle temporel de SMIL 3.0 permet aux auteurs de décrire les relations de synchronisation temporelles qui ne peuvent potentiellement être résolues durant la présentation du document, et qui sont connues comme des incohérences ou inconsistances temporelles.

### III.11 Cohérence des documents SMIL

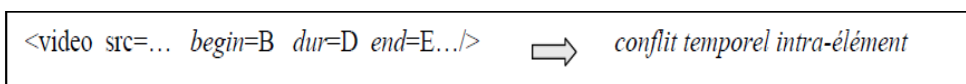
Une incohérence temporelle est la conséquence de la définition de relations de synchronisation temporelle qui ne peuvent être satisfaites simultanément au moment de la présentation. Les recherches menées en [Yang 00] ont montré que les incohérences temporelles en SMIL peuvent être définies comme des situations de conflit entre les valeurs d'attributs temporels associés aux éléments de la présentation. Deux situations de conflit temporel peuvent se présenter dans un document SMIL [Yang 00]:

- a. Le conflit temporel *intra-élément*,
- b. Le conflit temporel *inter-éléments*.

#### III.11.1 Conflit temporel intra-élément

Le conflit temporel intra-élément correspond à une situation d'incohérence temporelle au sein d'un même média. Cette situation est due à un conflit entre les valeurs des attributs temporels associés au même média. Pour détecter ce cas de conflit, il suffit de vérifier les attributs temporels associés à chaque média de la présentation. Par exemple, considérons le code SMIL de la

**Figure III.9** : les valeurs d'attributs temporels *begin*, *end* et *dur* associés au média vidéo sont respectivement: B, D et E secondes. Une situation de conflit temporel sera détectée dès que  $B+D \neq E$ .



**Figure III.9** Conflit temporel intra-élément

Le **Tableau III.5** suivant donne la liste des combinaisons possibles des valeurs d'attributs. Nous n'avons considéré que les valeurs absolues des attributs *begin* et *end*. La mention NS signifie que la valeur de l'attribut n'est pas spécifiée.

**Tableau III.5** Calcul de la durée active et situations de conflit intra-élément

Begin	End	Dur	durée active calculée
<b>B</b>	D	D	Conflit si $B \neq 0$
<b>B</b>	E	NS	$E - B$ sinon conflit
<b>B</b>	NS	D	D
<b>B</b>	NS	NS	Durée média
NS	E	D	Conflit
NS	E	NS	E
NS	NS	D	D
NS	NS	NS	Durée média

## Chapitre III

### III.11.2 Conflit temporel inter-éléments

Le conflit temporel inter-éléments correspond à une situation d'incohérence temporelle entre deux ou plusieurs médias différents d'une présentation. Ce cas de conflit est plus difficile à détecter car il met en jeu plusieurs médias à la fois. En effet, la détection d'une situation de conflit inter-éléments nécessite l'analyse de toutes les relations de synchronisation entre les médias de la présentation.

Dans l'exemple de la **Figure III. 10**, les durées des éléments audio (A2) et texte (Txt1) sont en conflit avec la durée de l'élément composite **par**: on ne sait pas si l'on doit arrêter les objets fils après 20 secondes du début du bloc, ou bien si l'on doit les laisser s'exécuter jusqu'à la fin, en ignorant ainsi la valeur de durée spécifiée pour l'élément **par**.

```

< par dur="20s" >
  < seq >
    < audio src=A1 dur="15s" />
    < audio src=A2 dur="10s" />
  </ seq>
  < img src=Img1 dur="10s" />
  < text src=Txt1 dur="30s" />
</ par>

```

**Figure III. 10** Conflit temporel inter-éléments

Un autre exemple qui montre une incohérence temporelle inter-elements est résumé par la présentation suivante :

```

<video id = "vid" region = "video" src = "vid.rm" begin = "id (aud)(end)" repeat = "1" />
<audio id = "aud" src = "aud.wav" begin = " id (vid)(end) " repeat = "1" />

```

On voit bien que l'incohérence temporelle est due au fait que le début d'exécution de *vid* est relié à la fin d'exécution de *aud*, ou que le début d'exécution de *aud* est relié aussi à la fin d'exécution de *vid*, d'où une situation d'interblocage.

### III.12 Inconsistances des documents SMIL

Les DMM apparaissent comme une nouvelle génération d'applications informatiques dont la conception à grande échelle met en jeu leur consistance. Cette dernière survient lors de la présentation d'un document dont la *conception est incohérente*. Cette incohérence provoque un comportement inattendu des médias. En effet, la spécification de tels documents, en particulier celle de SMIL, nécessite d'être totalement prise en charge afin de vérifier sa consistance. L'inconsistance d'une présentation SMIL peut être due à :

**Une incohérence des contraintes intrinsèques du document:** Le document SMIL doit

## Chapitre III

forcément être validé dès son édition. Certains éditeurs SMIL tels que GRINS offrent un module de validation de la consistance, permettant de vérifier si les contraintes temporelles imposées dans le document peuvent être satisfaites conjointement, dans un environnement centralisé.

**Exemple :** Cet exemple ci-dessous montre une incohérence lors d'exécution, car le début d'exécution de vidéo est relié à la fin d'exécution de audio, or le début d'exécution de audio est relié aussi à la fin d'exécution de vidéo, d'où une situation d'**interblocage**.

```
<video id = "vid" region = "video" src = "vid.rm »
      begin = "id(aud)(end)" repeat = "1" />
<audio id = "aud" src = "aud.wav"
      begin = " id(vid)(end) " repeat = "1" />
```

- **Une incohérence due aux perturbations du réseau :** Un document SMIL intrinsèquement consistant, pourrait produire une présentation inconsistante, lorsque les objets requis sont localisés sur un réseau. Les dispersions des objets médias sur Internet, ainsi que la variation des délais d'acheminement, génèrent des latences plus ou moins importantes et dont les effets peuvent être irréversibles. Les retards induits provoquent la non satisfaction des contraintes temporelles et causant ainsi une présentation SMIL incomplète.
- **Cas d'un média qui n'est jamais joué:** un certain décalage entre le début et la fin de certains média peut empêcher la lecture d'un média :

**Exemple 1:** soit le code SMIL suivant :

```
<par endsync="first" id="par1">
  <textstreamsrc="coeurtext.rt" begin="0s" dur="10s" id="textstream" />
  <video src="VideoCoeur.rv" begin="0s" dur="11s" id="video"/>
  <audio src="muzic.mp3" dur="11s" id="audio" />
  <imgsrc="TransverseImage2.gif" begin="11s" dur="10s" id="video2"/>
</par>
```

L'exemple1 montre une inconsistance lors de l'exécution des média : TransverseImage2 et Coeurtext. En effet, le temps de début de lecture du média TransverseImage2 est supérieur au temps de fin de lecture du média Coeurtext (la fin de la lecture est synchronisée avec la fin de la lecture du média Coeurtext).

**L'exemple 2:** soit le code SMIL suivant :

```
<par endsync="AudioCoeur" id="par1">
  <textstream src="Coeurtext.rt" begin="0s" dur="10s" id="textstream" />
  <video src="VideoCoeur.rv" begin="0s" dur="10s" id="video"/>
  <audio src="AudioCoeur.mp3" dur="10s" id="audio" />
   </par>
```



## Chapitre III

L'exemple 2 montre aussi un cas d'inconsistance lors de la lecture en parallèle des quatre média : Coeurtext, VideoCoeur, AudioCoeur et TransverseImage2, dont la fin de lecture est synchronisée avec la fin de lecture du média AudioCoeur. En effet, le temps de début de lecture du média TransverseImage2 est supérieur au temps de fin de lecture du média AudioCoeur. Donc, on peut remarquer que dans les deux exemples que le média TransverseImage2 n'est jamais joué.

En conclusion, nous pouvons dire que la complexité de la problématique de la synchronisation dans les systèmes multimédias est telle qu'il est nécessaire de développer des techniques de modélisation formelle adaptées aux besoins de telles applications. Un modèle formel pour les présentations multimédias devrait permettre de représenter tous les mécanismes de synchronisation possibles de la présentation, et doit proposer un mécanisme de validation des scénarios temporels ainsi modélisés. Un modèle formel pour les documents SMIL doit répondre, en plus des besoins classiques des systèmes multimédias, aux besoins spécifiques issus des comportements temporels propres à ce langage [Samp 03].

Pour résoudre le problème de cohérence temporelle et d'inconsistance des documents SMIL, une approche de modélisation des contraintes de synchronisation temporelle, spatiale et logique et de validation des incohérences est plus que nécessaire.

### III.13 Etat de l'art des approches de validation des documents SMIL

Il est clair que la cohérence temporelle d'un document SMIL a un impact direct sur la qualité de sa présentation et par conséquent sur la satisfaction de son auteur/utilisateur.

Pour créer alors des présentations multimédia en SMIL, l'auteur du DMM veut tout d'abord s'assurer de la validité du script SMIL au niveau syntaxique et sémantique. Les erreurs syntaxiques du script SMIL associé à une présentation multimédia pourrait être facilement vérifiées et corrigées à l'aide d'un éditeur dédié comme LimSee 2 [7], tandis que les erreurs sémantiques nécessitent des modèles formels de spécification et de validation du code SMIL. Un modèle formel doit vérifier les erreurs sémantiques du langage SMIL qui sont relatives aux valeurs d'attributs associés à ses éléments. Pour cela, plusieurs approches et modèles de formalisation et de validation de cohérence temporelle et de consistance d'un document SMIL existent dans la littérature et qui sont basés sur les réseaux de Petri, la logique temporelle ou les automates temporels. Nous exposons dans cette section quelques modèles et approches les plus connues dans la littérature :

**Little et Ghafor. 1990** [Litt 90], proposent un modèle modifié du réseau de Petri, appelé Composition Object Petri Net (OCPN) permettant de spécifier les relations de composition temporelle binaires ou n-aires entre les médias. Toutefois, le modèle OCPN ne traite pas les contraintes de synchronisation plus complexes en temps réel et donc il n'est pas assez puissant pour capturer toute la sémantique d'ordonnancement et de synchronisation des médias composant un document SMIL.

### Chapitre III

Dans l'approche de **Jourdan et al.** [Jour 01], présente la première tentative de définir une sémantique formelle pour SMIL. Cette approche est basée sur l'utilisation d'automates temporisés et a été utilisée lors de la conception de SMIL 2.0 pour améliorer la spécification. Le présent document se concentre principalement sur SMIL 1.0 et prend en considération que deux nouvelles fonctionnalités de SMIL 2.0. Malheureusement, le formalisme adopté ne semble pas être évolutive afin de couvrir toutes les fonctionnalités de la troisième version de la norme et ne permet pas à l'utilisateur de corriger les erreurs.

Le modèle RTSM (Real Time Synchronization Model) de **Yang 2000** [Yang 00] ainsi proposé, est également basée sur le modèle OCPN et définit deux types de lieux et de nouvelles règles de tir sont proposées. Le modèle RTSM peut capturer la sémantique temporelle d'un document SMIL et détecte les conflits temporels des médias. Toutefois, la vérification n'est pas incrémentale et nécessite l'analyse de l'ensemble du RTSM. De plus, la traduction du modèle SMIL vers RTSM génère une perte de la structure temporelle de SMIL.

**Chung et Anil.** [Chun 03] proposent un nouveau modèle qui est une amélioration de l'OCPN en ajoutant des jetons typés et un nouvel ensemble de règles de tir. Ce modèle permet de capturer la sémantique de cadencement et de synchronisation de SMIL. Toutefois, aucune des techniques de vérification n'est proposée pour vérifier la cohérence temporelle de la spécification SMIL.

**Sampaio et Courtiat.** [Samp 04] décrivent le modèle RT-LOTOS : une description formelle des éléments SMIL basée sur les automates temporels, ce qui permet la génération d'un ordonnancement valide. Le modèle RT-LOTOS prend en considération le problème de la qualité de service, mais les auteurs ne définissent pas une sémantique pour le langage SMIL. Ils comparent les comportements de différentes lectures des médias qui sont encore dépendants de l'implémentation. Cette représentation formelle des documents SMIL permet de détecter les incohérences temporelles, mais elle ne permet pas à l'auteur de les corriger. En outre, l'approche n'est pas extensible: l'automate qui décrit le comportement obtenu doit être reconstruit du début après les modifications.

L'approche de **Maazouz et al.** [Maaz 06] permet de trouver un ensemble plus complet d'inconsistances ainsi que la présence de chemins non accessibles du graphe d'accessibilité associé au TPN. Ces informations sont utiles pour corriger les erreurs dans les documents SMIL, mais le système ne permet pas à l'auteur de localiser les erreurs. En outre, l'approche souffre du problème d'explosion des états du graphe d'accessibilité d'où le temps de calcul qui augmente de façon exponentielle avec la taille du document.

**Bossi et Gaggi.** [Boss 07] proposent un sous ensemble de sémantique formelle pour la vérification des documents SMIL basé sur la logique d'Hoare. Les temporisateurs : *begin*, *and* et *dur* sont

## Chapitre III

évalués par des axiomes, tandis que pour les synchronisateurs : *par*, *seq* et *excl*, des règles plus complexes sont alors nécessaires. Malheureusement, cette approche offre une sémantique formelle pour la partie temporelle seulement mais elle ne prend pas en considération la vérification spatiale des documents SMIL.

**Bouyakoub et Belkhir** [Bouy 07] proposent une extension temporelle des réseaux de Petri (SMIL-Net en abrégé) pour la vérification des aspects temporels et hyper-temporelle des documents SMIL. À la fin du processus d'édition, la spécification temporelle de SMIL est traduite en SMIL-Net et des techniques de vérification peuvent être appliquées pour vérifier la cohérence temporelle directement sur le modèle.

Malheureusement, cette approche se concentre seulement sur la vérification temporelle. Bien que le conflit spatial soit un problème majeur qui affecte la qualité des présentations multimédias, cette approche ne traite pas la vérification spatiale des documents SMIL.

### III.14 Etude comparative des différentes approches de validation de SMIL

Les éditeurs visuel de SMIL corriger les erreurs syntaxiques mais ne corrige pas les erreurs sémantiques comme les conflits temporels dans les contraintes de synchronisation, Ou les conflits pendant la lecture des documents par exemple les interactions user (link, pause, replay,...), donc la vérification des documents SMIL est une tâche complexe nécessite des outils de vérification automatique base sur un sémantique formelle qui Controller l'aspect temporelle des éléments SMIL.

Jourdan et al. [Jour 01], fait une tentative de définir une sémantique pour SMIL basée sur les automates temporels mais leur méthode ne définit pas une trace d'erreur et une proposition pour corriger l'erreur. Yang 2000 [Yang 00] développer un logiciel appeler SAM base sur les réseaux de pétri pour Vérifier la satisfiabilité au Qualité de Service réseaux, il Transformer les documents SMIL a des réseaux de pétri puis générer des axiomes pour prouve des formules logiques (exprimer QoS) mais le logiciel SAM ne détecter pas les conflits sémantiques.

Sampaio et Courtiat. [Samp 04] propose une approche basée sur les automates pour la génération automatique des spécifications RT-LOTOS de documents SMIL 2.0, il réussit de Générer un ordonnancement valide pour la présentation avec la considération de QoS et peut trouver les erreurs mais ne pointer pas sur l'erreur en plus de ça l'approche est non extensible : l'automate qui décrit le comportement doit être reconstruit après chaque changement. Maazouz et al [Maaz 06] utiliser les réseaux de pétri pour décrire l'évolution temporelle des présentations SMIL par la transformation des éléments SMIL a des places et transitions, l'avantage majeure il détecter les objets non accessible plus des informations utile pour corriger les erreurs avec un inconvénient c'est le problème de l'explosion combinatoire qui augmente le temps de calcule.

**Chapitre III**

Bossi et Gaggi. [Boss 07] enrichir SMIL par les assertions qui exprimer les propriétés temporelle,

Les assertions écrites sous forme des triplets de Hoare qui compose la sémantique pour l'évaluation des objets et relations dans une structure complexe, Cette sémantique utilisée des règles d'inférence. Aucune transformation sur les documents SMIL, donc pas de perte sémantique mais les limites de cette approche :

- Pas de définition des intervalles valide pour les interactions utilisateur (événement)
- Pas de génération automatique des scenarios.

A cause de l'utilisation d'un moteur d'inférence sous forme API native pour lancer la preuve de Hoare, l'API ne peut pas faire des calculs supplémentaire et spécifiques l'hors la construction de preuve.

**Notre approche** basé sur une sémantique formelle définir les aspects temporels d'éléments SMIL au moyen d'un ensemble des règles d'inférence, Les règles dans l'esprit de la sémantique de Hoare, comme Bossi & Gaggi mais avec une implémentation **des règles et un moteur d'inférence purement en java (voir Table III.7)**, cette approche permet au moteur d'inférence de calculer les intervalle de temps valides dans lequel les interactions de l'utilisateur doivent se produire. En outre, il est capable de générer automatiquement des séquences d'événements pour tester les documents SMIL grâce à un module appelé générateur automatique des scenarios.

En plus de ça nous étendre l'ensemble des attributs couverts par des nouveaux attributs comme : **MIN, MAX, RepeatDur**, pour offrir a l'auteur un plus de contrôle sur la durée active des objets SMIL.

**Tableau III.6** Comparaison des différentes approches de validation de SMIL

Auteur	Vérification temporelle	Formalisme	Transformation	Aide de correction	Extensibility
Little	Y	PN+ITL	O	NI	N
Yang	Y(SMIL1)	PN	O	O	N
Jourdan	Y(SMIL1)	TA	O	N	N
Yu	N(SMIL1)	PN+TL	O	N	N
Newman	N(SMIL2)	PA	O	NI	N
Chung	N	PN	O	NI	N
Mazzouz	Y(SMIL2)	PN	O	P	N
Sampaio	Y	TA	O	N	N
Bossi	Y(SMIL2.1)	A	N	O	O
Bouyakoub	Y(SMIL2)	PN	O	P	N
Bossi	Y(SMIL2)	A	N	O	O
Bouyakoub	Y(SMIL3)	PN	O	P	N
Bouyakoub	Y(SMIL2)	PN	O	P	N
Dahmani	Y(SMIL)	PN	O	P	N
<b>Notre approche</b>	<b>Y(SMIL3)</b>	<b>A</b>	<b>N</b>	<b>O</b>	<b>O</b>

**O : oui / N : Non / NI : No information.**

**Tableau III.7** Comparaison entre l'approche de [A. Bossi, O. Gaggi,] et notre approche

Auteur	begin	end	Dur	par	Seq	Excl	min	max	endsync	Repeat Dur	Calculer les intervalles	Générer les scenarios	TYP Moteur d'inférence
BOSSI	O	O	O	O	O	O	N	N	O	N	N	N	API prologue
<b>Notre approche</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>O</b>	<b>Native JAVA</b>

**O : oui / N : Non**

## **Chapitre III**

### **III.15 Conclusion**

SMIL est un langage standard du W3C pour la synchronisation des médias permettant de spécifier les dimensions temporelle et spatiale des DMM. Le langage permet l'interaction de l'utilisateur au moyen de liens de navigation hypermédia, de clics de souris ou de boutons d'interaction.

Parmi les avantages de SMIL, on peut citer l'interopérabilité avec les autres technologies Web (XML, CSS, DOM, etc.), l'indépendance par rapport aux protocoles d'accès, et sa grande adaptabilité aux systèmes et aux utilisateurs. C'est un langage standard puissant pour la modélisation des DMM mais qui manque de formalisme pour vérifier la consistance et les incohérences temporelle et spatiale des DMM.

Dans le chapitre IV suivant, nous allons détailler la logique de Hoare qui a été utilisée dans notre travail pour concevoir notre outil de validation des documents SMIL.

## Chapitre IV : La logique de Hoare

### IV.1 Introduction

La preuve automatique, qui consiste à laisser l'ordinateur prouver les propriétés automatiquement, étant données une description du système, un ensemble d'axiomes et un ensemble de règles d'inférences. Cependant la recherche de preuve est connue pour être un problème non décidable en général (en logique classique), c'est-à-dire que l'on sait qu'il n'existe (et n'existera jamais) aucun algorithme permettant de décider en temps fini si une propriété est vraie ou fausse.

Il existe des cas où le problème est décidable (fragment gardé de la logique du premier ordre par exemple) et où des algorithmes peuvent donc être appliqués. Cependant même dans ces cas, le temps et les ressources nécessaires pour que les propriétés soient vérifiées peut dépasser des temps acceptables ou les ressources dont dispose l'ordinateur. Dans ce cas il existe des outils interactifs qui permettent à l'utilisateur de guider la preuve. La preuve automatique, par rapport au model-checking, a l'avantage d'être indépendante de la taille de l'espace des états, et peut donc s'appliquer sur des modèles avec un très grand nombre d'états, ou même sur des modèles dont le nombre d'états n'est pas déterminé (modèles génériques).

Parmi les méthodes de la preuve on trouve la logique de Hoare, parfois appelée logique de Floyd-Hoare, est une méthode formelle définie par le chercheur en informatique britannique Tony Hoare dans un article de 1969 intitulé « An Axiomatic Basis for Computer Programming » [Hoa 69]. La méthode de Hoare met en place un formalisme logique permettant de raisonner sur la correction des programmes informatiques. Hoare s'est inspiré du travail sur les méthodes formelles dans les organigrammes de Robert Floyd, qui lui n'avait pas eu connaissance des travaux d'Alan Turing sur le sujet.

La logique de Hoare a des axiomes et des règles d'inférence pour toutes les instructions de base d'un langage de programmation impératif. Hoare rajoute dans son article originel des règles pour les procédures, les sauts, les pointeurs et la concurrence.

## IV.2 Logique de Hoare

Dans la logique de Hoare [Hoa69], un programme est considéré comme un transformateur d'états. Un état représente ici les valeurs de toutes les variables d'un programme. L'exécution d'un programme a pour effet, si elle se termine, de transformer un état initial en un état final.

La spécification d'un programme s'effectuera en formulant des propriétés sur ces deux états.

## IV.3 Les triplets de Hoare

Un triplet de Hoare  $\{P\} \text{ prog } \{Q\}$  est composé du programme prog, de la pré-condition P et de la post-condition Q. P et Q sont des formules logiques représentant des propriétés sur les variables du programme. En général il s'agit de formules de la logique des prédicats.

Soit le programme divide suivant :

```

a :=0;
b :=x;
while (b >= y)do
  b := b - y;
  a := a +1
done

```

Voici un premier exemple de triplet de Hoare :

$$\{y \neq 0\} \text{ divide } \{x=b+(a*y) \wedge b \leq y\} \quad (1)$$

Il faut lire ce triplet comme suit :

Si la propriété  $y \neq 0$  est vraie avant l'exécution de divide (c'est-à-dire que l'état initial vérifie  $y \neq 0$ )

Et l'exécution de divide se termine,

Alors la propriété  $(a*y)+b = x \wedge b \leq y$  est vraie après l'exécution de divide.

Il faut comprendre qu'un triplet peut être vrai ou faux, exactement comme une formule logique. Par exemple le triplet ci-dessous n'est pas vrai car il existe des états initiaux pour lesquels l'état final ne sera pas tel que  $b = 0$ . :

$$\{y \neq 0\} \text{ divide } \{(a * y) + b = x \wedge b = 0\} \quad (2)$$

En revanche le triplet suivant semble vrai :

$$\{x \leq 0 \wedge y \leq 0\} \text{ divide } \{(a*y)+b = x \wedge 0 \leq b \leq y\} \quad (3)$$

#### IV.4 Correction d'un triplet de Hoare

Le premier triplet ci-dessus (1) est problématique car, bien que vrai selon notre description intuitive ci-dessus, il implique des cas où le programme ne se termine pas. Par exemple si  $x$  est négatif et  $y$  positif. On est donc amené à considérer deux formes de correction pour le programme.

##### IV.4.1 Correction partielle

Le triplet de Hoare suivant  $\{P\} \text{ prog } \{Q\}$  est vrai si pour tout état initial vérifiant  $P$ , si l'exécution de  $\text{prog}$  se termine, alors  $Q$  est vraie après l'exécution de  $\text{prog}$ . On dit que le programme  $\text{prog}$  est partiellement correct par rapport à  $P$  et  $Q$ .

La correction totale s'écrit avec des  $\langle \rangle$  (parfois avec des  $[ ]$ ).

##### IV.4.2 Correction totale

Le triplet de Hoare suivant  $\langle P \rangle \text{ prog } \langle Q \rangle$  est vrai si pour tout état initial de  $\text{prog}$  vérifiant  $P$ ,  $\text{prog}$  se termine et  $Q$  est vraie après l'exécution de  $\text{prog}$ . On dit que le programme  $\text{prog}$  est totalement correct par rapport à  $P$  et  $Q$ .

#### IV.5 Les assertions

Dans les triplets ci-dessus (2)  $\{y \neq 0\}$ ,  $\{x \leq 0 \wedge y \leq 0\}$  et  $\{(a*y)+b=x \wedge 0 \leq b \leq y\}$  sont des assertions.

Plus précisément dans un triplet de la forme  $\{P\} \text{ prog } \{Q\}$  les assertions  $P$  et  $Q$  sont en général des formules de la logique de prédicats où les prédicats portent sur les variables du programme (notées en caractères d'imprimerie  $X, Y, \dots$ ) et sur des variables supplémentaires (notées  $x, y, \dots$ ).

Notez que les expressions booléennes du programme, par exemple  $x \leq 0 \text{ and } y \leq 0$  ont un équivalent immédiat dans les assertions, ici  $x \leq 0 \wedge y \leq 0$ , où  $\leq$  est un prédicat binaire.

Les premières sont des expressions du programme, ayant à l'exécution la valeur vraie ou fausse alors que les deuxièmes sont des formules logiques décrivant des propriétés sur les variables du programme.

Elles servent à écrire :

- Des pré- et post-conditions,
- Des assertions spéciales pour les boucles :
- Les invariants de boucle sont des formules logiques destinées à établir qu'une propriété est vraie à chaque itération d'une boucle.



– Les variantes de boucle ne sont pas des formules, mais des expressions, dont la valeur doit diminuer à chaque itération, permettant ainsi de démontrer que l’itération s’arrêtera toujours.

**IV.6 La règle d’inférence**

Un système logique est un couple (J;R), où J est un ensemble de jugements, et R un ensemble de règles de déduction ou règles d’inférence permettant de dériver le sous-ensemble R(J) inclus dans J des jugements dits valides.

La forme des jugements est très variable, dans certains systèmes logiques il s’agit des formules, dans d’autres des équations de la forme  $\square \vdash F$  où  $\square$  est un ensemble de formules et F une formule.

Dans la logique de Hoare il s’agira de triplets de Hoare ou de formules logiques.

Une règle de déduction, ou règle d’inférence, est une fraction de la forme  $\frac{J_1 \dots J_n}{J}$   
 Où  $J_i$  et J sont des jugements, les  $J_i$  sont les prémisses et J est la conclusion.

Il existe des règles particulières appelées axiomes qui n’ont aucune prémisses :  $\frac{}{J}$

Bien entendu la signification de ce type de règle est "le jugement J est toujours vrai".

Voici un exemple de système de règles de déduction dont les jugements sont de la forme  $i < j$  où  $i, j \in \mathbb{N}$  composé des deux règles nommées INF0 et INF+ :

$$\text{INF0 } \frac{}{0 \geq 0} \qquad \frac{x \geq y}{x + 1 \geq y} \text{ INF+}$$

Si, comme dans la règle INF+, les jugements contiennent des variables, alors on peut appliquer la règle à toute valeur des variables. Par exemple les règles suivantes sont des instances de INF+ :

$$\text{INF+ } \frac{3 \geq 2}{4 \geq 2} \qquad \frac{5 \geq 1}{6 \geq 1} \text{ INF+}$$

Dans la première x vaut 3 et y vaut 2, dans la deuxième x vaut 5 et y vaut 1.

**IV.7 Arbre de déduction**

Un arbre de déduction dans un système logique (J;R) est une combinaison finie de règles de la forme

$$\frac{\frac{\frac{\vdots}{J_{11}} \dots \frac{\vdots}{J_{1k_1}}}{J_1} \dots \frac{\frac{\vdots}{J_{n1}} \dots \frac{\vdots}{J_{nk_n}}}{J_n}}{J}$$

où chaque règle appliquée est une instance d’une règle. J est la racine de l’arbre, les jugements n’ayant pas de prémisses sont les feuilles.

L’ensemble des arbres de déduction d’un système R est défini récursivement comme suit :

– Si r est une instance d’une règle de R, alors r est un arbre de déduction;

– Si t1...tn sont des arbres de déduction dont les racines sont J1...Jn, et  $\frac{J_1 \dots J_n}{J}$

est une instance de règle de R alors l’arbre suivant est un arbre de déduction :  $\frac{t_1 \dots t_n}{J}$

**IV.7.1 Arbre de déduction complet**

Un arbre de déduction complet est un arbre dans laquelle toutes les feuilles sont des (instances) d’axiomes.

Le principe des systèmes de déduction est que les règles définissent l’ensemble des jugements prouvables .

**IV.8 Jugement prouvable**

Soit (J;R) un système logique, un jugement J est prouvable dans (J;R) si il existe un arbre de déduction complet avec J à la racine. On parle alors d’arbre de preuve pur J.

Donc pour prouver qu’un jugement J est prouvable dans un système de déduction (J;R), il faut et il suffit d’exhiber un arbre de preuve pour J dans R.

**IV.9 Les règles de Hoare**

Une règle de Hoare est une règle de déduction, c’est-à-dire une fraction de la forme :

$$\frac{\text{triplet}_1 \dots \text{triplet}_n}{\text{triplet}}$$

Les prémisses et la conséquence sont des triplets de Hoare. Une telle fraction se lit de la manière

suivante : "Si les triplets prémisses sont vrais, alors le triplet conclusion aussi".

Étant donné un triplet de Hoare donné T, si il existe un arbre de déduction de Hoare complet ayant T à sa racine, alors le triplet est vrai.

### IV.9.1 Affectation

La règle d'affectation signifie intuitivement que si on veut que P soit vraie après l'affectation, c'est-à-dire dans l'état où seul x a été changée en E, on doit vérifier que  $P[x := E]$  (c'est-à-dire P dans laquelle on a remplacé x par E) est vraie avant l'affectation.

Remarquez que cette règle n'a pas d'hypothèse, c'est un axiome.

$$\text{AFF1} \frac{}{\{P[x \leftarrow E]\} x := E \{P\}}$$

### IV.9.2 Séquence

Pour la «séquence» d'instructions, la règle est relativement simple à construire :

$$\text{SEQ} \frac{\{P\} C_1 \{Q\} \quad \{Q\} C_2 \{R\}}{\{P\} C_1 ; C_2 \{R\}}$$

### IV.9.3 Conséquence

Les propriétés que l'on peut prouver directement à partir des règles vues jusqu'ici sont d'une forme très contrainte, or il faut pouvoir déduire de ces propriétés celles qui nous intéressent. Ceci se fait par simple déduction logique à partir des propriétés prouvées par les règles précédentes. Pour cela on ajoute la règle suivante (notez que les deux implications ne sont pas dans le même sens) :

$$\text{CONSEQ} \frac{P \Rightarrow P' \quad \{P'\} C \{Q'\} \quad Q' \Rightarrow Q}{\{P\} C \{Q\}}$$

### IV.9.4 Conditionnelle

On ajoute ensuite la règle de la conditionnelle, qui exprime bien que la post-condition doit être vérifiée dans les deux branches du if chacune sous la condition que le résultat du test est conforme à la branche. En particulier si une des branches est impossible, alors la prémisse correspondante est trivialement juste (règle CONSEQ).

$$\text{COND} \frac{\frac{\{P \wedge B\} \quad I_1 \quad \{Q\} \quad \{P \wedge \neg B\} \quad I_2 \quad \{Q\}}{\{P\} \quad \text{if } B \text{ then } I_1 \text{ else } I_2 \quad \{Q\}}}{\{P\} \quad \text{if } B \text{ then } I_1 \text{ else } I_2 \quad \{Q\}}$$

### IV.9.5 While

Pour prouver la spécification d'une boucle, on a besoin d'un invariant de boucle. On doit vérifier que lorsque l'invariant est vérifié au départ de la boucle et que la condition de boucle E est vraie, alors l'invariant est toujours vrai à l'afin de la boucle. Lorsque l'on sort de la boucle, l'invariant est alors encore vrai et la condition de boucle n'est plus vérifiée.

$$\text{WHILE} \frac{\{P \wedge B\} \quad C \quad \{P\}}{\{P\} \quad \text{while } B \text{ do } C \text{ done} \quad \{P \wedge \neg B\}}$$

#### IV.10 Conclusion

En informatique, les méthodes formelles sont des techniques permettant de raisonner rigoureusement, à l'aide de logique mathématique, sur des problèmes informatiques, afin de démontrer leur validité par rapport à une certaine spécification. Elles sont basées sur les sémantiques, c'est-à-dire sur des descriptions mathématiques formelles du sens d'un problème donné par ses caractéristiques.

Parmi ces méthodes il ya la méthode de Hoare qui est basé sur la sémantique axiomatique, ou le programme n'est plus qu'un transformateur de propriétés logiques sur l'état de la mémoire : si on a  $p$  vrai avant exécution, alors on a  $q$  vrai après. On ne s'intéresse plus à l'état précis de la mémoire tant que l'on sait dire si la propriété tient. Mais En général ce qui est difficile lors de la preuve d'un problème se situent à trois endroits :

- (a) l'adaptation de méthode aux problèmes spécifiques.
- (b) L'écriture de assertions, en particulier les variantes et invariants sont parfois durs à trouver, ils nécessitent une compréhension profonde de problème.
- (c) les preuves des étapes de déduction (règle CONSEQ) peuvent être arbitrairement difficiles (ex : théorème de Bertrand).

Dans le chapitre V suivant, nous allons détailler la conception et la réalisation de notre outil de modélisation et de validation des documents SMIL et qui est basée sur la logique de Hoare.

## Chapitre V : Conception et Réalisation d'un Outil de Modélisation et de Validation des Documents SMIL

### V.1 Introduction

Depuis son apparition, le langage SMIL suscite un intérêt croissant, notamment dans le cadre de l'édition de DMM synchronisés interactifs. Le standard SMIL, tel qu'il a été défini par le W3C, est un langage basé sur XML de description de DMM synchronisés interactifs. L'aspect temporel est donc un point crucial pour de telles présentations. En plus de la dimension temporelle, SMIL intègre les dimensions spatiale, hypermédia et les Meta informations. La complexité de la synchronisation temporelle des présentations SMIL rend difficile voire impossible de garantir la validité d'un scénario en se basant sur des méthodes informelles ou empiriques.

Ce chapitre présente notre proposition de modélisation et de validation des documents SMIL en utilisant la logique de Hoare [Hoa 69]. En effet, ces derniers permettent de formaliser et valider les contraintes de synchronisation temporelle des documents SMIL, ce qui contribue à leur production et diffusion.

### V.2 Notre proposition

Dans le cadre de ce mémoire nous sommes intéressés à la modélisation et validation des documents SMIL, Par la logique de Hoare au moyen d'un ensemble de règles d'inférence. Nous avons présenté un outil pour aider à l'utilisateur de produire des documents SMIL avec un code robuste et Claire et détecte automatiquement les conflits temporels dans les documents SMIL et permet à l'utilisateur de les corriger en fournissant des messages utiles. L'outil est basé sur une sémantique formelle définissant les aspects temporels de SMIL.

La validité et la complétude de notre approche a été examinée selon une sémantique opérationnelle qui formalise les changements dans l'état d'un lecteur décrit de manière informelle dans la recommandation SMIL.

**Notre approche** basé sur une sémantique formelle définir les aspects temporels d'éléments SMIL au moyen d'un ensemble de règles d'inférence, Les règles dans l'esprit de la sémantique de Hoare, avec une implémentation **des règles et un moteur d'inférence purement en java**, cette approche permet au moteur d'inférence de calculer les intervalles de temps valides dans lequel les interactions de l'utilisateur doivent se produire.

En outre, il est capable de générer automatiquement des séquences d'événements pour tester les documents SMIL grâce à un module appelé générateur automatique des scenarios.

En plus de ça nous étendre l'ensemble des attributs couverts par des nouveaux attributs comme :

**MIN, MAX, RepeatDur**, pour offrir à l'auteur un plus de contrôle sur la durée active des objets SMIL.

### V.3 Définition d'une lecture abstraite

Les règles d'inférence définissant la sémantique formelle décrivent comment l'exécution d'un morceau de code SMIL change l'état de la lecture. La notion d'état d'un lecteur (ou de la présentation) qui sous-tend notre modèle est déterminé par un ensemble de valeurs particulières décrivant des aspects importants des objets médias. Parce que nous sommes intéressés à décrire uniquement les aspects qui pourraient influencer sur la cohérence temporelle, un état décrit des instants significatifs :

- l'instant de temps de début et de fin de tous les éléments SMIL contenue dans la présentation.
- la durée de chaque objet continu et l'instant d'interaction capturé par le lecteur.

La plupart de ces informations peuvent être récupérées directement à partir des documents SMIL. La seule information utile qui est absent des préoccupations de la durée naturelle de chaque média continu et les événements dus aux interactions des utilisateurs. Plus précisément, la durée naturelle nous entendons le nombre d'instant pour lesquels un élément de média en continu joue en l'absence d'interaction de l'utilisateur ou d'autres spécifications temporelles.

### V.4 Définition d'état d'un lecteur

L'état S d'un lecteur est un triple  $\langle C, Fd(), Fe() \rangle$  où :

- C est une horloge, est une valeur qui enregistre la progression de temps;
- Fd() est la fonction " description " qui mappe un identifiant 'id' à un triple, où  $Fd(id) = \langle B\_id, E\_id, DUR\_id \rangle$  telle que B\_id, E\_id, DUR\_id sont des nombres réels indiquant l'heure de début, l'heure de fin et la durée naturelle de l'élément identifié par id;
- Fe() est la fonction " temps de l'événement " qui enregistre le dernier instant où un événement se produit.

L'état du lecteur ne contient pas la durée effective d'un élément média par ce qu'il peut être calculé comme la différence entre son début et de fin. Cela ne veut pas possible pendant la durée naturelle.

Notons ici que la fonction Fe() enregistre uniquement les événements interactifs et non pas le début et la fin d'éléments et les médias (événements non interactifs) qui sont entièrement décrite par la fonction " description Fd().

Les événements interactifs peuvent impliquer un élément de médias, comme un clic sur un élément ou du mouvement de la souris dessus, mais il existe également des interactions de l'utilisateur qui ne sont pas liés à un élément spécifique comme l'utilisateur tapant une touche du clavier.

Dans le premier cas, la fonction en entrée nécessite le type de l'événement et l'ID de l'élément, dans le second cas, l'id est absent. Une liste partielle des événements pris en charge peut être trouvée dans (**Tableau V.1**).

Raison	Description
Les abréviations	media ≡ cont   static   ref cont ≡ video   audio   animation static ≡ text   img   brush cmd ≡ media   par   seq   excl m ≡ id = "m" ev ≡ id.begin / end   id.activateEvent / click   id.mouseover / mouseout   accesskey('c') ...
Les objets SMIL	Text, img, video, audio, animation, ref.
Conteneur de temp	par, seq, excl.
Les attributs	begin, end, dur, min, max, repeatDur, endsync.
Les valeurs admis	begin: a positive time value t, an event ev, ev+t. min: a positive time value t. max: a positive time value t. end: a positive time value t, an event ev, ev+t, indefinite., dur: a positive time value t. Endsync = first.

**Tableau V.1** les elements SMIL , atributs, abreviation utilisé

Comme décrit, événements interactifs peuvent se produire à plusieurs reprises, par exemple, un utilisateur peut cliquer à des moments différents sur une bouton. Par conséquent, un élément média peut être lu plusieurs fois en réponse à une interaction de l'utilisateur. Pour représenter ce fait, nous aurions associé à chaque identifiant une séquence des tuples représentant ses diverses exécutions. Nous choisissons une solution différente, dire de créer un nouveau nom à chaque fois un élément de média est joué.

Cela signifie que nous pouvons avoir plusieurs identificateurs se rapportant à différents activations du même objet : ils se réfèrent au même fichier, mais sont considérés comme complètement différent du point de vue de la synchronisation. Les nouveaux noms sont générées par la fonction V; ainsi si id identifie un objet média alors V(id),.....,V(V(id)) , ..... sont des identifiants différents pour les différentes exécutions du id. Notre outil prend en charge tous les conteneurs par , seq et excl, et les attributs 'Begin' (décalages négatifs ne sont pas autorisés), 'End', 'Dur', 'Min', 'Max', 'RepeatDur' comme indiqué dans le tableau.

### V.5 Les type d'événements

Un lecteur entre dans un nouvel état en réponse à un événement. La spécification SMIL considère deux types d'événements :

#### V.5.1 Les événements interactifs

C'est à dire, les interactions de l'utilisateur, comme un clic sur une image ou le mouvement de la souris.



### V.5.2 Les événements non interactifs

C'est à dire, les événements en raison de la synchronisation SMIL, exemple le début ou la fin d'un objet média. L'état d'un lecteur doit enregistrer toutes ces informations.

### V.6 Le langage d'assertion pour SMIL

Les règles prévoient une sémantique axiomatique pour les aspects temporels d'éléments XML dans l'esprit de la logique de Hoare. Ils nous permettent de dériver des jugements sous la forme de triplets:  $\{P\}C\{Q\}$  où P et Q sont des assertions, respectivement, la précondition et la postcondition, et C est un élément SMIL.

Le langage d'assertion utilisée pour exprimer les pré / post conditions comprend un ensemble des fonctions représentant les aspects temporels importants des médias. Les assertions sont formées par des ensembles de contraintes sur les valeurs renvoyées par ces fonctions. Le tableau 3 énumère toutes les fonctions et abréviations utilisées dans les affirmations.

**Tableau V.2** Liste des fonctions et notation utilisé pour définit les règles d'inférence

Function	Pre/post condition	Description
$t_{cr} : I \rightarrow \mathbb{R}$	Pre	returns the current time instant in which the SMIL element <i>id</i> is evaluated
$dur : I \rightarrow \mathbb{R}$	Pre	returns a real value representing the time interval for which a continuous media item plays
$times : (\mathcal{E} \times (I \cup \{\perp\})) \rightarrow \mathbb{R}^*$	Pre	returns the sequence of all the time instants in which an event occurs
$time : (\mathcal{E} \times (I \cup \{\perp\})) \rightarrow \mathbb{R}^+$	Pre	returns the time instant of the next occurrence of an event
$begin : I \rightarrow \mathbb{R}$	Post	returns the time instant media item <i>id</i> starts
$end : I \rightarrow \mathbb{R}$	Post	returns the time instant media item <i>id</i> ends
Abbreviation	Description	
$begin_B(c)$	denotes $t$ if $\{begin(c) = t\} \subseteq B$	
$end_B(c)$	denotes $t$ if $\{end(c) = t\} \subseteq B$	

Par exemple, nous écrivons  $Begin(C) = 10$  pour dire que les médias C commence son exécution au temps d'horloge 10.

Compte tenu de l'affirmation B qui contient l'égalité  $Begin(C) = t$  ou  $End(C) = t$  nous utilisons également la notation  $Begin\_B(C)$  ou  $End\_B(C)$  pour indiquer l'instant de temps t.

Nous notons ici que le langage SMIL permet de multiples formats de valeurs d'horloge, par exemple 00:02:33, 2:33 et 153 s , Comme ils peuvent être aisément convertis en nombre réel représentant le nombre de secondes, nous supposons que toutes nos fonctions renvoient une valeur réelle.

Le langage d'assertion contient également des fonctions TCR (id), qui représente l'instant de temps auquel l'élément SMIL id est évalué, c'est à dire la valeur d'horloge d'état dans lequel, considérant un lecteur exécuter la présentation, les spécifications SMIL appellent ce temps, la base de synchronisation implicite, nous l'utilisons dans le pré condition.

La survenue d'interactions de l'utilisateur est représentée dans le pré condition d'un élément par une égalité de la forme  $\text{times}(\text{event}) = (t_1, t_2, \dots, t_n)$  ou  $t_1, t_2, \dots, t_n$  sont les prochains instants de temps dans lesquels l'événement se produit.

L'instant de temps dans laquelle un événement se produit est la valeur de l'horloge lorsque le lecteur enregistre la survenance de cet événement.

Nous utilisons aussi la fonction  $\text{time}(\text{event})$  pour représenter simplement la prochaine occurrence de l'événement, soit le premier élément dans la séquence.

Soit  $S$  un Etat et  $A$  une affirmation, nous disons que  $S$  satisfait l'assertion  $A$ , et écrire  $S \models A$  si les contraintes sur les valeurs réelles obtenues en appliquant à  $A$  les transformations suivantes sont réunies:

- chaque occurrence de  $\text{Begin}(\text{id})$ ,  $\text{End}(\text{id})$ ,  $\text{dur}(\text{id})$  et  $\text{tcr}(\text{id})$  est remplacé par le composant correspondant de l'état du lecteur;
- chaque occurrence de  $\text{times}(\text{event}) = (t_1, t_2, \dots, t_n)$  est remplacé par  $S\_C \leq t_1$  où  $S = \langle C, \text{Fd}(), \text{Fe}() \rangle$ .
- chaque occurrence de  $\text{time}(\text{event}) = (t_1, t_2, \dots, t_n)$  est remplacé par  $S\_C \leq t$  où  $S = \langle C, \text{Fd}(), \text{Fe}() \rangle$ .

Un état satisfait l'égalité  $\text{times}(\text{event}) = (t_1, t_2, \dots, t_n)$  si elle permet à la première occurrence de cet événement, c'est à dire, si la valeur de son horloge est inférieure ou égale à  $t_1$ . Depuis le lecteur entre dans un nouvel état en réponse à un événement, pour chaque instant de temps  $t_i, 1 \leq i \leq n$ , il existe un état  $S_i$  tel que  $S_i \models C = t_i$ .

Le triple  $\{P\} \{c\} Q$  peut être lu comme: chaque fois que l'évaluation de la commande SMIL  $C$  commence dans un état  $S_0$  qui satisfait l'assertion  $P$ , c'est à dire,  $S_0 \models P$ , puis il se termine dans un état  $S_f$  qui satisfait l'assertion  $Q$ , c'est-à-dire,  $S_f \models Q$ . Selon cette interprétation, la règle suivante de la conséquence est retenue:

$$\text{CONSEQ} \frac{P \Rightarrow P' \quad \{P'\}C\{Q'\} \quad Q' \Rightarrow Q}{\{P\}C\{Q\}}$$

D'une manière générale, dans le triple  $\{P\} c \{Q\}$  la condition  $P$  contient l'instant de temps de la commande  $C$ , la durée naturelle des éléments médias qu'elle définit et l'occurrence d'événements.

La postcondition  $Q$  contient la définition des instants de temps dans lequel les éléments définis en  $C$   $\text{Begin}$  et / ou  $\text{End}$ .

### V.7 Conventions de notation

nous utilisons un certain nombre de conventions de notation spéciales pour introduire l'ensemble des règles d'inférence décrivant la sémantique des éléments SMIL. Le tableau 1 énumère un ensemble des abréviations utilisées pour la représentation des éléments SMIL. Par exemple  $\langle \text{cmd } c \rangle$  représente tout élément SMIL avec l'attribut  $\text{id} = "c"$ .

En outre, nous utilisons la forme générale  $end='k'$  et  $dur='k'$  pour représenter les attributs d'un élément où la méta-variables  $k$  est soit l'une des valeurs admises pour l'attribut particulier, ou la valeur spéciale «void». La valeur «void» représente l'absence de cet attribut dans l'élément et nous permet de présenter une seule règle générale pour chaque Conteneur SMIL. En ce qui concerne l'attribut «Begin», nous supposons qu'il est toujours définie depuis son absence peut être représentée par la valeur  $k = "0"$ . Par exemple, `<video id= 'v' begin='0' dur='5' end='void'/>` et considéré comme un synonyme de `<video id= 'v' dur='5'/>`.

L'avantage de cette représentation est d'éviter la répétition des règles très similaires, mais nous avons besoin d'un ensemble de prédicats pour vérifier l'existence de la valeur d'un attribut avant de l'utiliser. Nous avons également besoin de classer les éléments qui apparaissent dans un document SMIL par rapport aux valeurs de leurs attributs "Dur" et "fin". Par conséquent, nous introduisons quelques prédicats auxiliaires et ensembles dont la description peut être trouvée dans (**Tableau V.3**).

**Tableau V.3** Liste des prédicats et ensembles utilisé pour définit les règles d'inférence

Name	Description
<i>finite(k)</i>	holds if $k$ is a real value
<i>indefinite(k)</i>	holds if $k$ is equal to ``indefinite``
<i>defined(k)</i>	holds if $k$ is not equal to ``void``
<i>NotDur</i>	contains all the statements with attributes $dur$ and $end$ equal to $void$
<i>Closure(c)</i>	contains $c$ and all the statements defined inside the element $c$ , at any level of nesting
<i>Indef(c)</i>	holds if in <i>Closure(c)</i> there are elements with attribute $end$ (or $dur$ ) equal to ``indefinite``

### V.8 Les règles d'inférence pour SMIL

La définition du langage de SMIL fournies par Bulterman et al. [Bult et Al 05]. ne contient pas une présentation formelle de la sémantique des éléments et attributs. Les recommandations SMIL est divisé en sections, dont certains sont définies " Normative ". Pour clarifier le sens de valeurs particulières pour les attributs, il est prévu un algorithme de mieux expliquer comment les instants de temps important sont calculées dans ce cas, mais aucune définition formelle ou outils de vérification ont été mis en œuvre par Le Groupe de travail multimédias synchronisées du W3C pour vérifier l'exactitude sémantique des éléments SMIL.

Dans ce chapitre, nous définissons un système formel qui est capable à savoir les conflits temporels d'une présentation multimédia définie en utilisant SMIL. Le système fournit une logique Hoare pour

SMIL par un ensemble de règles d'inférence décrivant comment l'exécution d'un morceau de code modifie l'état du lecteur.

Puisque la spécification SMIL manque un formelle sémantique opérationnelle, la solidité et la complétude de notre approche ne peut pas être formellement prouvé, mais on considère que nos sémantique est correct selon l'une quelconque sémantique opérationnel qui formalise les changements dans l'état d'un lecteur décrit dans la recommandation SMIL.

Nos règles visent à découvrir conflits sémantiques peut être la cause de comportements inattendus même se ils ne sont souvent pas considérées comme des erreurs par recommandation SMIL. Par conséquent, nos règles ne peuvent pas valider les présentations multimédias qui contiennent certains conflits, même se ils sont corrects en fonction de la spécification standard. Nous notons ici que nous ne limitons pas l'ensemble des comportements qui peuvent sera décrit par SMIL, mais seulement l'ensemble des documents qui décrivent ces comportements, contribuant ainsi à l'utilisateur de développer un code robuste et claire. En fait, la présence de conflits, en plus d'être la principale cause de comportement inattendu et affecte grandement la maintenabilité du code.

Nous commençons par examiner les éléments autonomes, c'est à dire, commandes SMIL dont la synchronisation ne se réfère pas à d'autres conteneurs ou éléments médias.

### V.8.1 Règles de preuve pour les objets médias

<pre> STATIC+BEGIN {A ∪ Pre} &lt;static m begin='k1' /&gt; {A ∪ Post} where Pre = {t<sub>cr</sub>(m) = start - k1} and Post = {begin(m) = start}  CONT+BEGIN {A ∪ Pre} &lt;cont m begin='k1' /&gt; {A ∪ Post} where Pre = {t<sub>cr</sub>(m) = start - k1, dur(m) = stop - start}       Post = {begin(m) = start, end(m) = stop}  MEDIA+BEGIN+END+DUR {A ∪ Pre} &lt;media m begin='k1' end='k2' dur='k3' /&gt; {A ∪ Post} where Pre = {t<sub>cr</sub>(m) = start - k1}, Post = {begin(m) = start} ∪ End and       End = {         {end(m) = start - k1 + k2} if finite(k2)         {end(m) = start + k3}    if finite(k3)         ∅                        otherwise       }  APPLICABILITY CONDITION: (defined(k2) ∨ defined(k3)) ∧ ((finite(k2) ∧ finite(k3)) ⇒ k3 = k2 - k1) ∧ (finite(k2) ⇒ ¬indefinite(k3)) ∧ (finite(k3) ⇒ ¬indefinite(k2)) </pre>
--

**Figure V.1** Les règles (Static+Begin, Cont+Begin, Madia+Begin+End+Dur)

Les axiomes qui vérifient l'exactitude des éléments médias sont listés dans la **Figure V.1**. Supposons, nous voulons vérifier le triple :

$\{P\}\langle \text{vidéo id} = 'v' \text{ begin} = '2' \rangle / \{Q\}$

Ou la précondition  $P = \{\text{dur}(v) = 5, \text{tcr}(v) = 0\}$  et la postcondition  $Q = \{\text{begin}() = 2, \text{end}() = 7\}$

Nous pouvons prouver son exactitude puisque nous ne pouvons instancier l'axiome CONT+BEGIN en utilisant les valeurs

Start = 2, K1= 2 et stop=7.

Le système peut également être utilisé pour trouver le temps de début et de fin des éléments médias, c'est à dire, en tant que base pour la mise en œuvre d'un lecteur. Dans ce cas, l'axiome CONT+BEGIN décrit la transformation de l'état du lecteur.

si le lecteur commence dans un état  $S_0 \vdash P$ , c'est à dire,  $S_0 = \langle 0, \text{Fd}(), \text{Fe}() \rangle$  tel que  $\text{Fd}(v) = \langle \perp, \perp, 5 \rangle$  nos règles montrent qu'il se termine dans un état  $S_f \vdash Q$ , c'est à dire,  $S_f = \langle 7, \text{Fd}(), \text{Fe}() \rangle$  tel que  $\text{Fd}(v) = \langle 2, 7, 5 \rangle$  Ainsi, nous obtenons les valeurs utilisées pour start et stop la vidéo.

Nos règles nous permettent de décrire également des éléments médias sont terminés avant leur durée naturelle en raison de la définition de end, ou dur. Par exemple, considérez la triple:

$\{\text{tcr}(v) = 0, \text{dur}(v) = 5\} \langle \text{vidéo id} = 'v' \text{ begin} = '2' \text{ end} = '3' \text{ dur} = 'void' \rangle / \{\text{begin}(v)=2, \text{end}(v)=3\}$

Nous pouvons instancier la règle MEDIA+BEGIN+END+DUR avec les valeurs start = 2, k1 = 2, k2 = 3 et k3= void. La condition d'applicabilité détient depuis k2 est finie . et K3 =void . Dans ce cas, conformément à la recommandation de SMIL, si le lecteur commence dans un état  $S_0$  qui satisfait les pré conditions c'est à dire, de telle sorte que  $S_0\_C = 0$ , puis l'état final atteint par le lecteur est  $S_f = \langle 3, \text{Fd}(), \text{Fe}() \rangle$  ou  $\text{Fd}(v) = \langle 2, 3, 5 \rangle$  et ainsi  $S_f \vdash Q$ .

Notez que dans la **Figure V.1**, la règle MEDIA+BEGIN+END+DUR définit le temps de fin d'un élément média m seulement si k2 et k3 sont pas égaux à "indefinite". En outre, la définition des éléments médias ne conduise pas à des conflits temporels moins que l'auteur définit à la fois les attributs end et dur. La condition d'application ne permet pas l'application de la règle en présence de valeurs incorrectes de ces attributs ; par exemple, lorsque les deux attributs Dur et la fin sont finis, la relation  $K_3 = k_2 - k_1$  doit détenir; par ailleurs, la condition d'applicabilité souligne le conflit temporel à l'utilisateur.

V.8.2 Règle de preuve pour la composition parallèle 'PAR'

PAR+BEGIN+END

$$\frac{\{A_i \cup \{t_{cr}(c_i) = init + k1\}\} c_i \{B'_i\} \quad \forall i \ 1 \leq i \leq n}{\{A'\} \langle par \ c \ begin='k1' \ end='k2' \ dur='void' \rangle c_1 \dots c_n \langle /par \rangle \{B\}}$$

where  $A' = \bigcup_{i=1}^n A_i \cup \{t_{cr}(c) = init\}$   
 $B = \bigcup_{i=1}^n B_i \cup \{begin(c) = init + k1\} \cup End$   
 $stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ max_{c_i} \{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$   
 $End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$   
 $B'_i = \begin{cases} B_i \setminus \{end(c_i) = stop\} & \text{if } c_i \in NotDur \wedge finite(k2) \\ B_i & \text{otherwise} \end{cases}$

APPLICABILITY CONDITION:  
 $finite(k2) \implies ((\neg Indef(c) \wedge k2 \geq k1) \wedge \forall c_i (end_{B_i}(c_i) \leq stop \vee c_i \in NotDur))$   
 $\wedge Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$   
 $\wedge \forall c_i \ begin_{B_i}(c_i) \geq init + k1$



Figure V.2 La règle PAR + Begin + End

Si les définitions des médias sont imbriquées dans la composition parallèle et séquentielle, l'évaluation de ces structures nécessite la définition de règles plus complexes. Nous commençons notre analyse en examinant la composition parallèle exprimé par l'élément par. Lorsque l'attribut dur n'est pas présent (c'est à dire, dur = "void"), l'attribut begin est présente (possible avec la valeur zéro) et l'attribut end est 'void', ou 'indefinite' ou un nombre réel.

La règle PAR+BEGIN+END décrite dans la **Figure V.2** définit la sémantique de la composition parallèle dans ces cas, dans le postcondition, nous distinguons les composants ...B1.....Bn pour qu'il soit clair que la postcondition contient des informations sur chaque ci, ... que ce soit un objet média ou une structure de synchronisation.

Pour prouver la validité de l'élément <par c> c1, c2, ..... cn < par c /> chaque ci doit être correct en supposant que son instant égal l'instant de temps de l'élément parallèle, plus le décalage donné par l'attribut begin ,c'est-à-dire, si (tcr(c) = init) est contenue dans la précondition de l'élément c, la précondition de chaque élément ci doit contenir (tcr(ci) = init+K1) tel que K1 >= 0, est la valeur de l'attribut et begin et init est l'instant de temps dans laquelle l'instruction par est évaluée.

L'évaluation de l'instant fin pour par n'est pas toujours possible, car il n'est pas possible de calculer le temps d'un élément média dans deux cas:

- si c'est un objet statique sans attribut end ou dur,
- ou si l'un de ces (end ou dur) attributs est égal à " indéfinite " .

De la même manière, l'instant fin d'un conteneur par ne peut être calculé si l'attribut fin (ou Dur) est égal à " indéfinite ", ou si elle contient un enfant avec cette valeur pour l'attribut fin (ou dur). Dans ces cas, l'élément se termine avec la présentation

Une fois que nous sommes capables de décider si une composition parallèle se termine, nous devons calculer l'instant d'arrêt stop.

Si l'élément c ne contient pas la définition de l'attribut end (end = 'void'), alors c termine lorsque tous ses enfants, qui ne sont pas des objets statiques, finissent leurs lectures, c'est-à-dire, au moment instant d'arrêt

$Stop = \text{Max} \{end(c_i)\}$  , Si non l'élément c termine a l'instant de temps  $stop = \text{init} + K2$  , avec une exception nous n'avons pas attribuer une valeur d'arrêt lorsque certains éléments définis à l'intérieur c ont une durée supérieure à c, défini avec un attribut dur ou end. Dans ce cas, l'auteur donne un double, et contradictoire définition de la durée des éléments concernés, qui générant un conflit temporel.

Notez qu'il ya un conflit temporelle même lorsque la composition parallèle a une durée finie, mais contient certains enfants d'une durée indéterminée, qui est, par définition, plus longue que toute autre valeur finie.

La condition d'applicabilité nous empêche d'appliquer la règle PAR+BEGIN+END dans ces cas :

En particulier, la condition  $(\text{finite}(K2) \Rightarrow \neg \text{indef}(c) \wedge K2 \geq K1)$  dit que, en présence d'une valeur finie de k2, la règle peut être appliqué à c, que si elle ne contient pas, à ne importe quel niveau d'imbrication un élément avec une durée indéterminée. En outre, l'élément doit se terminer après son début  $(K2 \geq K1)$ .

La condition d'applicabilité  $\text{indef}(c) \Rightarrow (\neg \text{defined}(K2) \vee \text{indefinite}(K2))$  indique que l'attribut end doit être égal à 'indefinite' ou 'void' si c ne se arrête pas.

Enfin, la condition  $(\diamond c_i \text{ beginB}(c_i) \geq \text{init} + K1)$  exprime que tous les enfants de c doivent commencer avec c ou après.

Illustrons avec un exemple simple comment nos règles trouver ces conflits. Considérons la composition parallèle suivant:

```
<par id='p' begin='0' end='5'>
  <image id='i' begin='0' end='5'>
  < text id='tx' begin='0' end='7'>
</par>
```

Même si le conflit temporel est évident puisque l'élément est simple (texte tx a un dure plus long que l'élément dans lequel il est contenu), nous essayons de vérifier la validité sémantique de cette déclaration pour montrer comment le système fonctionne.

Nous tenons à prouver que

$$\{tcr(p) = 0\} \langle \{begin(i)=0, end(i) \leq 5, begin(tx)=0, end(tx) \leq 5, begin(p)=0, end(p) = 5\}$$

mais la déclaration p n'est pas correcte puisque la règle PAR+BEGIN+END ne peut pas être appliquée.

En fait, tx et i n'appartiennent pas à l'ensemble NotDur, pour appliquer la règle il faut prouver les prémisses:

$$Si = \{tcr(i) = 0\} \{begin(i) = 0, end(i) = 5\}$$

$$Stx = \{tcr(tx) = 0\} \{begin(tx) = 0, end(tx) = 5\}$$

Le premier triple Si est valide et nous pouvons le prouver par l'axiome MEDIA+BEGIN+END+DUR, mais nous ne pouvons pas prouver la triple Stx qui n'est pas valide. Par conséquent, la règle PAR+BEGIN+END ne peut pas être appliquée depuis la prémisse Stx ne peut pas être vérifiée.

Dans ce cas, la réponse de notre outil c'est que la présentation contient un conflit sémantique puisque le média tx se termine à l'instant de temps 7 tandis que son père se termine à l'instant de temps 5.

### V.8.3 Règle de preuve pour la composition séquentielle 'SEQ'

La règle qui décrit la sémantique de la composition séquentielle est très similaire à la règle PAR+BEGIN+END par ce que les deux éléments peuvent exprimer la même synchronisation si les valeurs des attributs sont correctement définis.

Il ya seulement deux différences:

$$\frac{\text{SEQ+BEGIN+END} \quad \{A_i \cup \{tcr(c_i) = init_{c_i}\}\} c_i \{B'_i\} \quad \forall i \ 1 \leq i \leq n}{\{A'\} \langle seq \ c \ attribute\text{-list} \rangle c_1 \dots c_n \langle /seq \rangle \{B\}}$$

where  
`attribute-list`  $\equiv$  `begin='k1' end='k2' dur='void'`  
and

$$A' = \bigcup_{i=1}^n A_i \cup \{tcr(c) = init\}$$

$$init_{c_i} = \begin{cases} init + k1 & \text{if } i = 1 \\ end_B(c_{i-1}) & \text{if } i > 1 \end{cases}$$

$$B = \bigcup_{i=1}^n B_i \cup \{begin(c) = init + k1\} \cup End$$

$$stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ max_{c_i} \{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$$

$$End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$$

$$B'_i = \begin{cases} B_i \setminus \{end(c_i) = h\} & \text{if } (begin_B(c_i) = h) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$$finite(k2) \implies \neg Indef(c)$$

$$\wedge Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$$

$$\wedge finite(k2) \implies \forall c_i \ end_B(c_i) \leq stop \vee c_i \in NotDur$$

$$\wedge \forall c_i \ begin_B(c_i) \geq init + k1$$

Figure V.3 La règle SEQ+Begin+End.



- En premier lieu, l'instant de temps en cours de chaque enfant est égal à l'instance de temps de fin de l'enfant précédent, et non à l'instant de temps en cours de l'élément Seq.
- En second lieu, l'instruction Seq impose une durée égale à zéro à des éléments médias statiques qui n'ont pas une durée déterminée, c'est-à-dire,  $begin(c_i) = h$  et  $end(c_i) = h$  si  $c_i$  est un média statiques contenues dans NotDur.

Cela signifie qu'ils ne sont jamais joués dans l'écran de l'utilisateur. Dans ce cas, notre outil affiche un avertissement à l'utilisateur qui peut être considéré ou non.

#### V.8.4 Règle de preuve pour la composition exclusive 'EXCL'

Le langage SMIL fournit également un conteneur pour la composition exclusive d'éléments média, l'élément «Excl», dont la sémantique indique que seulement l'un de ses enfants est actif à un instant de temps donné.

Ceci est très similaire à la composition séquentielle où un seul enfant est actif à tout moment, mais "Excl" ne pas imposer un ordre quelconque dans la visualisation des enfants. Cela signifie que chaque enfant peut contenir l'attribut "begin" dans la définition, ou peuvent être activés par l'utilisateur,

EXCL+BEGIN+END

$$\frac{\{A \cup \{t_{cr}(c_i) = init + k1\}\} c_i \{B'_i\} \quad \forall i \ 1 \leq i \leq n}{\{A'\} \langle excl \ c \ begin='k1' \ end='k2' \ dur='void' \rangle c_1 \dots c_n \langle /excl \rangle \{B\}}$$

where  $A' = A \cup \{t_{cr}(c) = init\}$

$$B = \bigcup_{i=1}^n B_i \cup \{begin(c) = init + k1\} \cup End$$

$$stop = \begin{cases} init + k2 & \text{if } finite(k2) \\ \max_{c_i} \{end_{B_i}(c_i)\} & \text{if } \neg defined(k2) \end{cases}$$

$$End = \begin{cases} \{end(c) = stop\} & \text{if } \neg Indef(c) \\ \emptyset & \text{otherwise} \end{cases}$$

$$B'_i = \begin{cases} B_i \setminus \{end(c_i) = t_i\} & \text{if } \exists j \ begin_B(c_j) = t_i \vee (finite(k2) \wedge c_i \in NotDur) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$$finite(k2) \implies \neg Indef(c) \wedge k2 \geq k1$$

$$\wedge Indef(c) \implies (\neg defined(k2) \vee indefinite(k2))$$

$$\wedge finite(k2) \implies \forall i \ (end_B(c_i) \leq stop \vee c_i \in NotDur)$$

$$\wedge \forall(i) \ begin_B(c_i) \geq init + k1$$

$$\wedge \forall(i, j) \ (begin_B(c_i) \leq begin_B(c_j)) \implies (end_B(c_i) \leq begin_B(c_j))$$

**Figure V.4 La règle EXCL+Begin+End.**

par exemple, suivant un lien. Prenons l'exemple suivant:

```
<par>
  <img id = "a" /> <img id = "b" /> <img id = "c" />
  <excl id="e" dur="10">
    <video id = "video_a" begin="a.activateEvent"/>
    <video id = "video_b" begin="b.activateEvent"/>
    <video id = "video_c" begin="c.activateEvent"/>
  </excl>
</par>
```

Dans ce cas, l'utilisateur choisit un clip vidéo en cliquant sur un bouton d'image choisi entre les éléments de média a, b et c. La vidéo correspondant est activé par l'événement activateEvent, L'élément excl indique simplement qu'un seul clip vidéo joue à la fois: en fait, la vidéo en cours de lecture est arrêté lorsque l'utilisateur clique sur une autre image, choisir un autre clip vidéo.

L'exemple montre comment le conteneur «excl» ne traite pas de l'activation de ses enfants, mais avec leur désactivation, en fait, l'ordre de lecture des clips vidéo dépend entièrement sur les choix de l'utilisateur et non pas sur les définitions des éléments. La sémantique de l'élément excl est décrite dans la **Figure V.4** par la règle EXCL+BEGIN+END.

Comme les éléments « Par » et « Seq », l'élément« excl »commence à son instant courant, ou après des k1 instants si l'attribut «begin» est finite, et se termine lorsqu'il n'y a pas des enfants qui jouent.

Cela signifie que, elle peut avoir une durée instantanée si aucun enfant ne commence en même temps avec lui. Pour cette raison, l'attribut «end» de cette déclaration ne contient généralement pas la valeur spéciale " void ".

La règle EXCL+ BEGIN+END est très similaire à la règle qui décrit la sémantique de composition parallèle, par conséquent, nous ne répétons pas ici le problème de la fin de l'élément. Même dans ce cas, de prouver la validité de la déclaration <excl>c1, c2, ..... cn</excl> chacun ci doit être prouvé pour être correcte, en supposant que l'instant de temps de son élément parent égal son instant courant. Parce que le conteneur exclusive peut imposer un arrêt prématuré de la lecture de ses enfants, nous ne nécessitent pas de connaître à l'avance l'instant de temps dans lequel l'enfant ci se termine :

- 1- lorsque ci se termine en même temps avec « excl » que sauf car il ne contient pas l'attribut end ou dur dans sa définition (par exemple, k2 est finite et  $ci \in \text{NotDur}$ ) ou
- 2- lorsque la lecture de ci est arrêtée avant son terminaison en raison de l'interaction de l'utilisateur ou un autre événement externe

La condition de l'applicabilité empêche l'application de la règle en présence de conflits temporels. Parmi les conditions déjà évoquées pour la composition parallèle, la condition :  $(\diamond ci, cj ( \text{begin}(ci) \leq \text{begin}(cj) ) \implies ( \text{end}(ci) \leq \text{begin}(cj) ))$  précise que seul un enfant joue à chaque instant de temps donné, par exemple, si des enfants ci commence avant cj; elle se termine aussi avant le début de cj.

Nous notons ici que chaque enfant peut être joué à plusieurs reprises, depuis leurs exécutions sont généralement liée par interaction de l'utilisateur, par exemple, dans l'exemple précédent, l'utilisateur peut cliquer plusieurs fois sur les images a, b et c. Cette situation est résolue en appliquant la règle BEGIN+MULTIPLEPLAYBACKS à l'élément de média répéter.

### V.8.5 Règle de preuve pour une composition générale d'éléments lorsque l'attribut Dur est Défini

Nous ne considérons que l'utilisation d'attributs begin et end mais, comme déjà discuté pour la définition des éléments médias, les déclarations peuvent également contenir un attribut Dur dont la sémantique est très

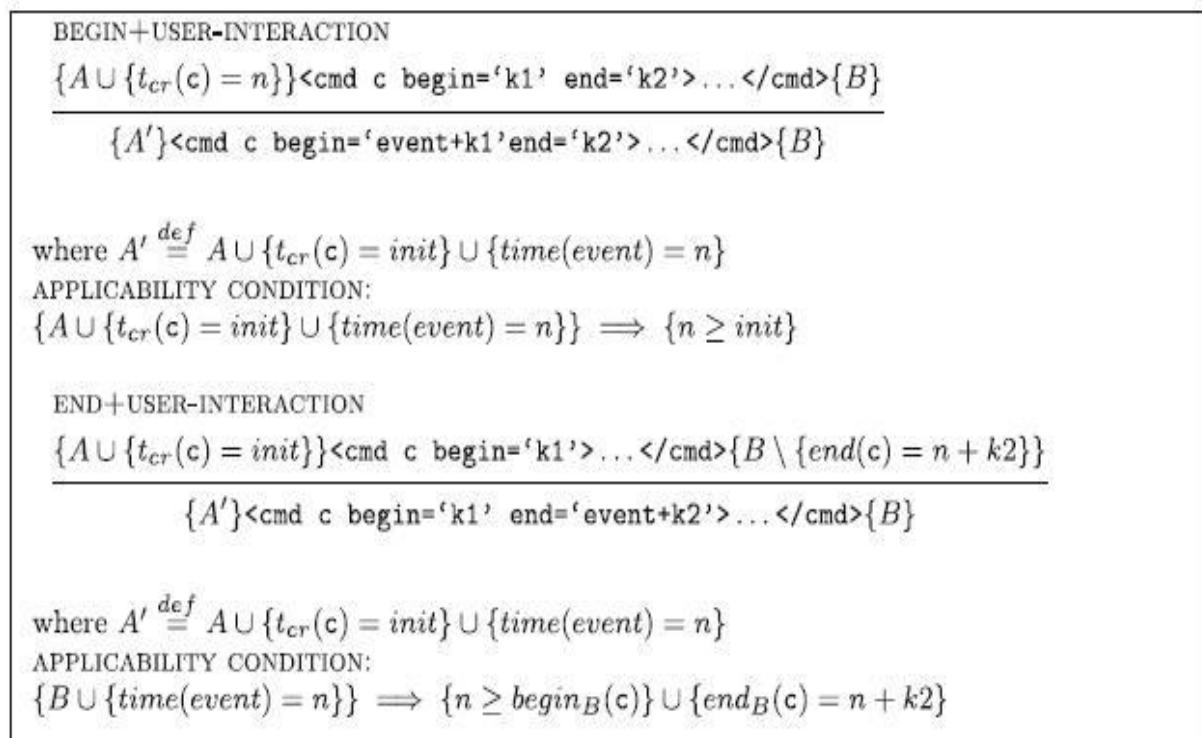
similaire à l'attribut end et donc une traduction facile peut être obtenue avec la règle CMD+BEGIN+END+DUR comme illustrer dans la **Figure V.5**.

<p style="margin: 0;">CMD+BEGIN+END+DUR</p> $\frac{\{A\} \langle \text{cmd } c \text{ begin}='k1' \text{ end}='k2' \text{ dur}='void' \rangle c_1 \dots c_n \langle / \text{cmd} \rangle \{B\}}{\{A\} \langle \text{cmd } c \text{ begin}='k1' \text{ end}='k4' \text{ dur}='k3' \rangle c_1 \dots c_n \langle / \text{cmd} \rangle \{B\}}$ <p style="margin: 0;">APPLICABILITY CONDITION:</p> $\begin{aligned} & \text{finite}(k3) \implies (\text{finite}(k2) \wedge k3 = k2 - k1) \\ & \wedge (\text{defined}(k4) \implies k4 = k2) \wedge (\text{indefinite}(k2) \iff \text{indefinite}(k3)) \end{aligned}$
---

**Figure V.5** La règle CMD + Begin + End + Dur

### V.8.6 Règles de preuve pour les éléments SMIL avec un événement interactif dans 'Begin' ou 'end'

SMIL permet aussi l'usage des événements comme valeurs possibles pour les attributs 'begin' et 'end' .  
 Considérons d'abord le cas dans lequel begin (ou end) d'un élément ou d'un groupe d'élément médias, se produit à la suite d'une interaction de l'utilisateur, par exemple lorsque l'utilisateur touches dans un caractère, disent 's', dans le clavier tel que décrit par l'élément suivant : `<cmd c begin = 'accesskey's'+K1'>.....</cmd>`  
 où 'accesskey's' signifie que l'utilisateur doit entrer le caractère 's' et  $K1 \geq 0$  représente un nombre de secondes.



**Figure V.6** La règle Begin + User-Interaction et End + User-Interaction

La validité de cette commande peut être prouvé que si nous connaissons déjà l'instant où l'événement accesskey (s) se produit. le lecteur enregistré la dernière occurrence d'un événement interactif dans l'état par la fonction Fe(c) qui enregistre l'instant de temps dans laquelle un événement se produit.

Depuis le lecteur entre dans un nouvel état en réponse à un événement, pour chaque occurrence d'un événement, Il existe un état tel que la valeur d'horloge est égale à l'instant de temps enregistré à partir la fonction Fe(), un événement interactif peut impliquer un seul élément médias, ou tout l'environnement .

Dans la précondition d'une commande, nous utilisons les Fonctions times(événement) et time(événement) pour faire des contraintes sur les événements d'entrée. Par exemple, la précondition {time(événement) = n } indique que l'état initial dans lequel l'élément est évaluée devrait permettre l'événement  $\rightarrow S_c \leq n . S_c = \text{l'horloge}$ .

La survenance de l'événement modifie l'instant courant de l'élément, qui est maintenant égale à l'instant de temps de l'événement.

La **Figure V.6** indique les règles pour faire face à des commande ou l'attribut begin ou end est lié à un événement interactif.

Reprenons l'exemple d'une entrée à partir du clavier:

$\{A'\} \langle \text{cmd } c \text{ begin} = \text{'accesskey's'+K1} \text{ end} = \text{'K2'} \rangle \dots \langle / \text{cmd} \rangle \{B\}$

Ou  $A' = \{ A \cup \{t_{cr}(c) = \text{init}\} \cup \{\text{time}(\text{accesskey's'}) = \text{keyin}\} \}$ .

Ces règles indiquent que l'élément doit être évaluée en se référant à l'instant de temps où l'événement se produit, c'est-à-dire, si  $\{\text{time}(\text{accesskey's'}) = \text{keyin} \in A'$  nous pouvons prouver la validité de l'élément c si nous pouvons prouver que :

$\{ A \cup t_{cr}(c) = \text{keyin} \} \langle \text{cmd } c \text{ begin} = \text{K1} \text{ end} = \text{'K2'} \rangle \dots \langle / \text{cmd} \rangle \{B\}$

L'entrée à partir du clavier doit se produire après l'évaluation de la déclaration, représentée par la valeur 'init' ou après son début si 'accesskey' est défini dans l'attribut fin de la commande.

La justesse de cette règle découle des considérations suivantes:

considérer un état initial  $S_0$  de telle sorte que la précondition  $A'$  retenue :  $S_0 \models \{ A \cup \{tcr(c) = init\} \cup \{time(accesskey's') = keyin\} \}$ . Donc  $S_0\_c = init$ , et  $init \leq keyin$  et il existe un état intermédiaire  $S_i$  tel que  $S_i\_c = keyin$  et  $Fe(accesskey's') = keyin$  Ainsi, par la monotonie,  $S_i \models \{ A \cup \{tcr(c) = keyin\} \}$  puis, par la validité des prémisses, nous avons que le lecteur atteint un état final  $S_f$  qui satisfait la postcondition  $B$ . Dans ce cas particulier  $\{begin(c) = keyin + K1\}$  inclus dans  $B$ . ce qui signifie que  $c$  commence exactement  $k1$  instants de temps après la survenance de l'événement  $accesskey's'$ , c'est à dire, notre règle respecte les spécifications standard.

Notez que tous les autres événements interactifs pris en charge par les spécifications SMIL (une liste partielle peut être trouvée dans **Tableau V.1**) pourraient être abordée de la même manière puisque le lecteur enregistre l'instant de temps dans laquelle l'événement se produit au moyen de la fonction  $Fe()$ ...au titre d'exemple,  $activateEvent$  représente l'instant de temps dans lequel un utilisateur clique sur un élément médias et, de notre point de vue, il n'est pas différent de l'utilisateur cliquant sur le clavier. Aussi, dans ce cas, la seule contrainte est que l'événement doit avoir lieu après l'évaluation de la commande pour être utile.

### V.8.7 Preuve règle pour de multiples exécutions d'un même élément

Un événement interactif peut se produire plus d'une fois, l'utilisateur peut cliquer plusieurs fois sur un bouton. Cela signifie que un objet qui se lie à son début à cet événement peut jouer plus d'une fois. Pour enregistrer toutes les exécutions d'un élément, nous considérons plusieurs lectures du même élément avec des nouveaux noms.

BEGIN+MULTIPLE-PLAYBACKS

$$\forall i 0 \leq i \leq n \{ A \cup \{tcr(\nu^i(c)) = start_i\} \} \langle cmd \nu^i(c) \text{ begin}='k1' \text{ end}='k2' \rangle \dots \langle /cmd \rangle \{ B'_i \}$$


---


$$\{ A' \} \langle f \rangle \langle cmd c \text{ begin}='event+k1' \text{ end}='k2' \rangle \dots \langle /cmd \rangle \langle /f \rangle \{ \bigcup_{i=1}^n B_i \}$$

where  $f \in \{par, excl\}$

$$A' \stackrel{def}{=} A \cup \{tcr(c) = init\} \cup \{sequence\}$$

$$sequence \stackrel{def}{=} times(event) = (start_0, \dots, start_n)$$

$$B'_i = \begin{cases} B_i \setminus \{end(\nu^i(c)) = h\} & \text{if } (begin_B(\nu^{i+1}(c)) = h) \\ B_i & \text{otherwise} \end{cases}$$

APPLICABILITY CONDITION:

$$\forall (i, j) i \leq j \implies (start_i \leq start_j)$$

$$\wedge \{ A \cup \{tcr(c) = init\} \cup \{sequence\} \} \implies \forall i (start_i \geq tcr(c))$$

$$\wedge \forall (i, j) (begin_B(\nu^i(c)) \leq begin_B(\nu^j(c))) \implies (end_B(\nu^i(c)) \leq begin_B(\nu^j(c)))$$

**Figure V.7** La règle Begin + Multiple-Playbacks

La règle BEGIN+MULTIPLE-PLAYBACKS modélises cette situation : l'exécution multiple d'un même élément est correcte dès que chaque lecture d'une seule est correcte. Nous notons que si un événement se produit avant la fin des médias alors il est immédiatement arrêté et redémarré. Dans ce cas, nous n'avons pas besoin de montrer instant de fin de la lecture dans les prémisses. Prenons l'exemple précédent où les utilisateurs

cliquer sur le chiffre «s» deux fois, à l' instant de temps  $start1 < start2$  et  $A' = \{A \cup \{times(\text{accesskey's'}) = (start1, start2)\}\}$ . Nous pouvons appliquer la règle pour prouver :

$\{A'\} < \text{cmd } c \text{ begin} = \text{'accesskey's'+K1} \text{' end} = \text{'K2'} > \dots \dots \dots < / \text{cmd} > \{B\}$

Si nous pouvons prouver la validité de deux exécutions de l'élément c :

$\{A1\} < \text{cmd } c \text{ begin} = \text{'accesskey's'+K1} \text{' end} = \text{'K2'} > \dots \dots \dots < / \text{cmd} > \{B1\}$

$\{A2\} < \text{cmd } c2 \text{ begin} = \text{'accesskey's'+K1} \text{' end} = \text{'K2'} > \dots \dots \dots < / \text{cmd} > \{B2\}$

Où c2 est un nouveau nom pour la deuxième exécution de c,  $A1 = \{A \cup \{tcr(c) = start1\}\}$  et

$A2 = \{A \cup \{tcr(c2) = start2\}\}$

Si l'événement interactif s'apparaît dans la définition de l'attribut end, nous avons besoin des considérations différentes.

A la différence du cas d'un événement défini dans *Bégin*, plusieurs occurrences d'un événement défini dans *end* ne causer pas de multiples lectures d'un élément, lecteur finira l'élément dès que l'événement se terminant se produit pour la première fois, occurrences ultérieures du même événement sont ignorés.

La règle *END+USER-INTERACTION* peuvent être utilisés dans ce cas, depuis n'importe quel état qui satisfait le premier élément d'une séquence d'événements, satisfait, par définition, également la séquence. La validité de la règle découle du fait que les occurrences suivantes du même événement doivent être ignorées.

### V.8.8 Règle de Preuve pour les éléments qui contiennent des événements interactifs à la fois dans *Begin* et *End*

**BEGIN+END+MULTIPLE-PLAYBACKS**

$$\frac{\{A'\} < \text{cmd } c \text{ begin} = \text{'k1'} > \dots < / \text{cmd} > \{B'_1\} \quad n > 1 \implies \{A''\} \nu(c) \{B''\}}{\{A''\} < \text{cmd } c \text{ begin} = \text{'event1+k1'} \text{ end} = \text{'event2+k2'} > \dots < / \text{cmd} > \{B\}}$$

where

$$A' \stackrel{def}{=} A \cup \{t_{cr}(c) = start_1\}$$

$$A'' \stackrel{def}{=} A \cup \{t_{cr}(c) = init\} \cup \{starts\} \cup \{stops\}$$

$$starts \stackrel{def}{=} times(event_1) = (start_1, \dots, start_n)$$

$$stops \stackrel{def}{=} times(event_2) = (stop_1, \dots, stop_m)$$

$$B \stackrel{def}{=} B_1 \cup B''$$

$$B'_1 = \begin{cases} B_1 \setminus \{end(c) = h\} & \text{if } (begin_B(\nu(c)) = h) \\ B_1 & \text{otherwise} \end{cases}$$

$$starts' \stackrel{def}{=} times(event_1) = (start_2, \dots, start_n)$$

$$stops' \stackrel{def}{=} \begin{cases} times(event_2) = (stop_2, \dots, stop_m) & \text{if } (end_B(c) = stop_1 + k_2) \\ stops & \text{otherwise} \end{cases}$$

$$A'' = A \cup \{t_{cr}(\nu(c)) = init\} \cup \{starts'\} \cup \{stops'\}$$

APPLICABILITY CONDITION:

$$m \leq n \wedge (\forall(i, j) \ i \leq j \implies start_i \leq start_j) \wedge (\forall(i, j) \ i \leq j \implies stop_i \leq stop_j)$$

$$\wedge A' \cup B \implies (stop_1 \geq begin_B(c) \wedge ((end_B(c) = stop_1 + k_2) \vee (end_B(c) = begin_B(\nu(c))))))$$

**Figure V.8** La règle *Begin + End + Multiple-Playbacks*

Un cas particulier concerne les médias qui contiennent des événements interactifs dans "Begin" et "End».

La règle BEGIN+END+MULTIPLE-PLAYBACKS étend le cas lorsque l'attribut "Bégin" a un événement unique, et pourrait être utilisé pour prouver la validité de chaque exécution de l'élément. Les seules différences pertinentes figurent dans la condition d'application: les deux séquences doivent être ordonnées et la séquence de "stars" doit avoir une longueur égale ou supérieure à la séquence «end ». En outre, le dernier événement doit être un événement d'arrêt.

### V.8.9 Gestion des événements non interactifs

Une autre possibilité offerte par la norme SMIL est de lier l'événement "Bégin" (ou end) d'un (groupe de) élément média m avec l'événement "Bégin" (ou end) d'autre (groupe de) élément média n.

Les événements non interactifs sont enregistrés différemment dans l'état du lecteur, par ce que ils ne sont pas suivis par la fonction  $F_e()$  mais par la fonction  $F_d()$ .

Par exemple, examiner les éléments :

```

<par id="p" end="au.end">
  <audio id="au" />
  <text id="tx" />
</par>
(1)

```

```

<cmd id="m" begin="n.begin+5"/>
(2)

```

Dans le cas (1), l'instruction "par" se termine lorsque le média au termine, dans le cas (2) l'élément médias m commence 5 s après le début de n.

Élément (1) peut être traité comme dans le cas d'événements interactifs : si nous savons déjà (à partir de la prémisse) la fin de au, exemple, end (au) = stop alors nous pouvons analyser l'élément < par id='p' end = 'stop' > ..... </ par> par conséquent, la règle suivante peut être appliquée au cas (1).

PAR+END+EVENT

$$\{A\} \langle \text{par } c \text{ end} = 'stop \pm k' \rangle c_1 .. c_n \langle / \text{par} \rangle \{B\}$$

$$\{A\} \langle \text{par } c \text{ end} = 'c_i . \text{end} \pm k' \rangle c_1 .. c_n \langle / \text{par} \rangle \{B\}$$

APPLICABILITY CONDITION:

$$\text{end}_B(c_i) = \text{stop} \wedge \text{begin}_B(c) \leq \text{stop} \pm k$$

La situation est plus complexe dans le cas (2), qui ne peut être analysé singulièrement depuis son évaluation besoin d'informations sur le "Bégin" attribut d'objet média n. Pour cette raison, nous devons considérer un ensemble d'éléments média comme indiqué par la règle suivante:

BEGIN+EVENT

$$\{A\} \langle \text{cmd } c \rangle \dots c_i \dots c'_j \dots \langle / \text{cmd} \rangle \{B\}$$

$$\{A\} \langle \text{cmd } c \rangle \dots c_i \dots c_j \dots \langle / \text{cmd} \rangle \{B\}$$

où  $c_i$ ,  $c_j$  et  $c_j'$  sont liées comme suit: il existe  $n$ ,  $m$  tel que  $n \in \text{Closure}(c_i)$  et

$m = \langle \text{cmd id} = 'm' \text{ begin} = 'n.\text{begin} + K' \rangle \dots \dots \langle / \text{cmd} \rangle$

$c_j'$  est obtenu à partir  $c_j$  en remplaçant  $m$  avec :

$m = \langle \text{cmd id} = 'm' \text{ begin} = 'begin(n) + K' \rangle \dots \dots \langle / \text{cmd} \rangle$

Une attention particulière est nécessaire si il ya plusieurs exécutions de l'élément média  $n$  raison des interactions de l'utilisateur. Par exemple, une  $v$  vidéo activé par un clic sur une image  $i$  et sa bande son associé a ' $a$ ' :

```
<par>
  <img id = "i" />
  <video id = "v" begin="i.activateEvent"/>
  <audio id = "a" begin="v.begin"/>
</par>
```

Dans ce cas, même la bande-son doit être jouée à plusieurs reprises. Si nous essayons de prouver la validité de cette composition parallèle, nous avons besoin d'analyser chaque composant. en appliquant la règle BEGIN+MULTIPLE-PLAYBACKS à la vidéo  $v$ , mais nous considérons aussi l'ensemble des exécutions associés de  $a$ ,  $\{a, a_2, a_3, \dots, a_n\}$ . Pour obtenir cela, nous devons affiner la règle BEGIN+EVENT en exigeant que  $c_j'$  est obtenu à partir de  $c_j$  en élargissant  $m$  avec toutes ses exécutions  $V_i(m)$ , où chaque  $V_i(m) = \langle \text{cmd } V_i(m) \text{ begin} = 'begin(V_i(n) + K)' \rangle \dots \dots \langle / \text{cmd} \rangle$

est associée à une lecture de  $n$ , comme indiqué dans la post-condition: quel que soit  $(i) \{begin(V_i(n)) = \text{value } i\} \in B$

**V.8.10 Règle de preuve pour la composition parallèle lorsque l'attribut « EndSync » est égal à 'First'**

<p>PAR+ENDSYNC=FIRST</p> $\frac{\{A\} \langle \text{par } c \text{ begin}='k1' \text{ end}='stop' c_1 \dots c_n \rangle \{B'\}}{\{A\} \langle \text{par } c \text{ begin}='k1' \text{ endsync}='first' c_1 \dots c_n \rangle \{B\}}$ <p>where <math>B = B' \setminus \bigcup_{i \neq j} \{end(c_i) = e_i\}</math></p> <p>APPLICABILITY CONDITION:</p> $\exists j   ((end(c_j) = \min_{c_i} \{end_{B_i}(c_i)\}) \wedge stop = end(c_j))$
---

**Figure V.9** La règle PAR + EndSync = First.

On considère un conteneur parallèle avec un attribut « endsync » égale à «First », La règle PAR+Endsync = 'First' traduit l'élément par à la même composition avec une valeur correcte de l'attribut 'end', c'est-à-dire, le premier instant de temps dans laquelle un enfant se termine.



V.8.11 Règle de preuve pour l'attribut « MIN »

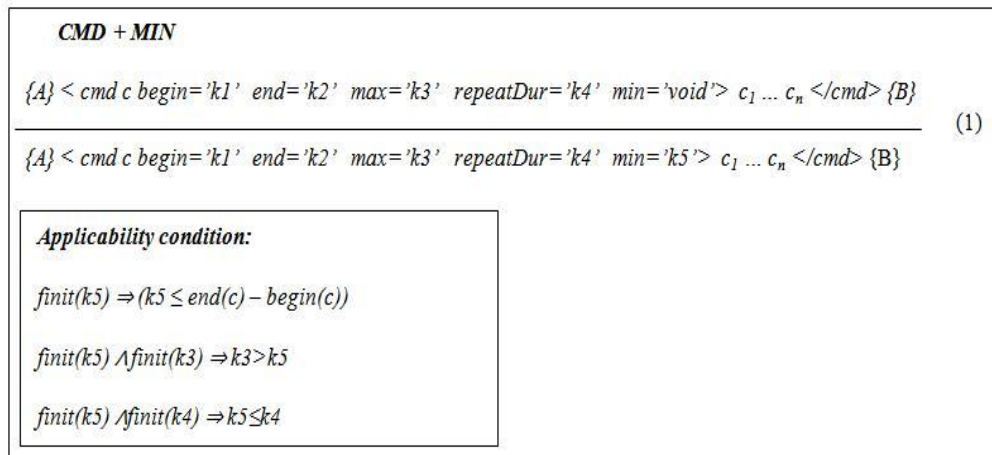


Figure V.10 La règle CMD + MIN

V.8.12 Règle de preuve pour l'attribut « MAX »

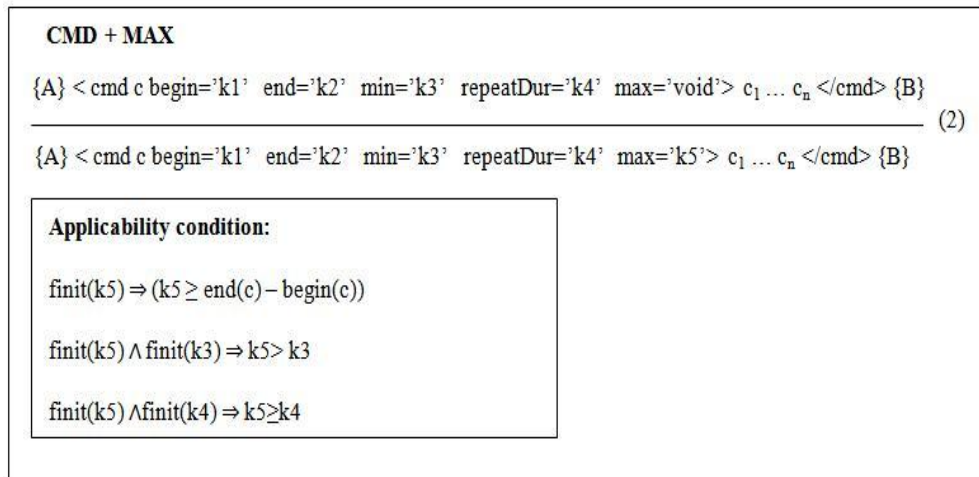


Figure V.11 La règle CMD + MAX

V.8.13 Règle de preuve pour l'attribut « RepeatDur »

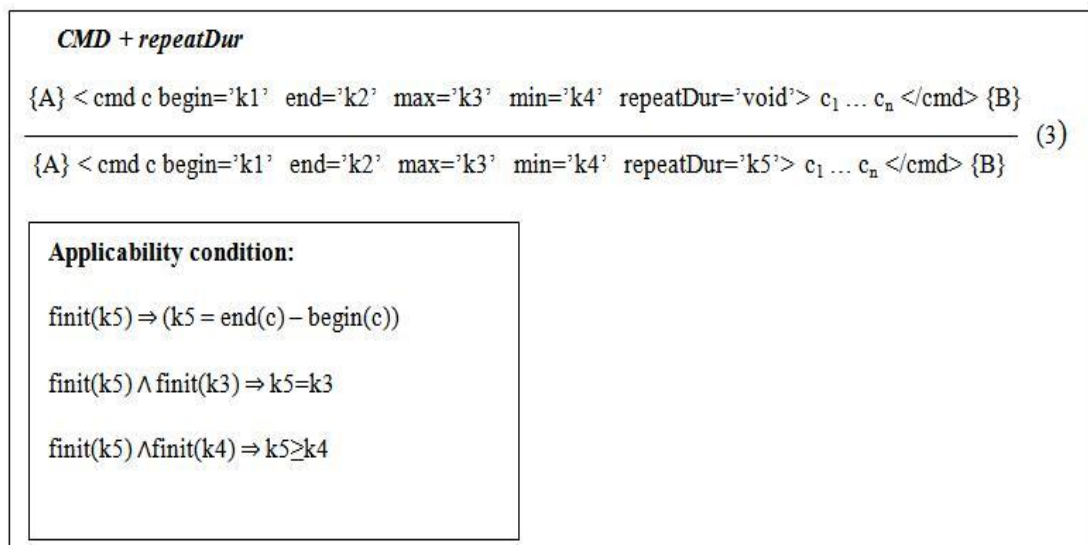


Figure V.12 La règle CMD + RepeatDur

## V.9 Implémentation

Dans ce mémoire, nous avons proposé une nouvelle approche de validation des documents SMIL basée sur la logique de Hoare, que nous avons implémenté en utilisant le langage Java. En effet, nous avons conçu et réalisé un outil formel de validation de SMIL.

## V.10 Environnement de développement JBuilder

Java est un langage objet basé sur le langage C++ permettant le développement d'applications avec des fonctionnalités qui rendent la programmation plus simple et plus sûre. Le langage Java est interactive sur le web (applet) gère les formats textes, XML, images et bien sur d'autres formats.

Un programme Java est portable, il peut s'exécuté sur des ordinateurs fonctionnant sous différents systèmes d'exploitation (par exemple Windows, Linux, Solaris, UNIX). Java est aussi portable d'une plate-forme à une autre sans recompilation. Le compilateur produit un langage intermédiaire appelé « bytecode » interprété par la machine virtuelle de Java (JVM) sur différentes machines. JAVA est un langage robuste qui peut être exploité pour développer un large éventail de programmes utilisant une interface utilisateur graphique, pouvant être appliqués en réseau et se connecter à des bases de données, et offrant d'autres fonctionnalités toutes plus sophistiquées les unes que les autres. La richesse du langage Java tient aussi à sa bibliothèque de classes (API : Application Programming Interface) qui permet de traiter des applications très divers (interfaces graphiques, chaînes de caractères, images, sons, mathématiques, etc.). Avec JAVA et ses bibliothèques graphiques AWT (Abstract Window Toolkit) et Swing, la conception d'interfaces est aisée, les applications sont agréables.

JBuilder 2006 est l'un des plus puissant environnement de développement, qui est un environnement professionnel pour le développement de solutions Java, intègre tous les concepts d'ingénierie moderne, WebServices, XML, Travail collaboratif, plugin Mobilset pour écrire des applications pour terminaux mobiles, conception d'EJB, JSP/Servlet et Struts, Ant, etc.

De même que pour les autres produits Borland, JBuilder offre CodeInsight, pour facilement voir apparaitre les méthodes et les paramètres, c'est le grand confort. L'éditeur de code Java est très puissant et très bien fait, et il y a des modèles de code déjà préparés.

La capacité de modélisation UML dans JBuilder favorise l'utilité de développer avec cet environnement de développement. JBuilder supporte aussi la réutilisation des packages externes parmi ces packages ceux qui aident à la manipulation des fichiers XML, RDF, RDFS, OWL et OWLS, ce qui répond à nos besoins.

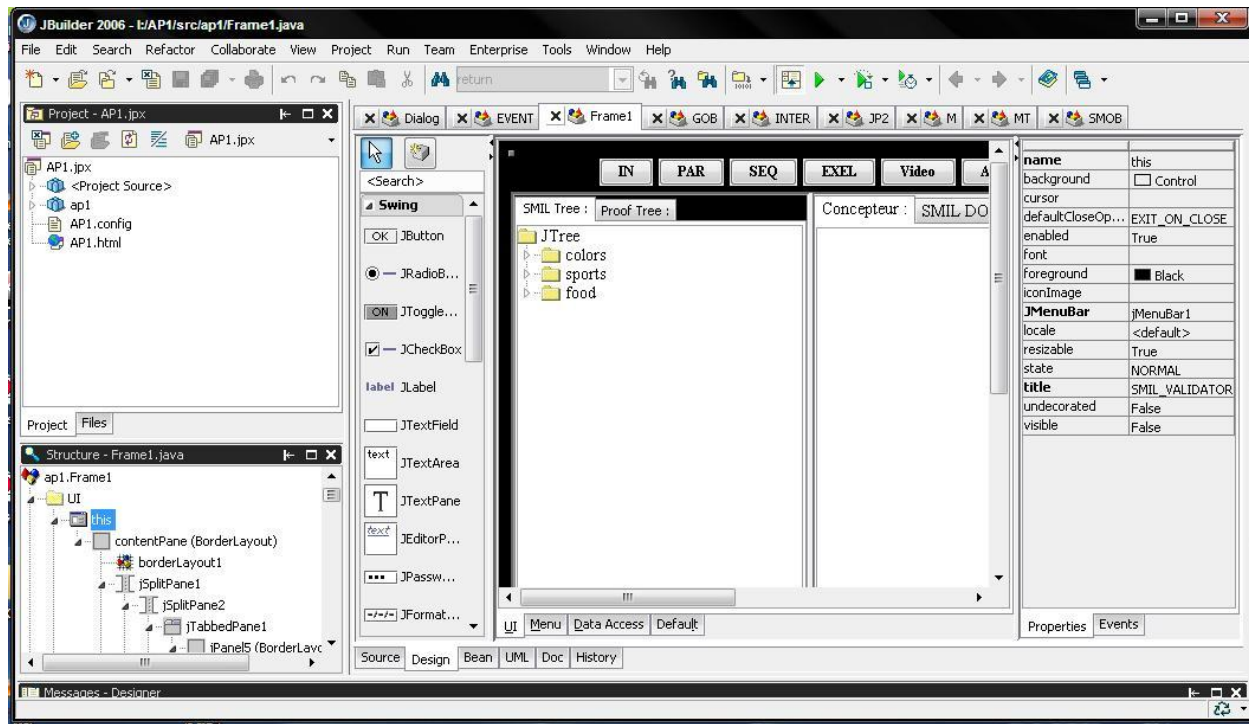


Figure V.13 : Environnement JBulider 2006

Notre outil réalisé de validation des documents SMIL est illustré dans la figure IV.14 ci-dessous :

### V.11 Architecture générale de l'outil

L'architecture générale de notre outil réalisé est illustrée par la figure ci-dessous :

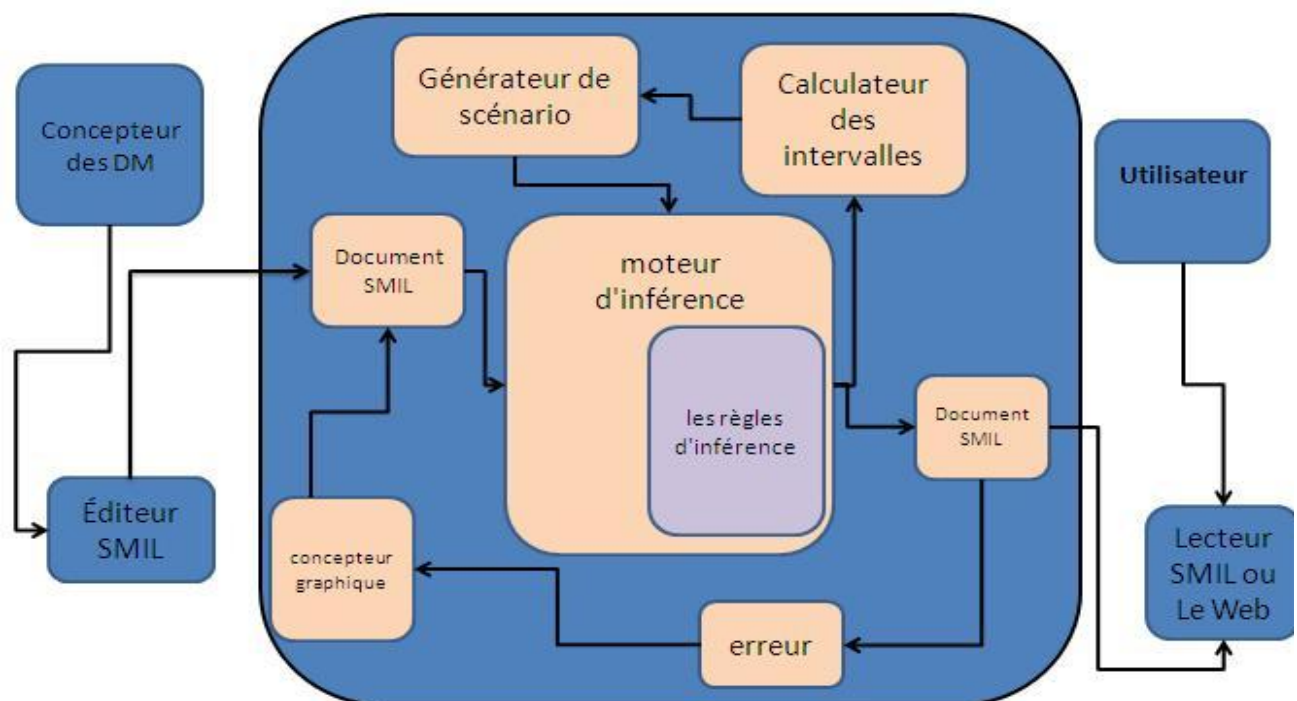


Figure V.14 Architecture générale de notre outil

La figure IV.14 illustre l'architecture générale du système: l'outil a été mis en place pour être utilisé en combinaison avec toute interface de création d'éviter de forcer les concepteurs à changer leurs préférences.

- **Le concepteur des DMM:** faire l'Édition d'un document dans un éditeur SMIL (LeemSee 2, .....), l'éditeur de texte ou bien le concepteur graphique intégrer dans notre outil.

- **Le concepteur graphique :** pour faire la conception graphique d'un document SMIL pour faciliter le travail.

- **Le moteur d'inférence :** utiliser les règles d'inférence pour raisonner et construire l'arbre de preuve, Si le document contient un conflit, la construction de la preuve échoue parce que l'un des prémisses nécessaires ne peuvent pas être prouvées ou les conditions d'application ne sont pas satisfaits et le système retourne l'élément qui contient l'erreur et donne des suggestions sur la façon de corriger..

**Les règles d'inférences :** Toutes les règles sémantiques introduites dans ce chapitre sont traduits en méthodes JAVA dans une seule class, qui montrent comment la condition est modifiée pour obtenir le post condition.

**Calculateur des intervalles :** le module calculer les intervalles valides pour chaque événement pour aider l'utilisateur à éviter la mauvaise définition des événements dans les attributs 'begin' ou 'end',

**Générateur des scenarios:** générer automatiquement des séquences d'événements utilisateur pour tester les documents SMIL.

### V.12 Conception UML

Les objets utilisés dans notre outil sont illustrée par la figure ci-dessous :

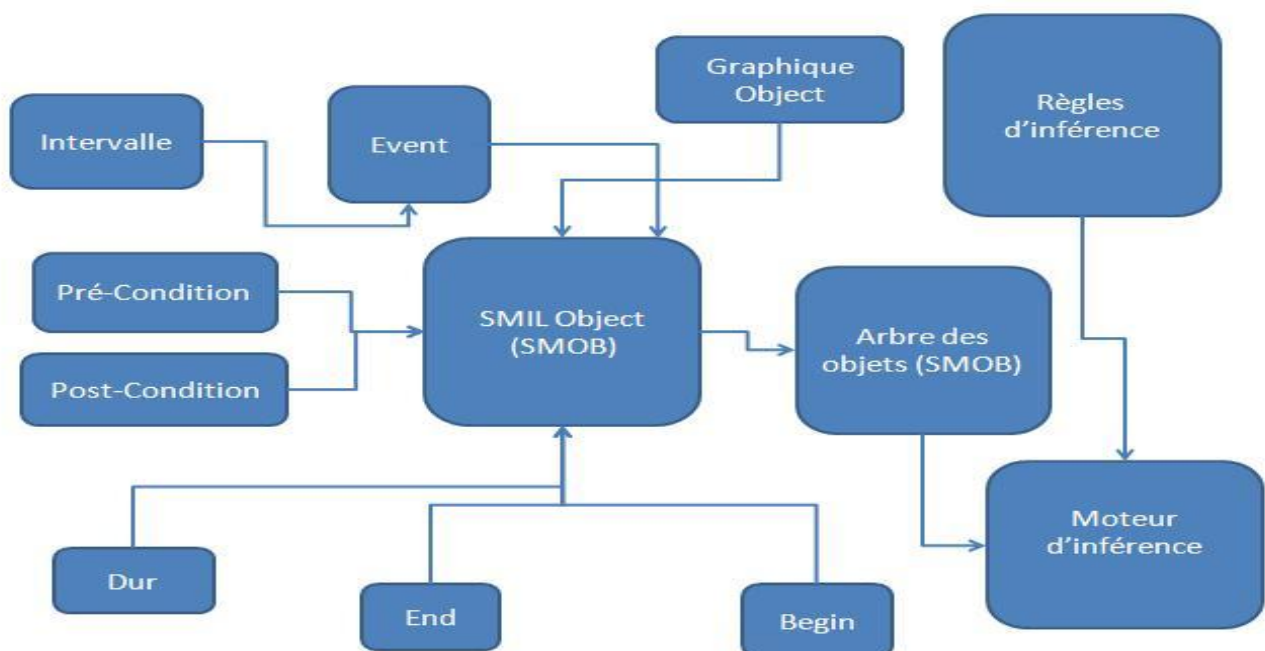


Figure V.15 Diagramme des objets

La classe SMOB représente un objet media ou un conteneur temporelle.

La classe ATT représente les attributs complexe begin et end de chaque objet SMOB.

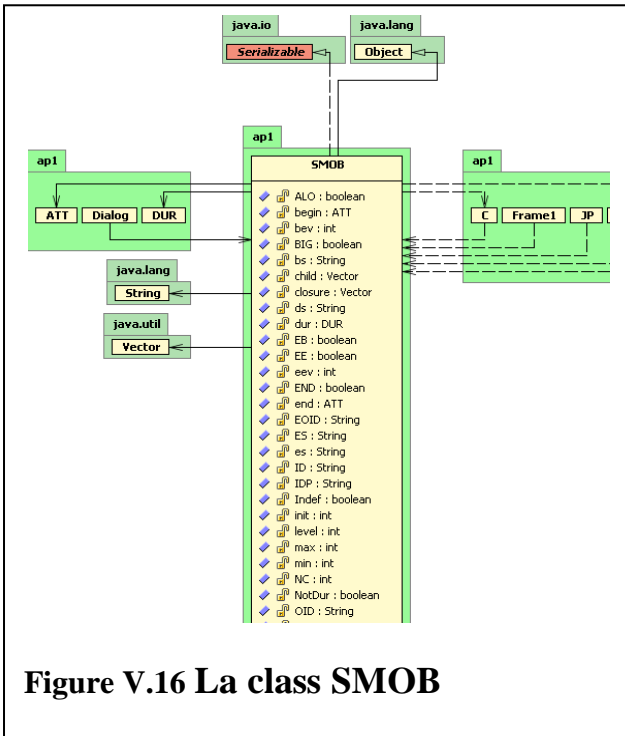


Figure V.16 La class SMOB

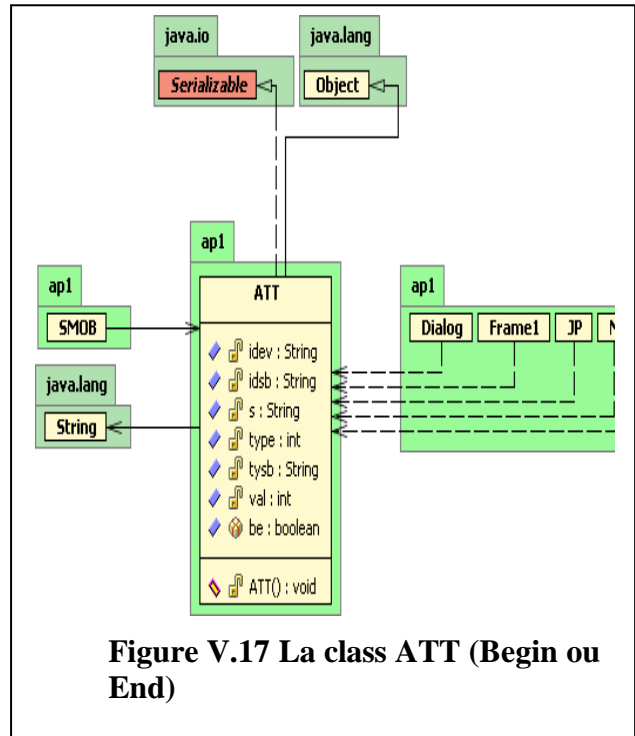


Figure V.17 La class ATT (Begin ou End)

La class C représente le pré / post condition, chaque objet SMIL comporte deux vecteur la premier pour les pré-conditions, la deuxième pour les post-conditions.

La class M regroupe des méthodes, ou chaque méthode représente une règle d'inférence.

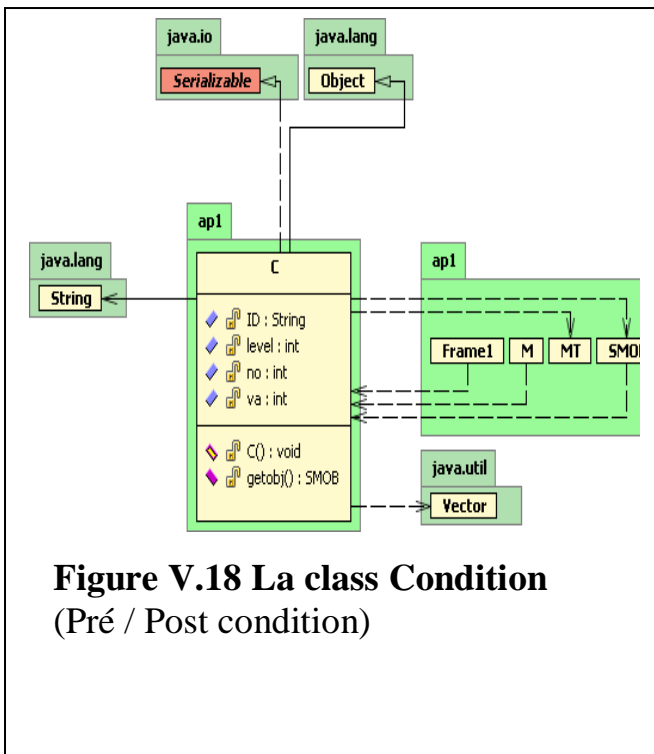


Figure V.18 La class Condition (Pré / Post condition)

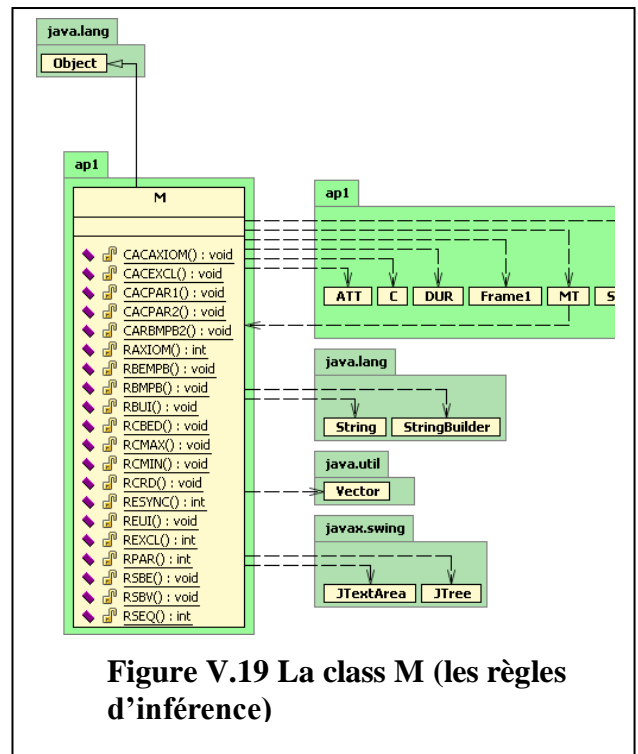
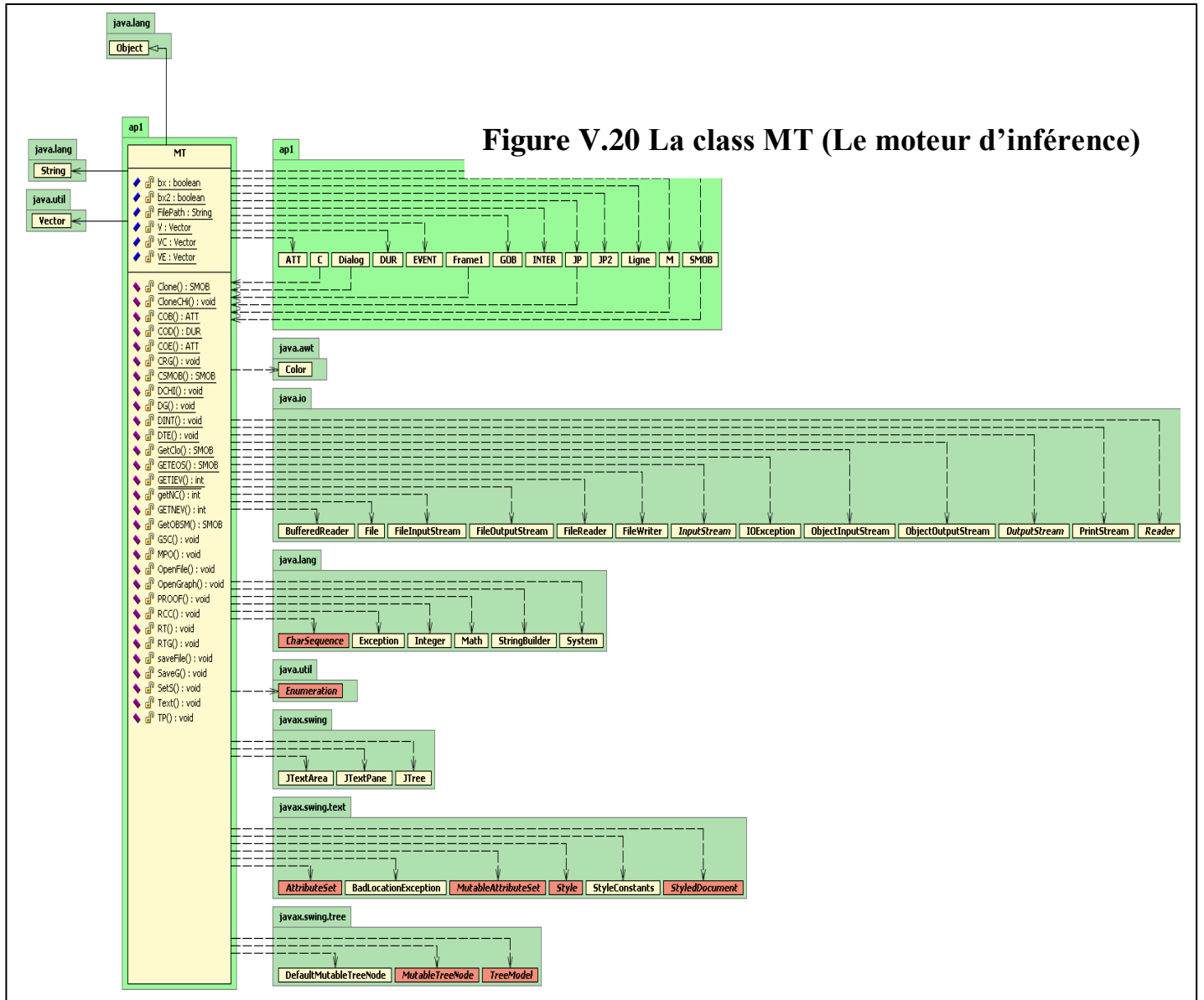
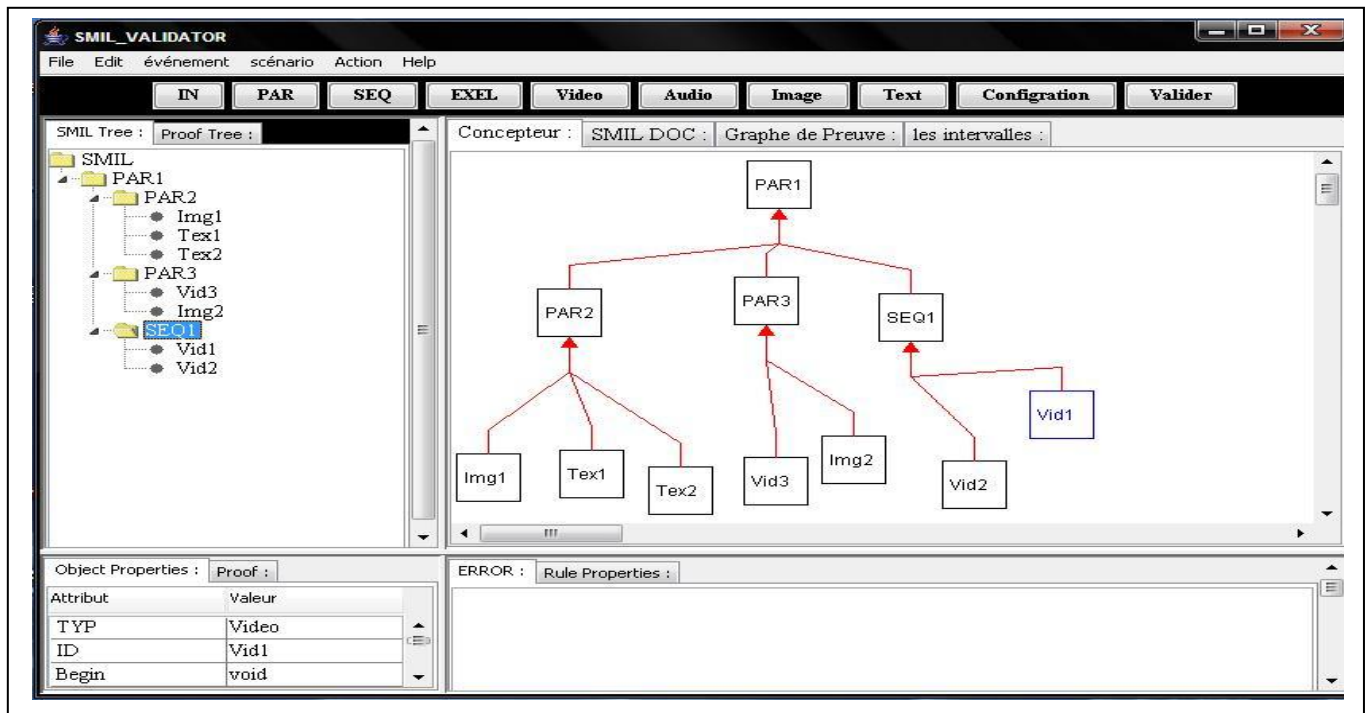


Figure V.19 La class M (les règles d'inférence)

La class MT regroupe tous les méthodes utilisés pour construire le moteur d'inférence son rôle est raisonner et construire l'arbre de la preuve.



Comme c'est illustré dans la **figure V.21**, l'interface graphique de notre outil de validation de SMIL.



**Figure V.21** L'outil SMIL\_Validator.

### V.13 Captures d'écrans de l'application

Barre de menu

#### Menue File :

Open File : ouvrir un fichier.

Save File : sauvegarder un fichier.

Save as : sauvegarder un fichier dans emplacement spécifique.

Exit : quitter l'application.

#### Menu Edit :

Undo : annuler la dernière action.

Redo : retour a la dernière action.

Copy : copy code SMIL

Past : coller code SMIL.

#### Menu Event :

Calculat Intevale : calculer les intervalles des événements.

Show Event Tree : voir l'arborescence des événements.

4-Menu Scénario :

Manuel : configuration manuel des scenarios.

Automatique : génération automatique des scenarios.

#### Action :

First : aller a la premier règle utiliser.

Next : aller a la règle suivante.

Previous : retour a la règle précédente.

Last : aller a la dernière règle utilisée.

**Help :**

Help Topics : l'aide du logiciel.

About SMIL\_Validator : des informations sur SMIL\_Validator

**Barre d'outil :**

Bouton IN : définir l'imbrication des objets.

Bouton PAR : ajouter un conteneur PAR.

Bouton SEQ : ajouter un conteneur SEQ.

Bouton EXCL : ajouter un conteneur EXCL.

Bouton VIDEO : ajouter un Vidéo.

Bouton IMAGE : ajouter Image.

Bouton TEXT : ajouter Texte.

Bouton Valider : Lancer la validation.

**Les panneaux :**

Panneau SMIL TREE : afficher la structure de document SMIL en arbre.

Panneau Proof Tree : afficher les règles du preuve en arbre .

Panneau Concepteur : pour modéliser le document SMIL graphiquement.

Panneau SMIL DOC : afficher le document SMIL en texte.

Panneau Event Tree : afficher l'arbre des événements.

Panneau Proof Graph : afficher le graphe du preuve.

Panneau Object properties : modifier les propriétés d'un objet SMIL.

Panneau Event properties : afficher les propriétés d'événement.

Panneau ERROR : afficher les erreurs.

Panneau Rule properties : afficher les propriétés d'une règle.



### Conclusion Générale & Perspectives

Les nouvelles technologies dans les systèmes informatiques (les smartphones, les tablettes, les équipements numériques,.....) et les systèmes de communication (3G++,4G, WIMAX, UMTS,.....) touchent la vie des millions de personnes chaque jour au tour du monde, ces technologies nécessitent une nouvelle génération d'applications et de données numériques qui doivent être conviviales et expressives.

Parmi les données numériques on trouve les documents multimédias qui deviennent de plus en plus répandus et utilisés dans différents domaines comme : le téléenseignement, la visioconférence, la publicité, la création des sites web, L'infographie,..... Etc.

Ces DMM sont caractérisés par leur structure spatiale (présentation graphique et mise en page), de leur structure logique (organisation en chapitres, sections, paragraphes, etc.) et de leur structure sémantique (hypertexte). 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, interaction de l'utilisateur...).

Les documents étudiés ici sont donc multimédias, temporisés et interactifs. L'introduction de la dimension temporelle et spatiale dans les documents conduit nécessairement à de nouveaux standards pour la conception et l'édition des DMM : comme XML, MHEG, ODA .....SMIL, ....etc.

Le standard SMIL est le plus utilisé à l'heure actuelle pour la synchronisation et la présentation des DMM complexes, mais ne garantit pas leur qualité de présentation.

La complexité de la synchronisation temporelle et spatiale des documents SMIL est telle qu'il est nécessaire de développer des techniques de modélisation et validation formelle pour garantir la validité d'un scénario. Un modèle formel pour les documents multimédias doit répondre, en plus des besoins classiques des systèmes multimédias.

La logique de Hoare qui est basée sur des axiomes et des règles d'inférence offre un cadre formel au document SMIL pour valider leurs éléments temporels.

Dans le cadre de ce mémoire nous sommes intéressés à la modélisation et la validation des documents SMIL par la logique de Hoare au moyen d'un ensemble de règles d'inférence. Nous avons présenté un outil d'aide permettant à l'utilisateur de produire des documents SMIL valides. Notre outil détecte automatiquement les conflits temporels dans les documents SMIL et permet à l'utilisateur de les corriger en lui affichant des messages utiles. L'outil est basé sur une sémantique formelle définissant les aspects temporels de SMIL au moyen d'un ensemble de règles d'inférence qui décrivent comment l'exécution d'un morceau de code change l'état du calcul d'un lecteur.

La validité et la complétude de notre approche a été examinée selon une sémantique opérationnelle qui formalise les changements dans l'état d'un lecteur décrit de manière informelle dans la recommandation SMIL.

Les perspectives tirées de ce mémoire sont diverses. Nous développer une application pour notre proposition avec un prototype bien détaillé et d'évaluer l'apport de ce prototype sur des documents SMIL. Il serait intéressant de développer cette prototype en utilisons les codes sources java pour réaliser les perspectives suivants :

- Adapter la méthode pour vérifier la cohérence spatiale des documents SMIL.
- Utiliser d'autres attributs de SMIL pour étendre l'expressivité.
- Intégrer les contraintes sur le réseau.

## Bibliographie

- [Adib 96] M Adiba, " Bases de Données Multimédias", Université de Grenoble – Chapitre de livre, Edition Hermès, Mars 2000)
- [Alle 83] J.F. Allen, "Maintaining knowledge about Temporal Intervals". Dans *Communication ACM*, Vol. 26, n°. 11, p. 837-843, novembre 1983.
- [Babe 93] K. Böhm, K. Aberer. An Object-Oriented Database Application for HyTime Document Storage. Technical report, GMD-IPSI, 1993
- [Blak 96] BLAKOWSKI G. et STEINMETZ R., "A Media Synchronisation Survey: Reference Model, Specification, and Case Studies", *IEEE Journal Of Selected Areas In Communications*, 14(1), pp. 5-34, janvier 1996.
- [Boss 07] Bossi, A., Gaggi, O.: Enriching SMIL with assertions for temporal validation. In: *Proceedings of ACM MM*, pp. 107–116, September 2007
- [Boss 11] Bossi, A., Gaggi, O: Analysis and verification of SMIL documents. *Multimedia Syst.* 17(6): 487-506 (2011)
- [Bouy 07] Bouyakoub, S., Belkhir, A.: Formal design of smil documents. In: *Proceedings of WEBIST*, 2007
- [Buch 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.
- [Buch 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.
- [Buford 94] J. F. Koegel Buford. *Multimedia Systems*, Edition : ACM Press, 1994.
- [Bult 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.
- [Bult 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.
- [Carcone 97] L. Carcone. *Formatage Spatial dans un Environnement d'Edition/Présentation de Documents Multimédia*, Mémoire CNAM-Grenoble, Décembre 1997.
- [Ccit 92] CCITT Press Release on MHEG, December 1992.
- [Chen 95] CHEN Z., TAN S., CAMPBELL R. et LI Y. , "Real Time Video and Audio in the orld Wide Web", *Proceedings of the Fourth International Conference on the World Wide Web* , Boston, MA, USA, décembre 1995.
- [Chun 03] S.M. Chung and L.P. Anil . "Timed Petri Net Representation of the Synchronized Multimedia Integration Language (SMIL) of XML", In *Proc 2003 International Symposium on Information Technology (ITCC 2003)*, 2003, pp 711-716.
- [Dani 91] Danish Standards Association. SGML-ODA, Présentation des concepts et comparaison fonctionnelle. afnor technique, 1991.
- [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.
- [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
- [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.
- [Flanck 02] S. Flanck. *Multimedia Technology in Context* , *IEEE Multimedia*, Juillet - Septembre 2002.
- [Gaff 02] F. Gaffiot. *Le Grand Gaffio*, Dictionnaire Français-Latin. Hachette, 2000.
- [Gerevini 02] A. Gerevini & N. Bernhard. *Qualitative Spatio-Temporal Reasoning with RCC-*

- and Allen's Interval Calculus: Computational Complexity*, Proceedings : the Conference ECAI'02, pp. 312-316, Lyon, France, 21-26 Juillet 2002.
- [Gold 81] Goldfarb C., "A Generalized Approach to Document Markup", ACM SIGPLAN-SIGOA Symposium on Text Manipulation, Portland, Oregon, USA, June 1981, pp. 68-73.
- [Halasz 90] F. Halasz & M. Schwarz. *The Dexter Hypertext Reference Model*, NIST hypertext Standardization Workshop, Gaithersburg, MD, 16-18 Janvier 1990.
- [Hamb 72] C. L. Hamblin. Instants and Intervals. In Proceedings of the 1st Conference of the International Society for the Study of Time, pages 324–331, New York, 1972.
- [Hard 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.
- [Hanappe 99] P. Hanappe. *Design and Implementation of an Integrated Environment for Music Composition and Synthesis*, Thèse de Doctorat, Université Paris 6, Avril 1999.
- [Hoar 69] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. of the ACM*, 12(10):576–585, 1969.
- [Hosc 98] P. Hoschka, Recommandation du W3C, "Synchronized Multimedia Integration Language (SMIL) 1.0". En ligne à: T, 15 juin 1998. <http://www.w3.org/TR/REC-smil>
- [Huij 02] Yu, H., He, X., Gao, S., Deng, Y.: Modeling and analyzing SMIL documents in SAM. In: MSE, pp. 132–135. Newport Beach, California, December 2002
- [ISO 86] International Organization for Standardization (ISO), Information Processing –Text and Office Systems – Standard Generalized Markup Language (SGML), ISO 8879- 1986.
- [ISO 92] ISO/IEC 10744:1992. Hypermedia/Time-based structuring language (HyTime). Geneva, 1992. Revised edition (ed. 2), 1997.
- [Jour 01] Jourdan, M.: A formal semantics of SMIL: a web standard to describe multimedia documents. *Comput. Standards Interfaces* 23(5), 439–455 (2001)
- [Jour 04] M. Jourdan, N. Layaïda, C. Roisin. Le temps dans les documents. *Techniques de l'Ingénieur*, 2004.
- [Jour 99] M. JOURDAN, N. LAYAÏDA, et C. ROISIN "Le temps dans les documents". *Techniques de l'ingénieur*, 1999, Vol. 249.
- [Labe 07] I. LABED. " Méthodes et Outils pour la Construction de Scènes Multimédia Distribuées". Thèse de Doctorat, Université Mentouri – Constantine- Algérie, 08 Décembre 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.
- [Laya 96] Layaïda, S. (1996), « MADEUS: Un modèle de document multimédia structuré», *TSI*, vol. 15, num. 9, 1996.
- [Laya 97] N. Layaïda. *Madeus : Système d'Edition et de Présentation de Documents Structurés Multimédia*. Thèse de Doctorat, Université Joseph Fourier – Grenoble I. Juin 1997].
- [Leco 99] D. LECOMPTE, D. COHEN, P. DE BELLEFONDS, J. BARDA. "Les normes et les standards du multimédia XML, MPEG-4 et 7, MP3, HTML, Web3D... et les autres". Edition DUNOD – PARIS 1999.
- [Litt 90] T.D.C. Little and A. Ghafor. "Synchronization and storage models for multimedia objects", *IEEE journal on selected areas in communications*, Vol 8, 1990.
- [Little 93] T. D. C. Little & A. Ghafoor. *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.
- [Lecomte 00] D. Lecomte, D. Cohen, P. De Bellefonds & J. Barda. *Les Normes et les Standards du Multimédia*, Edition : Dunod (2eme), 2000.
- [Lo Presti 02] S. Lo Presti. *Langage de Spécification et de Description de Présentations Multimédias*. Thèse de Doctorat, Institut National Polytechnique de Grenoble, 27
- [Lu 96] G. Lu. *Communication and Computing for Distributed Multimedia Systems*, Edition : Artech House Publishers, 1996.
- [Mart 98] J. MARTINEZ. "Multimédia et bases de données". Proc. of the IEEE 5th Int'l Workshop on Multi-Media Data Base Management Systems (MMDBMS'98), Dayton, Ohio, 5-7 August 1998, pp. 38-45.
- [Maub 91] P. Maubourguet. *Grand Larousse Universel*, Edition : Larousse, 1991.

- [Mazo 06]. Mazouz, S., Dahamani, D., Kaddouri, L.: Formal approach for the coherence control of SMIL documents. *Int. J. Comput. Sci. Appl.* 3(2), 126–144 (2006)
- [MHEG 92] MHEG. "Information Technology - Coded Representation of Multimedia and Hypermedia Information Objects - Part1: Base Notation. Working Document S.7. ISO/IEC JTC1/SC29/WG12. November 1992.
- [Mich 98] Michard A., "XML langage et applications", Edition Eyrolles, Novembre 1998.
- [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.
- [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,
- [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
- [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.
- [Rois 94] ROISIN C. et VATTON I., "Merging Logical and Physical Structures in Documents", Proceedings of the Fifth International Conference on Electronic Publishing, Document Manipulation and Typography, pp. 327-337, Wiley Publishers, Darmstadt, Allemagne, avril 1994.
- [Rois 99] Roisin. (1999), « Documents Structurés Multimédia » Thèse HDR, 1999 à l'Institut National Polytechnique de Grenoble, France.
- [Samp 03] P.N.M. Sampaio. *Conception Formelle de Documents Multimédias Interactifs: Une approche s'Appuyant sur RT-LOTOS*, Thèse de Doctorat, Université Paul Sabatier, Toulouse III, France, 2003.
- [Samp 04] Sampaio, P.N.M., Courtiat, J.-P.: An approach for the automatic generation of RT-LOTOS specifications from SMIL 2.0 documents. *J. Braz. Comput. Soc.* 9(3), 39–51 (2004)
- [Sant 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..
- [Séna 96] SÉNAC (P.), DIAZ (M.), LÉGER (A.) et DE SAQUI-SANNES (P.). – Modeling logical and temporal synchronization in hypermedia systems. *IEEE Journal of Selected Areas on Communications*, vol. 14, n° 1, p. 84-103 (1996).
- [Sena 94] P. Sénac, M. Diaz, P. De Saqui-Sannes, "Toward a formal specification of multimedia synchronization", *Annals of telecommunications*, May /June 1994
- [Steinmetz 95] R. Steinmetz & K. Nahrstedt. *Multimedia: Computing, Communications & Applications*, Edition : Prentice Hall, 1995.
- [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>.
- [Wahl 94] T.Wahl, K. Rothermel. *Representing Time in Multimedia Systems*. In Proceedings of the International Conference on Multimedia Computing and Systems. Boston, Massachusetts. 1994, pp. 538-543.
- [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.
- [Yang 00] Yang C.C., "Detection of the Time Conflicts for SMIL-based Multimedia Presentations", Proceeding of 2000 International Computer Symposium (ICS2000)- Workshop on Computer Networks, Internet, and Multimedia, Chiayi, Taiwan, 2000.

### Webographie

- [1] <http://www.w3.org/TR/1998/REC-smil-19980615/>
- [2] <http://www.w3.org/TR/2001/REC-smil20-20010807/>.
- [3] <http://www.w3.org/TR/2005/REC-SMIL2-20050107/>
- [4] <http://www.w3.org/TR/SMIL3/>
- [5] <http://www.macromedia.com>.
- [6] <http://www.w3.org/XML/1998/06/xmlspec-report.htm>.
- [7] <http://limsee2.gforge.inria.fr>
- [8] <http://limsee3.gforge.inria.fr>.
- [9] <http://www.oratrix.com/>
- [10] <http://www.adobe.com>
- [11] <http://www.manalee.com>
- [12] <http://www.apple.com/quicktime.html>.
- [13] <http://www.realnetworks.com>.
- [14] <http://www.ambulantplayer.org>.
- [15] <http://go.microsoft.com/fwlink/?LinkId=54758>
- [16] <http://opera.inrialpes.fr/pocketsmil/>.
- [17] <http://www.rubicdesign.com/>
- [18] <http://www.w3.org/TR/2003/REC-MathML2-20031021/>
- [19] <http://www.w3.org/TR/xforms11/>
- [20] <http://www.francophonie.hachette-livre.fr>.

## Annexe A

Dans cette annexe A, nous allons expérimenter notre outil de modélisation et de validation d'un document SMIL à travers une étude de cas de jeux vidéo Metal Gear Solid 4 (MGS4) sur PS4 définie comme suit :

### Etude cas : Cas du DMM interactif de jeux vidéo Metal Gear Solid 4 (MGS4)

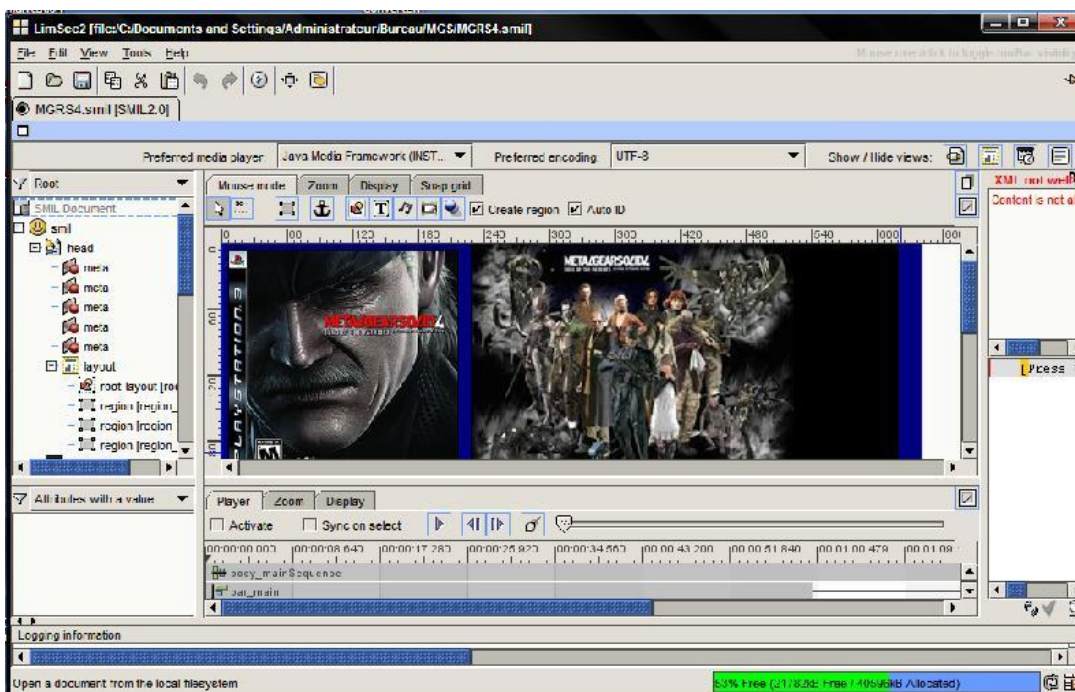
Dans le cadre de publicité sur le site de KONAMI.COM le constructeur du jeu vidéo Metal Gear Solid 4 (MGS4) faire une publicité pour le jeu, qui consiste à donner des Informations et des images et vidéos sur leur jeux:

Actualité de jeux, histoire de jeux, personnages, les astuces, contrôle de jeux, ..... etc.,

Le DMM interactif de MGS4 est composé d'une description textuelle (un texte décrivant l'histoire de jeux, les personnages, ....) et une séquence des médias : audio, image et vidéo décrivant les scènes de ce jeux.

A tout moment, l'utilisateur peut arrêter, représenter, revenir au début ou sortir complètement du document.

Dans notre travail d'expérimentation, on a utilisé l'éditeur LimSee 2 [7] pour éditer le DMM interactif MGS4 (voir **Figure A. 1**) :



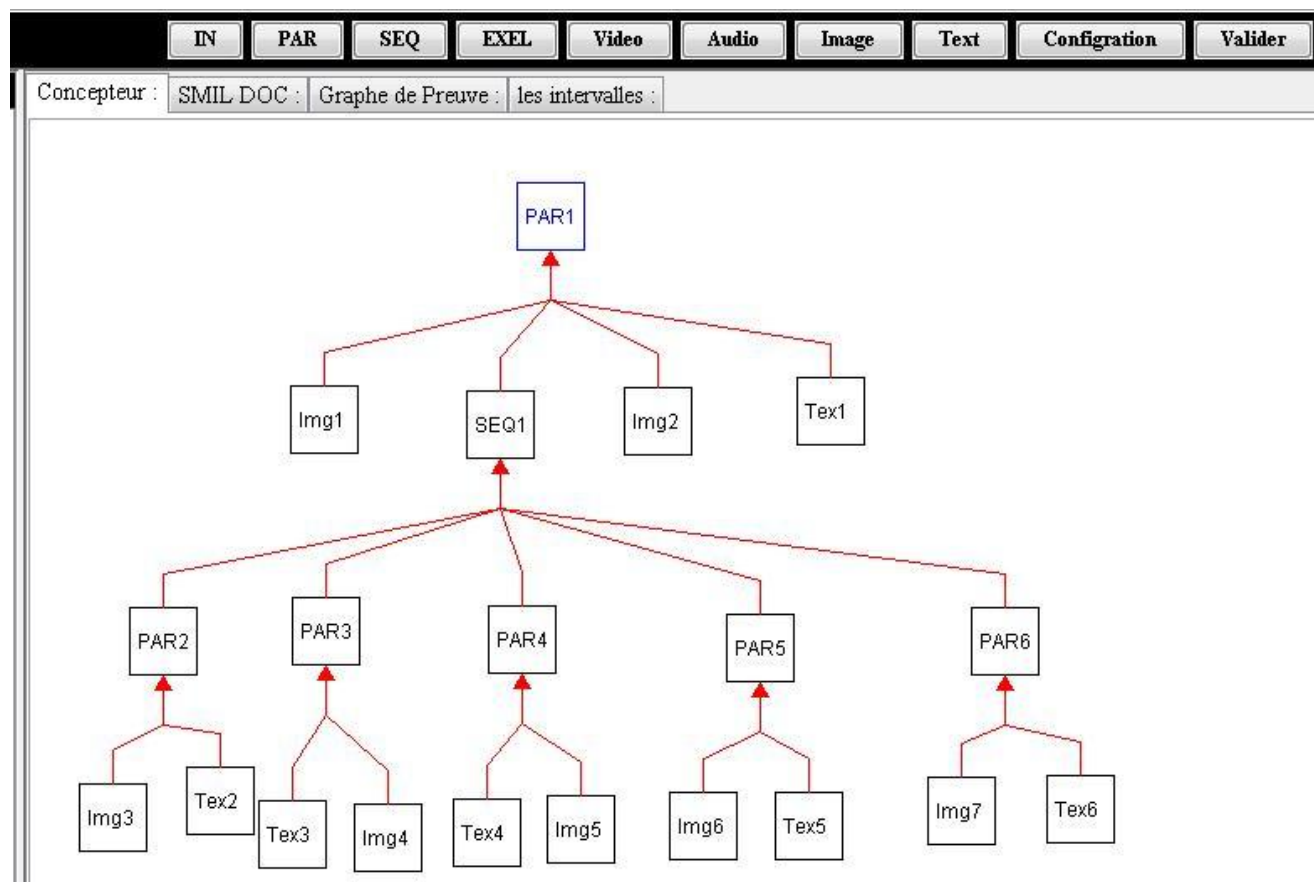
**Figure A.1** : Edition du DMM interactif MGS4

Et le code SMIL ci-dessous est automatiquement généré :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN" "http://www.w3.org/2001/SMIL20/SMIL20.dtd">

<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta
      name="title"
      content="The title"/>
    <meta
      name="author"
      content="The author"/>
    <meta
      name="copyright"
      content="The copyright"/>
    <meta
      name="abstract"
      content="The presentation abstract"/>
    <meta
      name="generator"
      content="LimSee2 ( http://wam.inrialpes.fr/software/limsee2 )"/>
  </head>
  <layout
    type="text/smil-basic-layout">
    <root-layout
      id="rootLayout"
      width="640"
      height="480"
      backgroundColor="navy"/>
  </layout>
</smil>
```

Notre outil réalisé traduit automatiquement le code SMIL généré par LimSee 2 en arbre des objets SMIL (voir **Figure A.2**)



**Figure A.2:** Exemple d'un arbre des objets SMIL correspondante au DMM interactif MGS4.

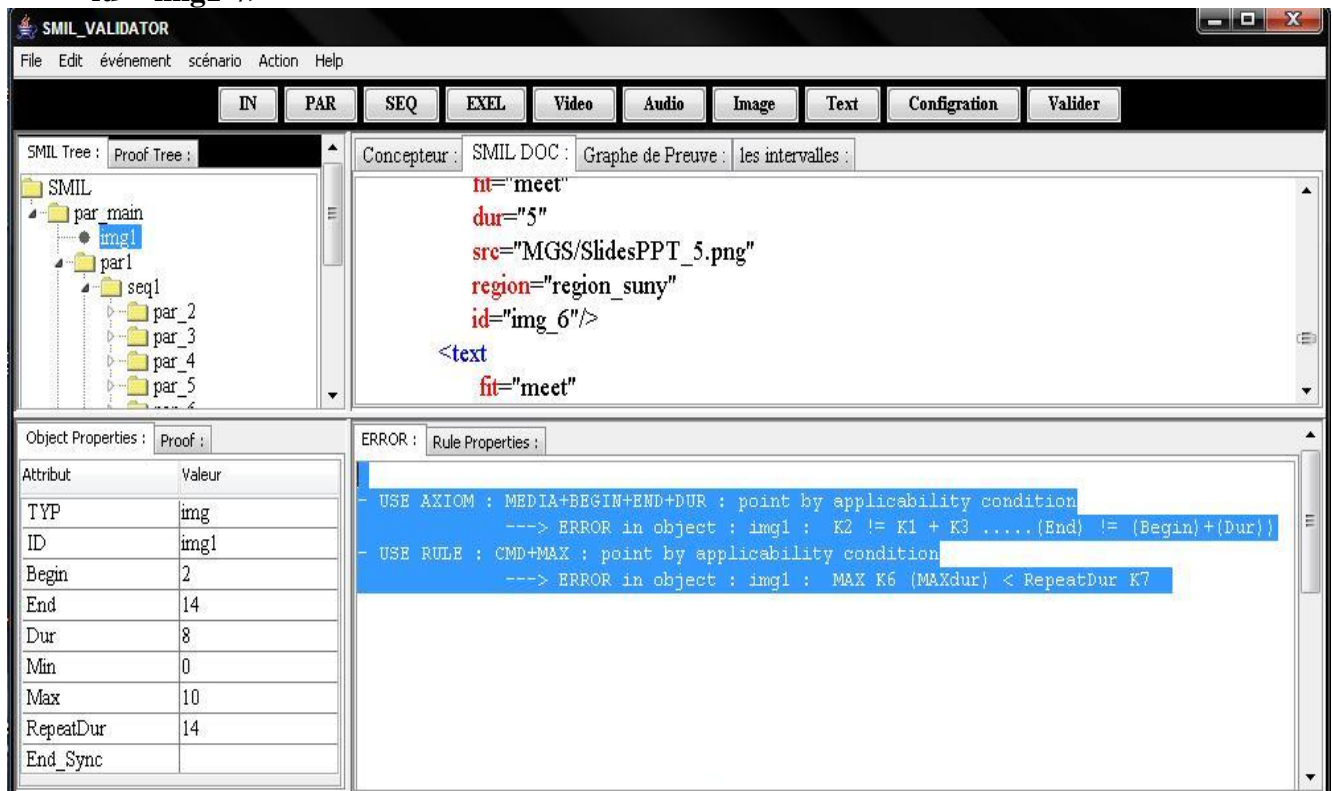


Exemple de vérification et de correction d'incohérences temporelles intra-média son montré dans la **Figure A.3**

L'objet `img1` :

```

```



**Figure A.3:** Cas d'une incohérence intra-média

L'image 'img1' contient deux conflits intra-média

1<sup>er</sup> conflit détecté par l'axiome MEDIA+BEGIN+END : par ce que -- End != Begin+dur.

2<sup>e</sup> conflit détecté par la règle CMD+MAX : par ce que – Max < RepeatDur

Cas de validation de l'incohérence temporelle inter-médias de l'élément <PAR> dans la **Figure A.4**

L'objet `par2` :

```
<par id="par2" dur="10">
  
  <text
    fit="meet"
    dur="5"
    src="MGS4/MGS/MEDIAS/text_afqwobfzgtcsw.rt"
    region="region_quffb"
    id="text_1"/>
</par>
```

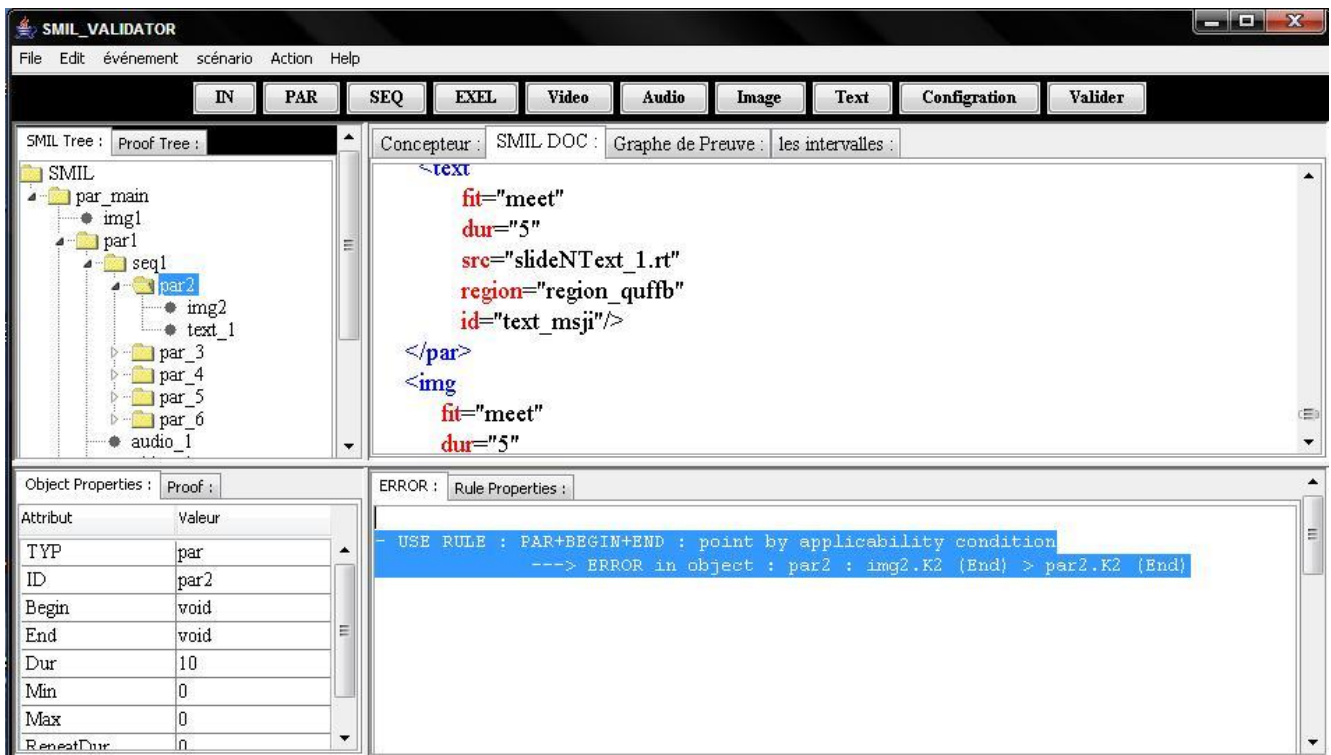


Figure A.4: Cas d'une incohérence inter-média 'PAR'.

L'élément 'par2' contient un conflit inter-média

1<sup>er</sup> conflit détecté par la règle PAR+BEGIN+END : par ce que -- img2.end > par2.end.

Cas de validation de l'incohérence temporelle inter-médias de l'élément <SEQ> dans la Figure A.5

L'objet seq1 :

```
<seq id="seq1" dur="50">
  <par id="par2" dur="10"> ..... </par>
  <par id="par3" dur="10"> ..... </par>
  <par id="par4" dur="10"> ..... </par>
  <par id="par5" dur="10"> ..... </par>
  <par id="par6" dur="20"> ..... </par>
</seq>
```

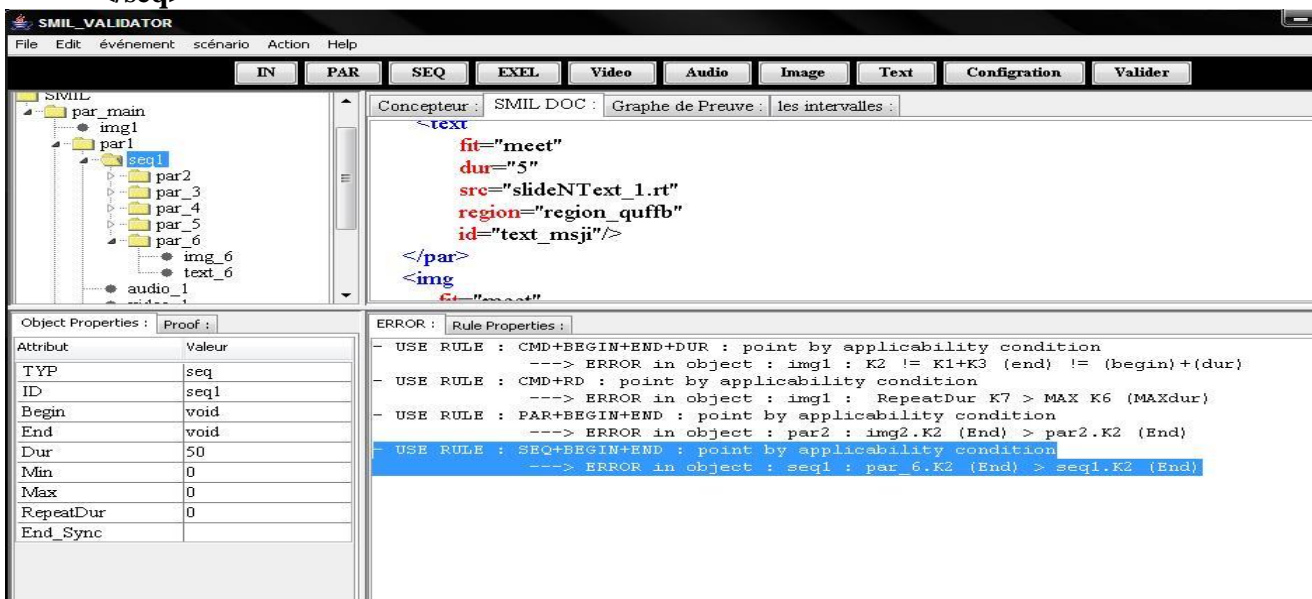


Figure A.5: Cas d'une incohérence inter-média 'SEQ'

L'élément 'seq1' contient un conflit inter-média

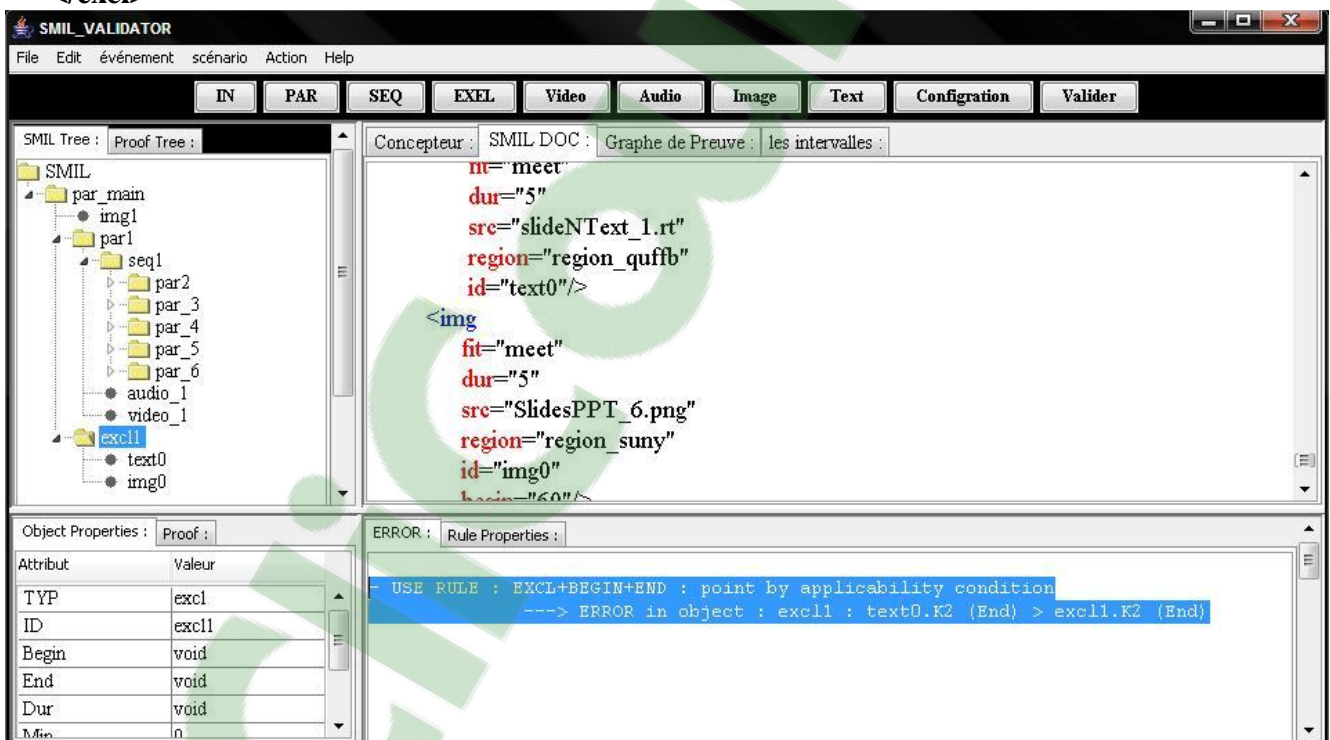
1<sup>er</sup> conflit détecté par la règle SEQ+BEGIN+END : par ce que -- par6.end > seq1.end.

**Par2.dur+ Par3.dur+ Par4.dur+ Par5.dur+ Par6.dur > seq1.dur**

Cas de validation de l'incohérence temporelle inter-médias de l'élément <EXCL> dans la **Figure A.6**

L'objet excl1 :

```
<excl id="excl1" dur="20">
  <text
    fit="meet"
    dur="25"
    src="slideNText_1.rt"
    region="region_quffb"
    id="text0"/>
  
</excl>
```



**Figure A.6:** Cas d'une incohérence inter-média 'EXCL'

L'élément 'excl1' contient un conflit inter-média

1<sup>er</sup> conflit détecté par la règle EXCL+BEGIN+END : par ce que -- text0.end > excl1.end.

Cas de validation de l'incohérence temporelle multiple-Play-Back d'un élément dans la **Figure A.7**

L'élément par3 :

```
<par
  id="par_3"
  dur="10">
  <video
    fit="meet"
    dur="5"
    begin=ev1
    src="MGS/vid.mp4"
```

```

        region="region_suny"
        id="video3"/>
<text
    fit="meet"
    dur="5"
    src="MGS4/MGS/MEDIAS/text_ddxsqfierjklqhnuyzzq.rt"
    region="region_guffb"
    id="text_2"/>
</par>

```

Object Properties : Proof :

Attribut	Valeur
TYP	par
ID	par_3
Begin	void
End	void
Dur	10
Min	∞

ERROR : Rule Properties :

```

- USE RULE : PAR+BEGIN+END : point by applicability condition
  ---> ERROR in object : par_3 : video3(2).K2 (End) > par_3.K2 (End)
- USE RULE : PAR+BEGIN+END : point by applicability condition
  ---> ERROR in object : par_3 : video3(3).K2 (End) > par_3.K2 (End)
- USE RULE : SEQ+BEGIN+END : point by applicability condition
  ---> ERROR in object : seq1 : par_6.K2 (End) > seq1.K2 (End)

```

**Figure A.7:** Cas d'une incohérence inter-média 'PAR' --- Begin+multiple-Play-Back.

L'élément 'par3' contient un conflit inter-média

1<sup>er</sup> conflit détecté par la règle PAR+BEGIN+END : par ce que la vidéo video3 jouer plusieurs fois et dépasser la fin de son conteneur parent.

vidéo3 (2).end > par1.end.

vidéo3 (3).end > par1.end.

## Résumé

Nos jours, les ordinateurs et l'Internet jouent un rôle très important dans notre vie quotidienne. On assiste à une véritable révolution technologique avec l'explosion des systèmes multimédias et du World Wide Web qui comporte : le téléenseignement, la visioconférence, la publicité, la création des sites web, l'infographie, etc. Il est devenu alors possible de combiner différents médias (son, vidéo, texte, images, animations ...) pour produire des documents plus sophistiqués communément appelés Documents Multimédia (DMM en abrégé). SMIL (Synchronized Multimedia Integration Language) est un standard du World Wide Web Consortium (W3C) créé en 1998 qui permet de spécifier l'organisation spatiale et temporelle des DMM interactifs et synchronisés. En effet, la création des DMM à l'aide du langage SMIL est une tâche difficile et génératrice d'erreurs, en particulier lorsque la structure temporelle d'un document est complexe. C'est pour cela, nous nous sommes intéressés dans ce mémoire à l'étude et à la vérification de cohérence temporelle des documents SMIL en utilisant la logique de Hoare, ce qui contribue à leur production et diffusion.

## Mots clés :

Documents Multimédias; DMM; World Wide Web; Synchronisés; W3C; Structure Temporelle; DMM Interactifs; SMIL; Cohérence Temporelle; Logique De Hoare; Validation.

# Résumé

Nos jours, les ordinateurs et l'Internet jouent un rôle très important dans notre vie quotidienne. On assiste à une véritable révolution technologique avec l'explosion des systèmes multimédias et du World Wide Web qui comporte : le téléenseignement, la visioconférence, la publicité, la création des sites web, l'infographie, etc. Il est devenu alors possible de combiner différents médias (son, vidéo, texte, images, animations ...) pour produire des documents plus sophistiqués communément appelés Documents Multimédia (DMM en abrégé). SMIL (Synchronized Multimedia Integration Language) est un standard du World Wide Web Consortium (W3C) créé en 1998 qui permet de spécifier l'organisation spatiale et temporelle des DMM interactifs et synchronisés. En effet, la création des DMM à l'aide du langage SMIL est une tâche difficile et génératrice d'erreurs, en particulier lorsque la structure temporelle d'un document est complexe. C'est pour cela, nous nous sommes intéressés dans ce mémoire à l'étude et à la vérification de cohérence temporelle des documents SMIL en utilisant la logique de Hoare, ce qui contribue à leur production et diffusion.

## Mots clés :

Documents Multimédias; DMM; World Wide Web; Synchronisés; W3C; Structure Temporelle; DMM Interactifs; SMIL; Cohérence Temporelle; Logique De Hoare; Validation.