

# Sommaire

## Chapitre I : Introduction

I.1 Contexte .....	1
I.2 Problématique.....	2
1.3 Plan du mémoire.....	2

## Chapitre II : Les Documents Multimédias (DM)

II. 1 Introduction .....	3
II. 2 Intérêt .....	3
II.3 Les médias de base ..	4
II.3.1 Définition d'un média ...	4
II.3.2 Définition du multimédia....	4
II.3.3 Les différents médias ...	5
II.3.4 Typologie des médias ..	6
II.4 Les DM.....	7
II.5 Modélisation d'un DM... ..	7
II.5.1 Les approches opérationnelles ...	8
II .5.2 Les approches à base de modèles de relations....	9
II.5.3 Modélisation des informations temporelles ...	9
II.5.4 Modélisation des informations spatiales ....	13
II.6 Les dimensions d'un DM ...	13
II.6.1 Dimension spatiale .....	14
II.6.2 Dimension temporelle .....	14
II.6.3 Dimension hypermédia ....	15
II.6.4 Dimension logique .....	15
II. 7 Domaine d'applications des DM.....	16
II.8 Synchronisation des média .....	17
II.8.1 Synchronisation intra-media.....	17
II.8.2 Synchronisation inter-média .....	17
II.8.3 Synchronisation des lèvres .....	18
II.9 Les présentations multimédia .....	19
II.9.1 L'édition .....	19

---

II.9.2 Le contrôle .....	20
II.9.2.1 Cohérence qualitative .....	20
II.9.2.2 Cohérence quantitative.....	21
II.9.3 La présentation .....	22
II.10 Les standards de structuration des DM .....	22
II.10.1 SGML .....	22
II.10.2 HyTime.....	23
II.10.3. XML .....	24
II.10.4 ODA .....	25
II.10.5 MHEG .....	25
II.10.6 SMIL ....	26
II.11 Conclusion.....	27

### **Chapitre III : Analyse et Validation des documents SMIL**

III.1 Introduction.....	28
III.2 Définition de SMIL .....	29
III.3 Historique de SMIL .....	30
III.4 Pourquoi SMIL .....	31
III.5 La modularisation et le profilage .....	32
III.5.1 Structure d'un document SMIL .....	33
III.5.2 Module Méta-Information.....	35
III.5.3 Module Layout ...	36
III.5.4 Le module Media Objects .....	37
III.5.5 Manipulation du temps ...	39
III.5.6 Le temps et la synchronisation .....	39
III.5.6.1 Les attributs de temporisation ...	40
III.5.6.2 Les attributs clipBegin et clipEnd .....	41
III.5.7 Les attributs repeatCount et repeatDur et fill ...	41
III.5.8 Les éléments de synchronisation .....	42
III.6 Le module Linking ...	45
III.7 Le module Content Control .....	46
III.8 Le module Animation.....	47
III.9 Le module Transitions.....	48

III.10 Analyse des documents SMIL...	48
III.10.1 Structuration...	49
III.10.2 Navigation.....	49
III.10.3 Placement temporel et synchronisation.....	49
III.11 Cohérences des documents SMIL...	50
III.11.1 Conflit temporel intra-élément ....	50
III.11.2 Conflit temporel inter-éléments .....	51
III.12 Inconsistances des documents SMIL ...	52
III.13 Les approches de modélisation et validation des documents SMIL.....	54
III.14 Conclusion ...	56
 <b>Chapitre IV : Conception et Réalisation d'un outil de Modélisation et de Validation des Documents SMIL</b>	
IV.1 Introduction .....	57
IV.2 Notre proposition .....	57
IV.2.1 Les Réseaux de Petri simples .....	58
IV.2.2 Les réseaux de Petri temporels.....	59
IV.2.3 Modélisation d'un document SMIL .....	62
IV.2.3.1 Modélisation des éléments SMIL de base.....	62
IV.2.3.2 Modélisation de l'opérateur de lecture en séquence <seq> .....	63
IV.2.3.3 Modélisation de l'opérateur de lecture en parallèle <par>.....	64
IV.2.3.4 Modélisation de l'opérateur de lecture en exclusivité <excl>.....	66
IV.2.3.5 Les objets des liens hypermédia.....	67
IV.2.3.6 Exemples de modélisation d'un document SMIL du cœur .....	69
IV.3 Validation d'un document SMIL.....	72
IV.4 Expérimentation et évaluation .....	75
IV.4.1 Architecture générale de l'outil.....	75
IV.4.2 Environnement de travail. ....	76
IV.4.2.1 Environnement matériel.....	76
IV.4.2.2 Environnement logiciel .....	76
IV.4.3 Implémentation des modules de l'outil.....	77
IV.5 Conclusion.....	85
<b>Conclusion &amp; Perspectives.....</b>	<b>86</b>
<b>Bibliographie.....</b>	<b>88</b>
<b>Annexe A.....</b>	<b>95</b>

## Résumé

Grâce à la puissance des ordinateurs et au débit des communications actuels, il est possible de combiner différents médias (son, vidéo, texte, images, animations ...) pour produire des documents plus sophistiqués communément appelés Documents Multimédia (DM en abrégé). Ces documents sont devenus de plus en plus répandus et utilisés dans différents domaines. Ils sont caractérisés par l'intégration de la synchronisation de plusieurs média (audio, vidéo, texte, image, ...).

Le langage SMIL (*Synchronized Multimedia Integration Language*) qui est un standard du World Wide Web Consortium (W3C) suscite un intérêt croissant, notamment dans le cadre de l'édition de DM synchronisés. L'aspect temporel est donc un point crucial pour de telles présentations. Ce langage permet aussi d'intégrer d'autres informations d'ordre spatial, hypermédia et des Méta 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. Ce mémoire utilise les réseaux de Petri temporels de Merlin pour la modélisation et la validation des documents SMIL ce qui contribue à leur production et diffusion.

**Mots clés :** Documents Multimédias (DM), SMIL, Réseaux de Petri temporels, Formalisation, Validation.

## Abstract

Thanks to the power of computers and to the rate of the actual communications, it is possible to combine various media (sound, video, text, images, animations) to produce more sophisticated documents collectively called Multimedia Documents (abbreviated MD).

The language SMIL (*Synchronized Multimedia Integration Language*) that is a standard of the World Wide Web Consortium (W3C) arouses an increasing interest, in particular within the framework of the edition of synchronized MD. The temporal aspect is thus a crucial point for such presentations. This language also allows to integrate other information of spatial order, hypermedia and Meta information.

The complexity of the temporal synchronization of SMIL presentations makes difficult even impossible to guarantee the validity of a scenario by using informal methods. This manuscript uses the Time Petri nets of Merlin for modeling and validation of SMIL documents what contributes to their production and distribution.

**Key words:** Multimedia Documents (MD), SMIL, Time Petri nets, Formalization, Validation.

## Liste des Figures

### Chapitre II

<b>Figure II.1</b> : Le processus de création des DM.....	4
<b>Figure II.2</b> : Définition d'un DM.....	7
<b>Figure II.3</b> : Information temporelle attachée aux objets .....	8
<b>Figure II.4</b> : Le TimeLine d'un DM .....	10
<b>Figure II.5</b> : Les réseaux de points temporels .....	10
<b>Figure II.6</b> : Le modèle des relations de Wahl and Rothermel .....	12
<b>Figure II.7</b> : Trois relations spatiales directionnelles entre deux points .....	13
<b>Figure II.8</b> : Les quatre dimensions d'un DM .....	14
<b>Figure II.9</b> : Dimension spatiale (droite) d'une présentation multimédia (gauche) du DM de la ville d'Oran .....	14
<b>Figure II.10</b> : Dimension temporelle d'un DM .....	15
<b>Figure II.11</b> : Dimension hypermédia d'un DM .....	15
<b>Figure II.12</b> : Dimension logique d'un DM .....	16
<b>Figure II.13</b> : Exemple d'une synchronisation intra-média .....	17
<b>Figure II.14</b> : Exemple d'une synchronisation inter-média .....	18
<b>Figure II.15</b> : Exemple d'une synchronisation des lèvres ( <i>lip-synchronization</i> ) .....	18
<b>Figure II.16</b> : Cycle de vie d'un DM .....	19
<b>Figure II.17</b> : Exemple d'un scénario valide.....	20
<b>Figure II.18</b> : Exemple d'un scénario invalide .....	21
<b>Figure II.19</b> : Exemple d'un scénario valide.....	21
<b>Figure II.20</b> : Exemple d'un scénario invalide .....	21

### Chapitre III

<b>Figure III.1</b> : Signification du langage SMIL.....	29
<b>Figure III.2</b> : Exemple d'un document SMIL .....	34
<b>Figure III.3</b> : Structure d'un document SMIL .....	35
<b>Figure III.4</b> : Durée d'un élément multimédia .....	41
<b>Figure III.5</b> : Illustration des attributs clipBegin, clipEnd .....	41
<b>Figure III.6</b> : Illustration des attributs repeatCount et repeatDur .....	42
<b>Figure III.7</b> : Illustration de l'attribut fill .....	42
<b>Figure III.8</b> : Exemple d'éléments joués en parallèle .....	43

<b>Figure III.9</b> : Exemple d'éléments joué en séquence.....	44
<b>Figure III.10</b> :Exemple d'éléments jouées en parallèle et en séquence .....	44
<b>Figure III.11</b> :Exemple de synchronisation en enchainement .....	45
<b>Figure III.12</b> : Exemple d'une partie temporelle cliquable.....	46
<b>Figure III.13</b> : Conflit temporel intra-élément.....	51
<b>Figure III.14</b> : Conflit temporel inter-éléments .....	51

## Chapitre IV

<b>Figure IV.1</b> : Réseau de Petri marqué.....	59
<b>Figure IV.2</b> : Exemple d'un RdP t-temporel.....	61
<b>Figure IV.3</b> : Modélisation d'un document SMIL par le t-RdP.....	62
<b>Figure IV.4</b> : Modélisation du média objet de base par le t-RdP.....	63
<b>Figure IV.5</b> : Modélisation de l'élément <seq> par le t-RdP .....	64
<b>Figure IV.6</b> : Modélisation de l'élément <par> à l'aide du t-RdP l'attribut (endsync = 'last'), (b) l'attribut (endsync = 'first'), (c) l'attribut (endsync = 'id-object') .....	66
<b>Figure IV.7</b> : Modélisation de l'élément <excl> à l'aide du t-RdP.....	66
<b>Figure IV.8</b> : Modélisation des éléments de liens hypermédia : (a): Show=Pause, (b): Show=Replace, (c): Show=New, à l'aide du t-RdP.....	69
<b>Figure IV.9</b> : Exemple du DM interactif du cœur.....	70
<b>Figure IV.10</b> : Exemple de modélisation du DM interactif du cœur à l'aide des t-RdP .....	72
<b>Figure IV.11</b> : Architecture générale de l'outil.....	75
<b>Figure IV.12</b> : Extraction des mots clés .....	78
<b>Figure IV.13</b> : Procédure d'extraction des mots clés.....	78
<b>Figure IV.14</b> : Implémentation de l'opérateur <par>.....	80
<b>Figure IV.15</b> : Implémentation de l'opérateur <seq>.....	80
<b>Figure IV.16</b> : Implémentation es médias .....	81
<b>Figure IV.17</b> : Procédure de détection des incohérences temporelles intra-média .....	82
<b>Figure IV.18</b> : Procédure de détection des incohérences temporelles inter-médias.....	84
<b>Figure IV.19</b> : Procédure de détection des inconsistances.....	84
<b>Figure IV.20</b> :Exemple d'une présentation du document SMIL du Cœur à l'aide du player RealOne.....	85

## Liste des Tableaux

### Chapitre II

<b>Tableau II.1</b> – Les Relations d’Allen.....	12
--	----

### Chapitre III

<b>Tableau III.1.</b> Fonction et extensions des éléments médias.....	38
---	----

<b>Tableau III.2.</b> Les attributs Media Objects.....	38
--	----

<b>Tableau III.3.</b> Les valeurs acceptées par l’attribut fill.....	42
--	----

<b>Tableau III.4</b> Les valeurs acceptées par l’attribut endsync.....	43
--	----

<b>Tableau III.5</b> Les attributs de test de <switch/>.....	47
--	----

<b>Tableau III.6</b> Calcul de la durée active et situations de conflit intra-élément.....	51
--	----

<b>Tableau III.7</b> Comparaison entre les différentes approches de modélisation et de validation des documents SMIL.....	56
---	----

## Chapitre I : Introduction

### I.1 Contexte

Les percées technologiques récentes en matière de multimédia ont permis d'accroître les possibilités d'interaction entre l'homme et la machine. La manipulation digitale de graphiques, de sons 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'elles manipulent est plus riche, puisqu'elles intègrent dans ces documents du son et des images animées. Ces nouveaux types de documents électroniques sont communément appelés *documents multimédia* (DM en abrégé). Le développement des réseaux de communication rapides permet par ailleurs de relier ces documents pour constituer ainsi des réseaux de *documents hypermédia* à l'intérieur desquels on peut envisager de naviguer à la manière du *World Wide Web*.

La présentation des DM à l'utilisateur a jusqu'ici été effectuée principalement en fonction de 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.

Les standards et les modèles employés pour représenter les documents classiques, comme *SGML* et *XML*, sont devenus inadaptés pour représenter de tels documents. De ce fait, de nouveaux standards comme *HyTime*, *MHEG*, *SMIL* émergent pour les compléter.

Le standard SMIL est le plus utilisé à l'heure actuelle pour la synchronisation et la présentation des DM complexes, mais ne garantit pas leur qualité de présentation. En effet, il faut intégrer des outils formels de modélisation de SMIL comme les réseaux de Petri temporels, les automates temporels, la logique temporelle, etc. pour valider les présentations des documents SMIL.

Le travail présenté dans ce mémoire a pour objectif de contribuer au domaine de la modélisation et la validation des documents SMIL.



## I.2 Problématique

SMIL (*Synchronized Multimedia Integration Language*) est une application XML normalisée par le W3C [1]. C'est un langage de balisage qui permet de spécifier les différentes structures d'un DM à savoir logique, spatiale, temporelle, hypermédia, etc. Il est composé d'un ou de plusieurs objets multimédias accessibles au moyen d'une URL (<img>, <audio>, <video>, etc., et l'attribut 'src'). Il permet de définir le placement temporel et spatial de ces objets (<region>, 'left', 'top', etc). On peut par exemple définir le temps de début et le temps de fin d'un média ('debut', 'fin', 'dur') ou encore sa durée. On peut également définir la présentation parallèle de deux ou plusieurs médias ou leur présentation en séquence. Comme on peut présenter des segments spatiaux ou temporels d'un média ou les deux en même temps (clip-begin, clip-end, coords). Ce standard donne la possibilité d'associer des liens temporels aux différents médias. Comme il donne la possibilité de décrire la présentation (<meta>) de ses différents composants et même des segments temporels des médias continus tel que l'audio et la vidéo (<metadata>, 'RDF', 'duration').

La complexité de conception d'un document SMIL rend difficile voire impossible de garantir leur qualité de présentation à partir du langage uniquement. Par contre, en se basant sur les outils mathématiques, les techniques formelles offrent au langage SMIL une sémantique formelle, complète et dépourvue d'ambiguïté, ainsi que des techniques d'analyse des propriétés logiques et temporelles facilitant ainsi sa validation. En outre, la vérification de cohérence temporelle et spatiale, et de consistance d'un document SMIL est plus que nécessaire avant sa présentation.

Ce mémoire vise à contribuer, plus précisément, à la modélisation et la validation d'un document SMIL en utilisant les réseaux de Petri t-temporels (RdP) de Merlin [Merl 74]. La vérification de cohérence temporelle intra et inter-média et de consistance d'un document SMIL est prise en charge par notre outil ainsi réalisé.

### 1.3 Plan du mémoire

Ce mémoire est organisé comme suit :

Le chapitre II est une introduction au multimédia et aux documents multimédia (DM) comprenant les définitions des objets médias, et leurs synchronisations multimédia

Le chapitre III concerne l'étude du langage SMIL : analyse et validation des documents SMIL.

Le chapitre IV présente la conception et la réalisation de notre outil de modélisation et de validation des documents SMIL.

L'annexe A : resume l'expérimentation de notre outil en utilisant une étude de cas du document SMIL de la ville d'Oran.

## **Chapitre II : Les Documents Multimédias (DM)**

### **II. 1 Introduction**

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

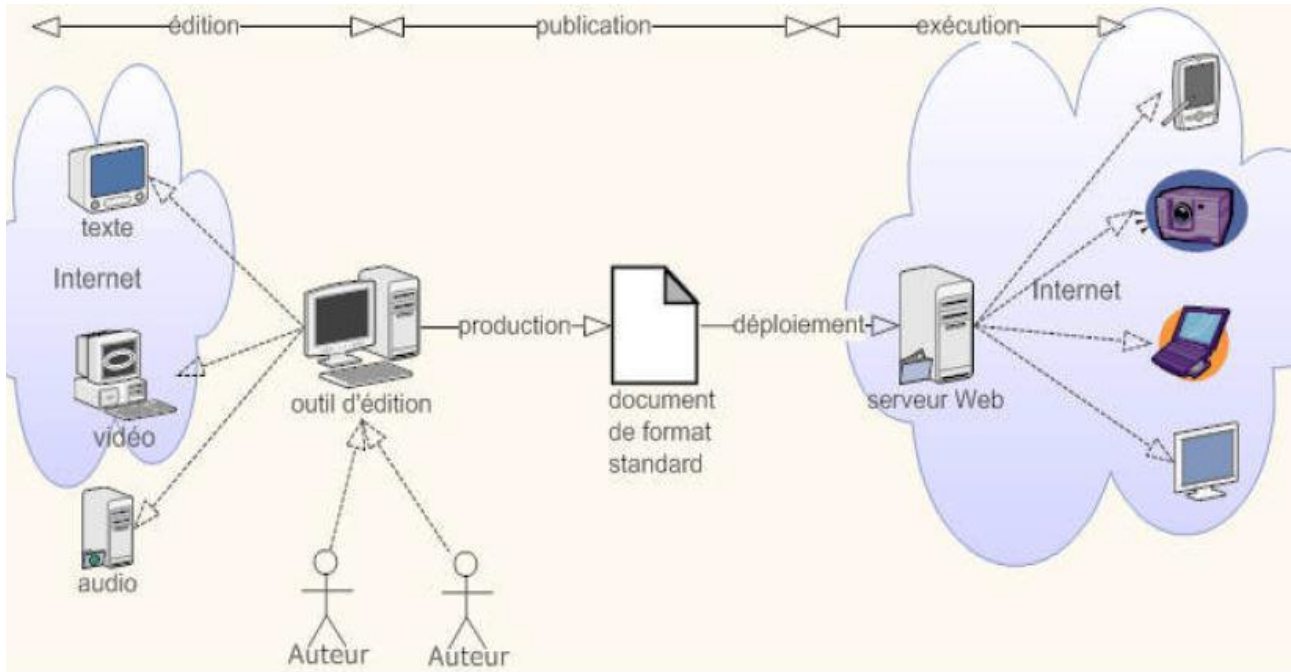
Grâce à la puissance des ordinateurs et au débit des communications actuels, il est possible de combiner différents médias (son, vidéo, texte, images, animations ...) pour produire des documents plus sophistiqués communément appelés Documents Multimédia (DM en abrégé) [Laya 96], que l'on peut diffuser aussi bien sur support fixe (CD, DVD...) qu'à travers un réseau. Les DM manipulés possèdent une organisation très proche des supports papiers. Ils possèdent en effet une structure logique (décomposition en parties, chapitres, sections...) et une structure spatiale (placement précis des objets à l'écran). En revanche, ils sont très peu organisés au niveau temporel, alors que des DM par exemple éducatifs (encyclopédie, atlas, cours...) nécessitent des scénarios très planifiés prenant en compte des interactions avec l'utilisateur.

Les DM apparaissent comme une nouvelle génération d'applications informatiques dont la conception à grande échelle met en jeu une problématique nouvelle. Au cœur de cette problématique se situe la caractérisation de la structure temporelle et spatiale d'un document qui peut être exprimé sous la forme d'une conjonction de contraintes (temporelles/spatiales) de synchronisation, dont certaines dépendent directement d'interactions avec l'utilisateur (caractère multimédia du document). La qualité de la présentation d'un document repose alors en grande partie sur la cohérence globale de ces contraintes de synchronisation et la possibilité ultérieure de les respecter effectivement sur une plate-forme multimédia donnée.

### **II. 2 Intérêt**

Avec l'évolution des applications sur l'Internet, un DM est considéré comme étant une composition complexe d'objets multimédias selon quatre dimensions : temporelle, spatiale, logique et hypermédia. Cette évolution est le résultat de nouvelles capacités technologiques des ordinateurs (CPU, mémoire, disques,...), des réseaux (débit, protocoles), et des techniques numériques de traitement des données multimédias: son, vidéo, hypertexte, animation (codage, compression/

décompression) et qui rend la création des DM de plus en plus complexe. En effet, un DM intègre des médias de types différents à partir de différentes sources par des compositions spatiale et temporelle (**Figure 1**).



**Figure II.1:** Le processus de création des DM

## II.3 Les médias de base

### II.3.1 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*" [Gaff 02], après le mot est changé par le sens de l'intermédiaire et de moyen de communication de la pensée [Maub 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 et finalement il a été raccourci en média. Cette dernière acception est généralisée en informatique en considérant un média comme un moyen de transmettre, de stocker ou de présenter des informations [Leco 99].

### II.3.2 Définition du multimédia

D'après [20], le multimédia est une technique permettant de rassembler sur un même support des moyens audiovisuels (texte, son, images fixes et animées) et des moyens informatiques (programmes, données) pour les diffuser simultanément et de manière interactive.

D'après [Rois 99], le qualificatif de multimédia se rencontre dès lors qu'une application traite des documents qui ne sont plus seulement textuels mais sous forme: image, audio etc. C'est ainsi

que des documents comportant uniquement du texte et des images sont appelés multimédia. À l'autre extrême de cette échelle, on trouve des documents caractérisés par une dynamique beaucoup plus importante grâce à l'intégration d'informations textuelles, d'images mais aussi du son, de la vidéo, des images animées, des éléments contrôlés par des programmes externes, etc.

Malgré qu'il est difficile de s'accorder sur une définition unique, de part son usage dans des contextes très différents, la plupart des auteurs s'accordent pour dire que le multimédia désigne l'intégration de texte, son, images fixes ou animées, sous forme numérique, et l'utilisation d'un logiciel permettant l'interactivité [Kirm 08].

### **II.3.3 Les différents médias**

Les médias peuvent être vus comme étant les briques fondatrices du multimédia. Un ensemble bien ordonné de médias permet une communication plus intéressante et plus dynamique qui peut mieux capter l'attention des interlocuteurs [Jedi 05]. Les média de base sont: le texte, l'image, l'audio et la vidéo et sont définis comme suit [Adib 00] :

#### **- Texte**

Le texte est un média souvent représenté comme une liste de chaînes de caractères, correspondant à un document entier ou à une partie, par exemple le titre ou le résumé. C'est le média artificiel le plus ancien au monde et celui qui a fait l'objet des premiers développements en informatique. Ce type de média est souvent représenté par le format de codage qui est l'ASCII.

Un texte est généralement découpé en chapitres, sections, paragraphes qui composent le contenu du document. Une base de documents est organisée comme un index de chaîne de caractères (par exemple: index de titres, de mots clés).

#### **- Image**

Les images correspondent à des dessins, de la peinture, de la photographie ou de l'impression. Elles sont aussi une identité bidimensionnelle (composées de points ou de pixels possédant chacun une couleur). En 3D, les images sont conçues comme des assemblages complexes de formes correspondant à des objets réels.

Des formats différents sont utilisés pour la digitalisation des images parmi lesquels : GIF, JPEG, TIFF, etc. C'est une donnée spatiale qui peut être stockée dans les formats bitmap (grille de points bi-directionnelle) et vectoriel (collection de formes et de leur positions respectives dans l'espace défini).

**- Audio**

L'audio inclut les données de type vocal : paroles digitalisées, musique digitalisée, synthèse de paroles, etc. Par exemple une musique ou une phrase prononcée par un ou plusieurs locuteurs sont de type audio, et le domaine de la reconnaissance de la parole, permet d'identifier un locuteur spécifique, de formuler des requêtes sur des documents parlés, etc.

Des formats différents sont utilisés pour la digitalisation du son parmi lesquels : WAV, MP3, etc. L'audio est l'un des deux principaux média utilisés aujourd'hui, avec la vidéo. C'est un média temporel dont l'information est représentée par un signal périodique et continu. C'est ce signal qui est émis des périphériques comme les haut-parleurs et perçu par l'oreille de l'utilisateur humain.

**- 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.3.4 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.

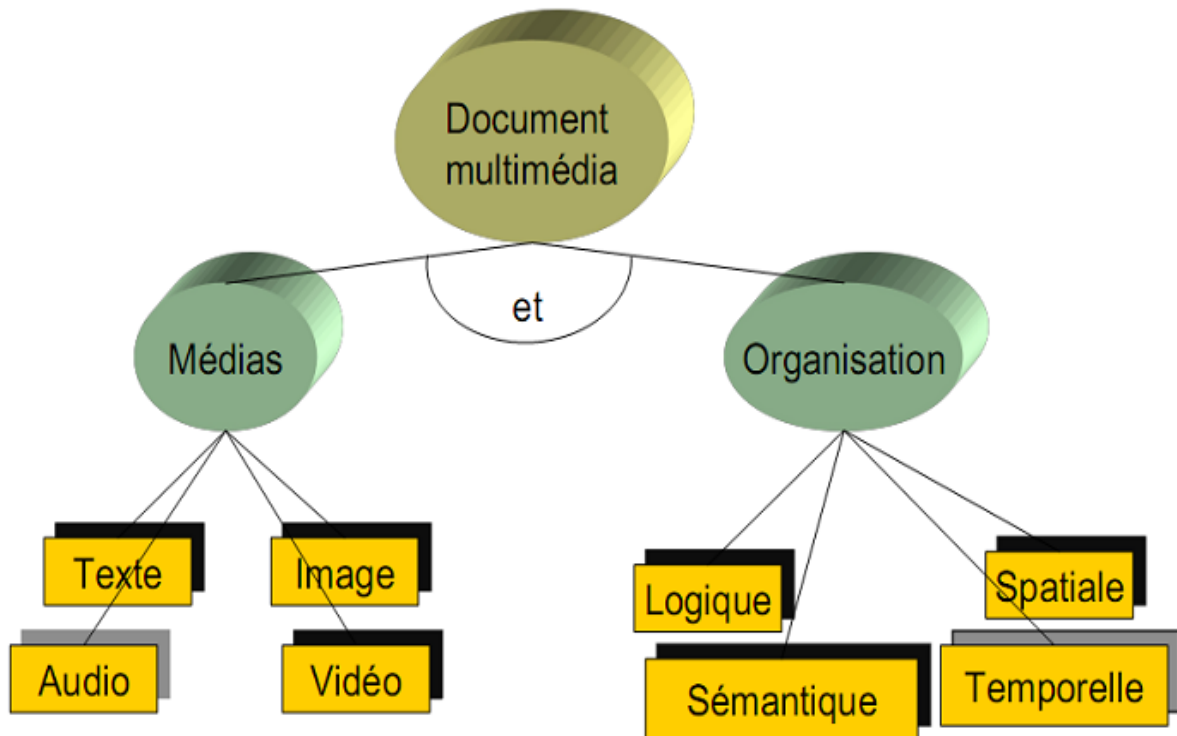
Ils peuvent aussi être caractérisés par leur mode de perception:

- **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.4 Les DM

Un DM 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 (figure II.2).



**Figure II.2 :** Définition d'un DM [laya 97]

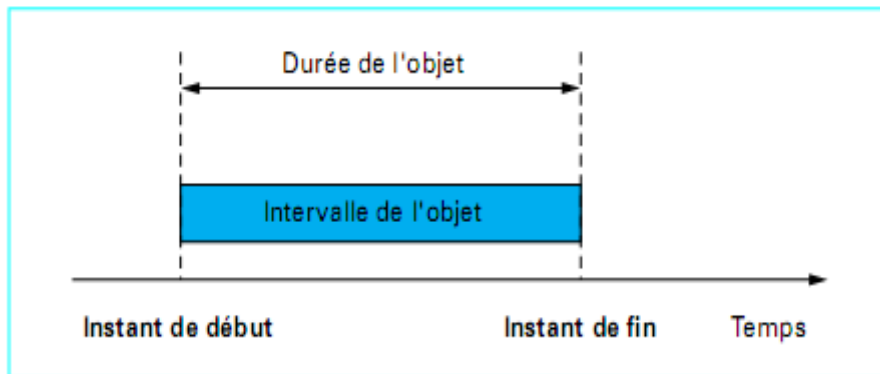
Ainsi, les documents que nous qualifions dans la suite de DM sont non seulement caractérisés par des contenus de nature diverse: statique comme les textes, les graphiques, les tableaux et les images, ou dynamique comme l'audio, les vidéos et les animations, mais aussi par l'organisation temporelle de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

## II.5 Modélisation d'un DM [Jour 99]

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

Différents travaux théoriques, qui proviennent notamment du domaine de l'intelligence artificielle (IA) dans lequel le problème de la synchronisation temporelle est crucial, ont permis de formaliser l'information temporelle des objets ainsi que des relations entre eux. Ainsi, tout objet multimédia peut être manipulé à travers trois informations temporelles principales (**Figure II.3**):

- son instant de début ;
- sa durée de présentation ;
- son instant de fin.



**Figure II.3 :** Information temporelle attachée aux objets

Lors de l'édition d'un DM, la spécification des informations de présentation (instants de début et fin du média, ses positions sur l'écran, ses liens éventuels, etc.) se fait selon deux types d'approches [Mare 09] :

1. **Les approches opérationnelles ou impératives:** trois familles d'outils ont été identifiées, les outils qui s'appuient sur un axe temporel absolu, les langages de scripts et les représentations par structures de graphes (arbre ou réseau de Petri);
2. **Les approches à base de modèles de relations :** deux familles ont été définies, celles qui s'appuient sur l'algèbre d'instant et l'algèbre d'intervalles.

### II.5.1 Les approches opérationnelles

Dans cette classe les informations d'instant de début, de fin, de positionnement et hypermédia sont spécifiées explicitement; c'est-à-dire, en donnant leurs valeurs exactes.

Selon les trois familles, ces informations sont exprimées soit en s'appuyant sur une métaphore graphique d'un axe (TimeLine) pour représenter les dimensions considérées dans l'approche, comme Flash et Director de Macromedia (devenus propriété d'Adobe) [5]. Soit avec des langages de scripts, qui viennent généralement pour compléter les approches à base du TimeLine comme Lingo [5] qui est associée à Director. Soit avec des structures de graphes comme les réseaux de Petri OCPN [Litt 90] et HTSPN [Séna 96].

## II.5.2 Les approches à base de modèles de relations

Dans cette classe, les informations d'instants de début, de fin, de positionnement et hypermédia sont exprimées à l'aide de relations (temporelle et spatiales) entre les médias. Des modèles sont proposés et utilisés dans des systèmes d'édition et de présentation des DM interactifs, comme Madeus [Laya 97] ou dans des standards, comme SMIL [2] qui utilisent le modèle d'Allen [Alle 83], MediaStudio qui utilise le modèle de *Wahl et Rothermel* étendu [Wahl 94].

### II.5.3 Modélisation des informations temporelles

Pour la spécification des informations temporelles, deux classes de base peuvent être distinguées [Vanb 90]: La première classe exprime le temps par des instants dans l'espace de temps [Hamb 72], tandis que la deuxième utilise les intervalles comme unités atomiques [Alle 83]. Cela débouche sur deux types de représentation :

#### 1. Une représentation fondée sur les **instants**, en particulier les instants de début et de fin des objets

Dans les relations à base d'**instants** [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 :

#### 1.1 TimeLine

L'axe de temps est utilisé pour représenter la dimension temporelle d'un DM. 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.4**). De même, nous pouvons trouver ce modèle dans le domaine des réseaux sous forme d'estampille [Sant 93][Bult 05].



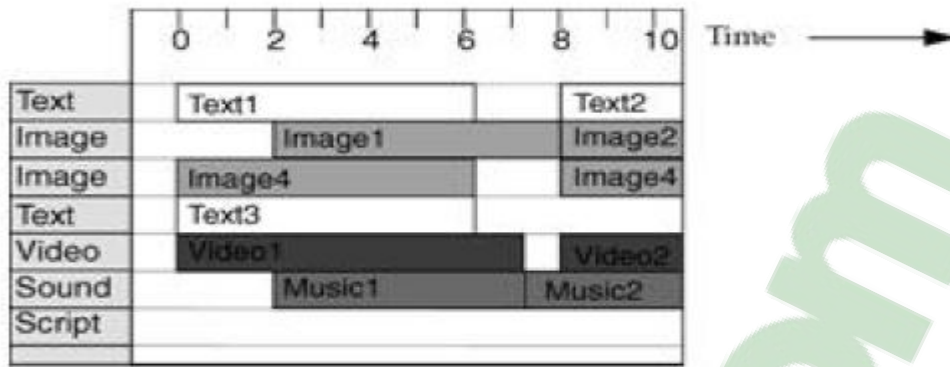


Figure II.4 : Le TimeLine d'un DM

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.5). 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.

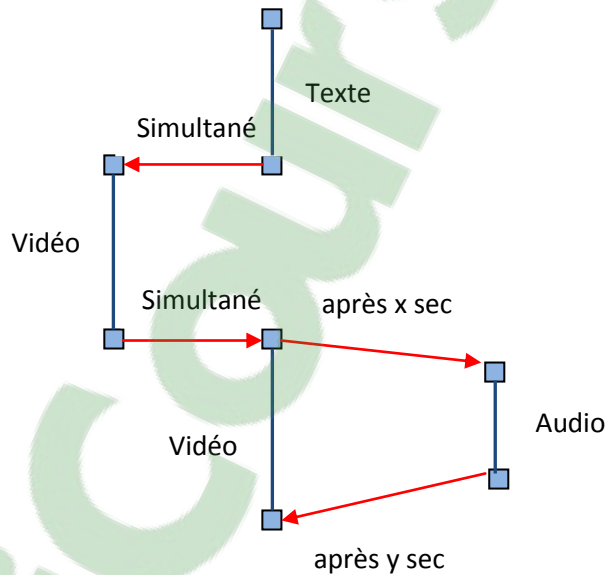


Figure II.5 : Les réseaux de points temporels

2. Une représentation fondée sur les intervalles correspondant à la durée des objets

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] :

- **Les intervalles discrets** : Ce sont des unités dont la présentation ne dépend pas du temps, comme le texte ou les images fixes.
- **Les intervalles déterministes continus** : Ils correspondent à des données dont la présentation dépend du temps mais la valeur de durée est connue a priori. Des flots audio et vidéo sont des exemples de telles unités temporelles.
- **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.

Dans les relations à **base d'intervalles**, les relations possibles entre deux éléments multimédias se réduisent à toutes les combinaisons de placement possibles de deux intervalles sur une droite orientée.

## 2.1 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.1**) et leurs relations inverses, l'égalité étant elle-même son inverse. Ces treize relations se répartissent en deux classes: celle des relations de séquentialité notée **seq** dans le **Tableau II.1** et celle des relations introduisant le parallélisme de présentation, notée **par** dans le **Tableau II.1**. Dans l'avant-dernière colonne du **Tableau II.1**, 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  $x$ .

Tableau II.1 – Les Relations d’Allen

Relation	Symbole	Inverse	Relation à base d’instantants équivalente	Classe
x avant y	b	bi	$x^- < x^+ < y^- < y^+$	Seq
y suit 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

2.2 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.6). Ils ont ainsi augmenté l’expressivité des 29 relations tout en simplifiant l’édition d’un scénario.

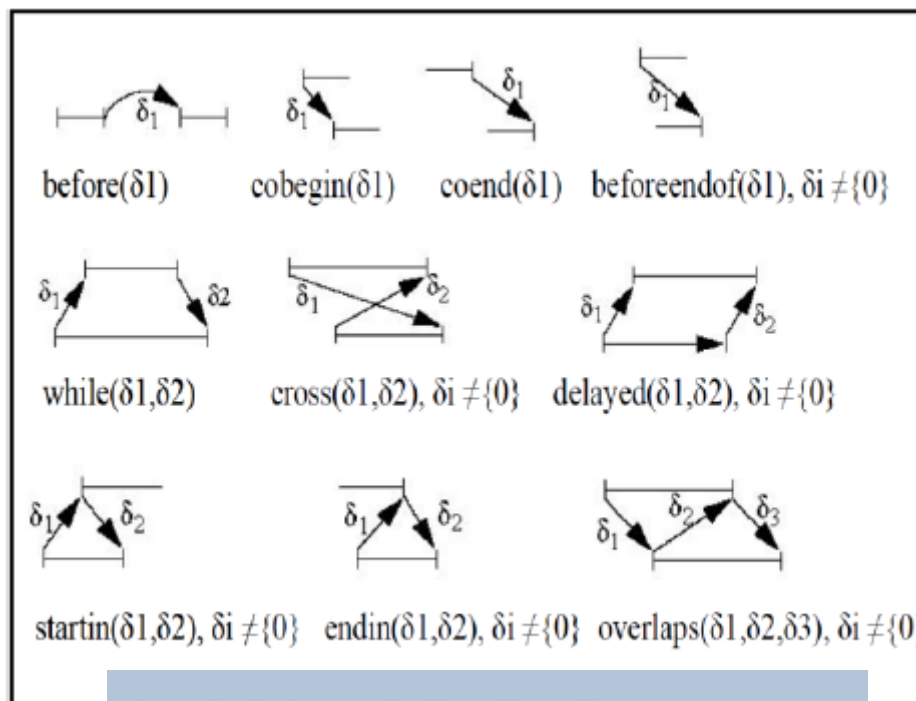
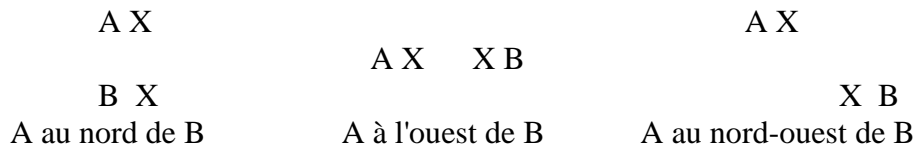


Figure II.6 : Le modèle des relations de Wahl and Rothermel

### II.5.4 Modélisation des informations spatiales

Pour décrire la présentation spatiale d'un DM, la plus part des systèmes utilisent la représentation spatiale directionnelle [Papa 94] qui permet de définir des orientations dans l'espace entre deux médias. Dans cette représentation, un média est représenté par deux intervalles correspondant à ces projections sur les deux axes x et y. L'ensemble des relations directionnelles est obtenu en combinant les intervalles de deux médias sur les axes x et y.

Soient deux points A et B, représentant par exemple les centres de gravité des objets multimédia visibles (comme des images, des vidéos ...). Neuf relations spatiales directionnelles sont possibles entre ces deux points : nord, nord-est, nord-ouest, est, ouest, sud, sud-est, sud-ouest, même position. La **figure II.7** présente trois relations spatiales directionnelles entre A et B (nord, ouest et nord-ouest).



**Figure II.7** Trois relations spatiales directionnelles entre deux points

### II.6 Les dimensions d'un DM

Un DM contient plusieurs types d'objet multimédia: texte, son, image, et vidéo etc. Lors de la création d'une présentation multimédia, le modèle document doit permettre la présentation de toutes les relations qui peuvent exister entre les objets multimédias. Ces relations peuvent porter sur la description de l'organisation logique du document, la synchronisation temporelle, la présentation spatiale des médias et l'interconnexion entre les différents éléments (liens hypermédias). Il est donc possible d'analyser un DM selon quatre dimensions (**Figure II.8**) [Rois 99]:

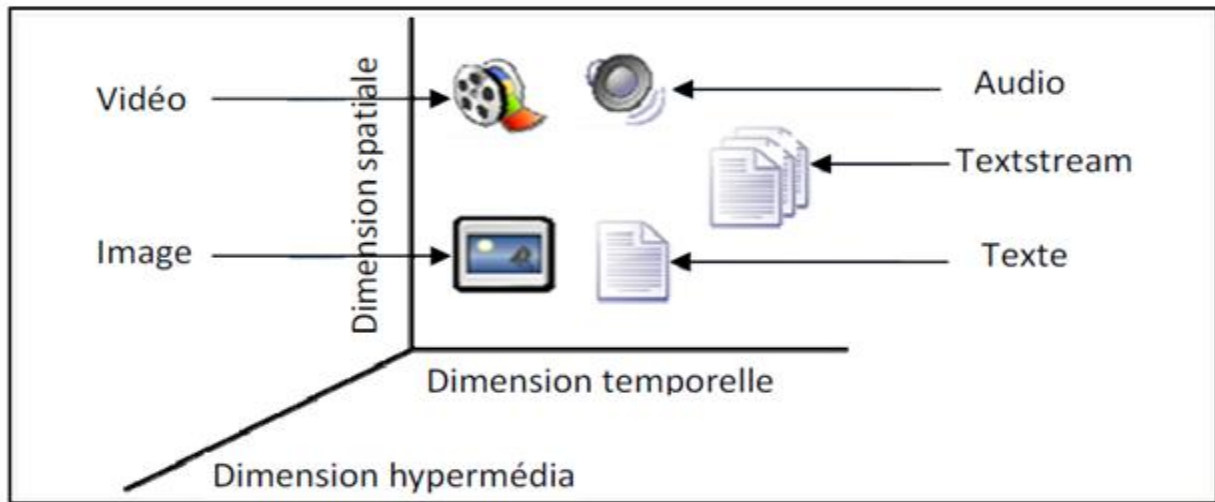


Figure II.8: Les quatre dimensions d'un DM

II.6.1 Dimension spatiale

Cette dimension concerne l'affectation dans le temps des composants multimédia d'un document aux ressources physiques (l'espace). Par exemple, un élément vidéo doit disposer d'un espace géométrique sur l'écran délimité par une certaine zone. De la même façon, un élément audio doit disposer d'un canal audio pour toute la durée de sa présentation. Cette opération de mise en correspondance, appelée formatage spatio-temporel, tient compte de différents paramètres du document qui peuvent dépendre aussi bien de sa structure temporelle que de sa structure logique. La figure II.9 présente la structure spatiale d'une présentation multimédia du DM de la ville d'Oran comportant trois objets visibles.



Figure II.9 : Dimension spatiale (droite) d'une présentation multimédia (gauche) du DM de la ville d'Oran

II.6.2 Dimension temporelle

La caractéristique majeure d'un DM est sa dimension temporelle. En effet, quelle que soit leur granularité, les objets multimédias sont reliés temporellement entre eux. Ceci définit un ordre global de présentation. Cet ordre peut être modélisé grâce à un scénario temporel. Un tel scénario est présenté dans la figure II.10 sous forme de Timeline.

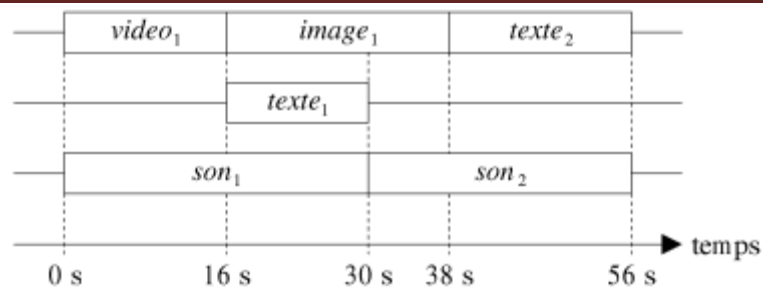


Figure II.10 : Dimension temporelle d'un DM

### II.6.3 Dimension hypermédia

La dimension hypermédia d'un document permet de décrire des liens qui peuvent exister entre certains médias, voire même entre différents documents. Ces liens dits hypermédia permettent de définir des relations de type sémantique entre des documents ou entre des parties de documents, comme les renvois ou les références. Définir plusieurs liens, c'est définir un réseau de documents qui constitue ainsi un support pour la navigation dans un grand espace d'informations à la manière du World Wild Web. Un lien est défini par une ancre de départ (un élément), une ancre d'arrivée (un autre élément) et le lien qui porte une certaine sémantique (renvoi, référence, annotation, etc.). Au niveau de l'interface utilisateur, un lien nécessite d'être activé explicitement par une interaction de l'utilisateur. Cette dimension est illustrée dans la figure II.11.

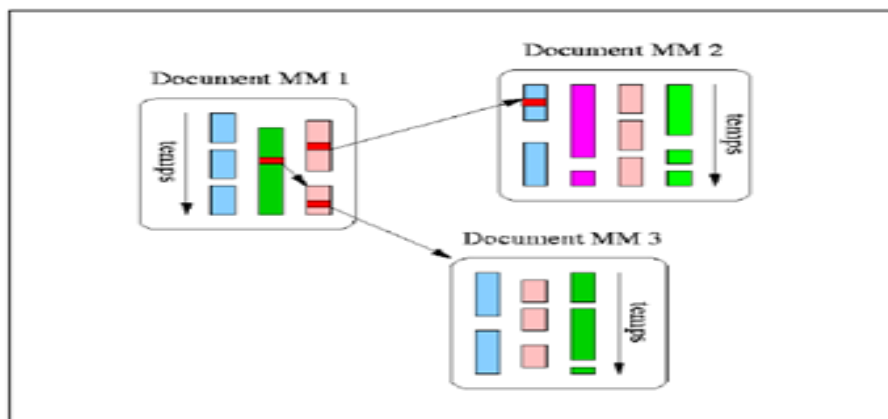
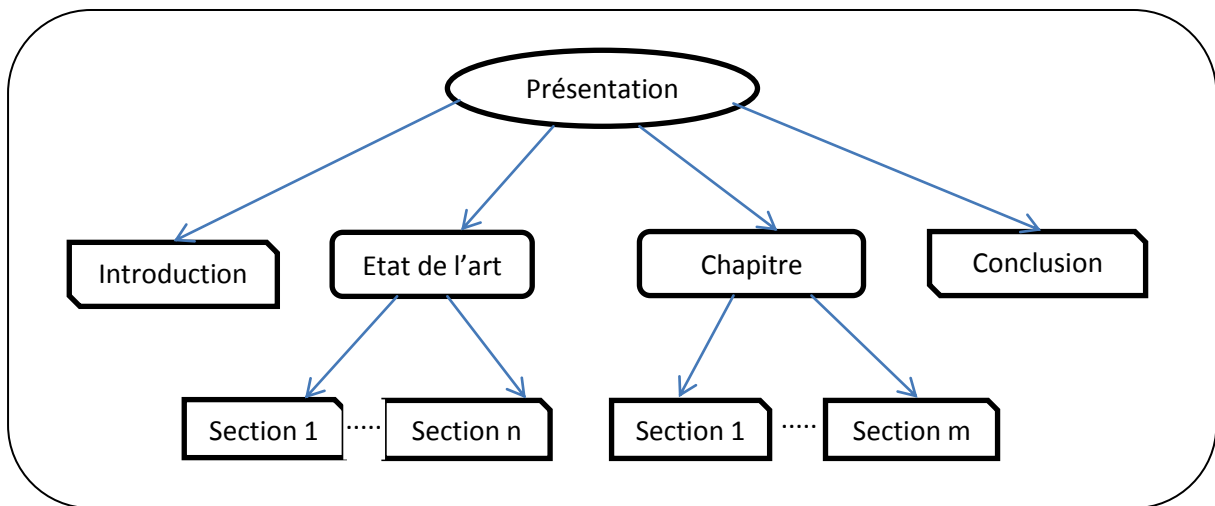


Figure II.11 : Dimension hypermédia d'un DM

### II.6.4 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.12 présente une structure logique d'un DM.



**Figure II.12 :** Dimension logique d'un DM

La racine de la structure logique représente l'ensemble du DM. Ce document peut se décomposer en plusieurs parties (Introduction, Chapitre . . .) et sous-parties (Section). Chaque partie ou sous-partie peut contenir un nombre fini d'objets multimédias (texte, image, vidéo, son . . .).

## II. 7 Domaine d'applications des DM

Selon Blakowski [Blak 96], un système multimédia est caractérisé par :

- Le nombre de média manipulés dans l'application
- La nature temporelle des média manipulés (discrets ou continus)
- Le niveau d'intégration des différents média

Un système multimédia est donc une application qui supporte le traitement de plusieurs média dont au moins un est de nature temporelle (c'est-à-dire un média continu).

Parmi les domaines qui utilisent les systèmes multimédia et, plus précisément, la notion de DM, l'enseignement assisté par ordinateur (EAO) est actuellement le plus important [Stem 97]. En effet, l'EAO peut tirer parti des caractéristiques de différents médias pour réaliser des supports pédagogiques attractifs grâce aux images, aux animations et au son, et qui soient de plus interactifs et adaptables aux élèves grâce aux fonctions de navigation hypermédia.

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

## II.8 Synchronisation des média

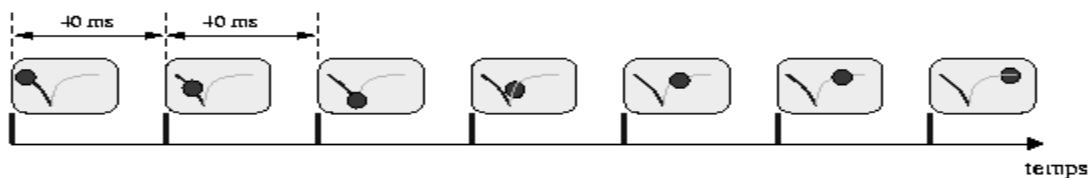
La synchronisation est une caractéristique importante des documents multimédias qui concrétise la sémantique d'une présentation multimédia conçue par un auteur. Dans le cas des objets médias temporisés, le facteur temps apparaît comme une dimension essentielle de l'information.

### II.8.1 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.13**).



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

### II.8.2 Synchronisation inter-média

La synchronisation inter-média est la composition entre les médias artificiels spécifiés explicitement par l'auteur du document. Ce sont les liaisons qui définissent le contenu de haut niveau d'abstraction, ou scénario de la présentation multimédia.

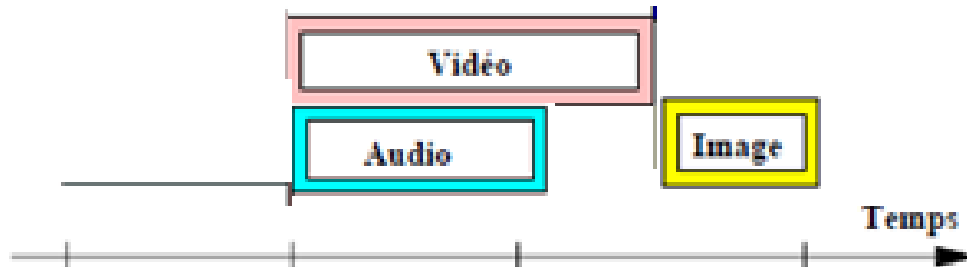
La synchronisation inter-média peut être synthétique (à gros grain) ou naturelle (fine). La granularité des éléments composés ainsi que les relations temporelles entre eux représentent les critères de classification.

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.14**).



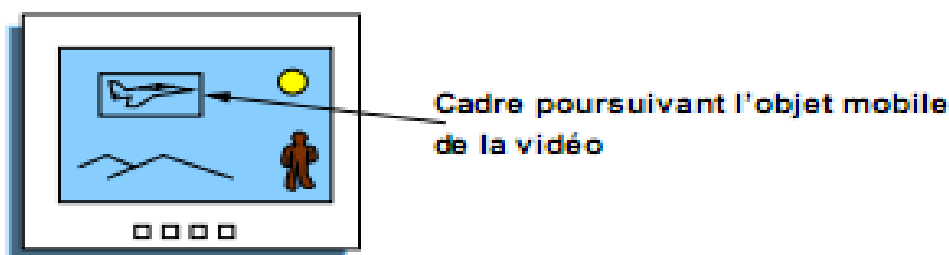
**Figure II.14** : Exemple d'une synchronisation inter-média

### II.8.3 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.15**. Dans cet exemple, un élément graphique (carré) est utilisé pour annoter un élément en mouvement dans une séquence vidéo (Avion) [Chen 95]



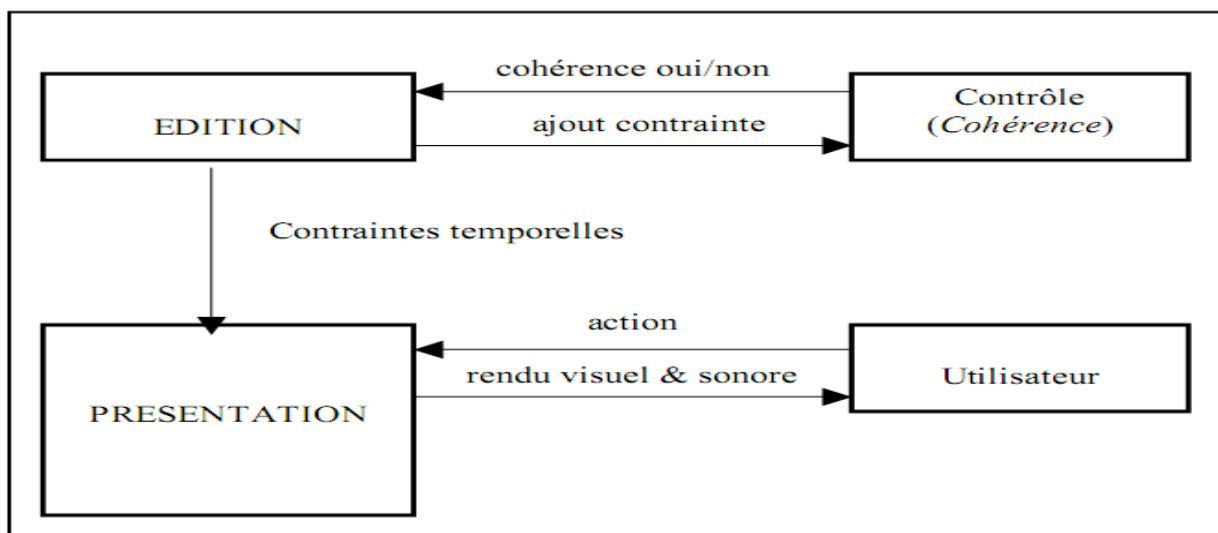
**Figure II.15** : Exemple d'une synchronisation des lèvres (*lip-synchronization*)

## II.9 Les présentations multimédia

Une présentation multimédia ou scénario est un ensemble de données regroupant plusieurs média temporels de types variés dans le but d'être présenté un nombre indéfini de fois d'une manière coordonnée spécifiée par un auteur au moyen d'un procédé quelconque.

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.16**.



**Figure II.16:** Cycle de vie d'un DM

### II.9.1 L'édition

En effet, il est souhaitable tout d'abord d'assister l'auteur dans la conception des documents, cette phase est appelée 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.9.2 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.

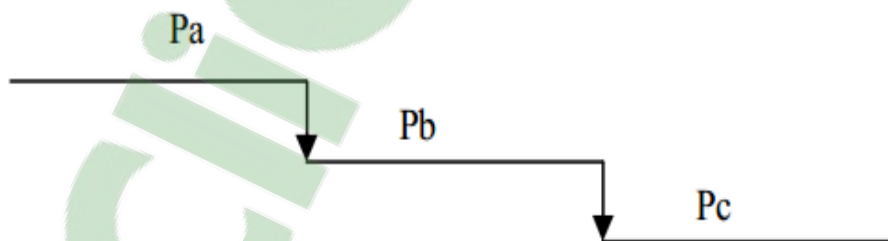
La création du scénario d'un document est un processus incrémental : l'auteur crée un premier scénario et doit être sûr de suite qu'il est cohérent, il le prévisualise, revient au scénario, ajoute/supprime une relation. Il est donc nécessaire à la fois de vérifier en temps-réel la cohérence temporelle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

En effet, la présentation d'un DM 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 DM.

### II.9.2.1 Cohérence qualitative

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

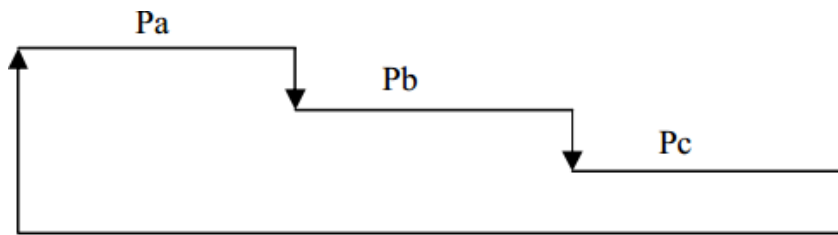
Exemple : soient trois médias 'a', 'b' et 'c', reconnus par leurs durées de présentations respectives:  $P_a$ ,  $P_b$  et  $P_c$ , et reliés par les relations *meets* (a, b) et *meets* (b, c) comme c'est schématisé dans la **Figure II.17** suivante :



**Figure II.17:** Exemple d'un scénario valide

Il est à remarquer qu'aucune incohérence n'est relevée sur le scénario de la **Figure II.17**.

**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.18**:



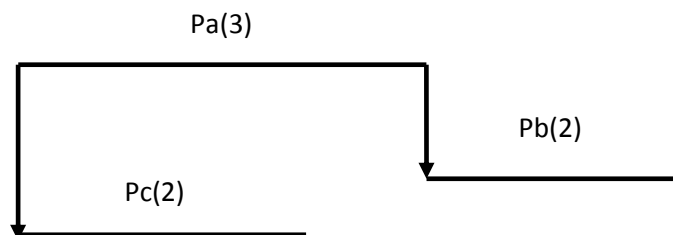
**Figure II.18 :** Exemple d'un scénario invalide

Dans la **Figure II.18**, la fin du scénario n'est jamais atteinte c'est donc un cas d'incohérence qualitative.

### II.9.2.2 Cohérence quantitative

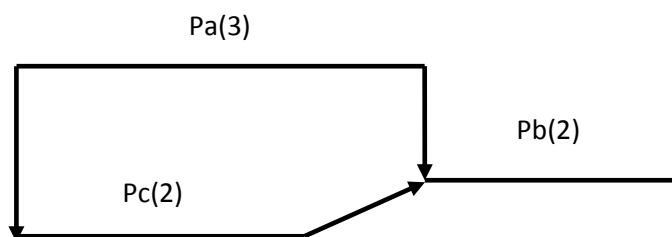
Cette consistance concerne la validité des combinaisons des relations temporelles parmi les objets médias de la présentation par rapport à leurs durées d'exécution.

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.19** suivante :



**Figure II.19:** 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.20** suivante :



**Figure II.20:** 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.9.3 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 DM correspond bien au scénario désiré.

### **II.10 Les standards de structuration des DM**

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 DM.

#### **II.10.1 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.10.2 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.10.3. 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 DM 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.10.4 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.10.5 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 [Ccit 92], ISO ainsi que d'autres normes et architectures et des applications définies par l'utilisateur.

### **II.10.6 SMIL**

SMIL (*Synchronized Multimedia Integration Language*) est un standard du W3C [1], et c'est aussi un langage d'édition et de présentation des DM 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).

SMIL fait l'objet de notre travail et sera donc étudié en détaille dans le chapitre III.

## II.11 Conclusion

Partout où l'image et le son s'avèrent être des éléments décisionnels, le recours au multimédia est voué à un développement rapide. Cependant, cette technologie de multimédia n'est réellement exploitable que si nous disposons de standards (en cours d'élaboration actuellement) et d'un modèle puissant permettant d'exploiter toute la richesse et la complexité du multimédia.

Par rapport à des données classiques (données formatées ou texte), les données issues des technologies multimédia (images et vidéo) requièrent une nouvelle approche. Tout d'abord, du point de vue de l'utilisateur, ces données peuvent être abordées selon différents aspects: temporel, spatial et interactif. L'objectif de l'utilisateur est de disposer de la bonne information sous la forme la plus adaptée aux contraintes temporelles et spatiales.

Si les « nouveaux » média numériques comme la vidéo, l'audio sont des composants essentiels des stations de travail, ils ne pourront être banalisés qu'à partir de la disponibilité :

- d'application et standards utilisant ces média,
- de canaux de télécommunication et de distribution adéquats.

Ces deux facteurs fixent la vitesse à laquelle évolue cette nouvelle technologie. En ce qui concerne notre domaine de recherche, seul le premier facteur nous intéresse. Mais pour que cette technologie soit réellement utilisable il nous apparaît nécessaire de gérer, d'accéder et surtout d'exploiter des DM de plus en plus complexes et de plus en plus volumineux en utilisant des standards comme le langage SMIL. Ce dernier 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.

La complexité de la synchronisation temporelle et spatiale des présentations SMIL rend difficile voire impossible de garantir la validité d'un scénario en se basant sur des méthodes informelles ou empiriques. Ainsi, notre recherche se focalise sur un aspect particulier de la gestion des documents SMIL, à savoir la modélisation et la validation des documents SMIL que nous allons étudier dans le chapitre III suivant.

## Chapitre III : Analyse et validation des documents SMIL

### III.1 Introduction

SMIL acronyme (*Synchronized Multimedia Integration Language*) [1] est un langage déclaratif recommandé par le World Wide Web Consortium (W3C) pour les documents multimédia (DM) depuis 1998 qui permet d'intégrer des objets médias (images, sons, textes, vidéo, animations, hypertexte), des structures spatiales et des scénarios temporels dans un document. Il supporte aussi des animations, des transitions et même des synchronisations entre des objets médias. 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.

SMIL est un format d'intégration, c'est-à-dire qu'il ne décrit pas les médias faisant partie d'une présentation multimédia mais plutôt leur composition temporelle et spatiale ainsi que les hyperliens qui lient ces médias. En effet, un auteur d'un document SMIL peut :

- décrire le comportement temporel de la présentation. SMIL utilise les opérateurs `seq` et `par` pour spécifier qu'un ensemble d'objets est joué respectivement en séquence et en parallèle. On peut spécifier explicitement la durée d'un média. La date de début ou de fin d'un média peut être spécifiée par un délai par rapport à celle d'un autre média. Enfin, des attributs de synchronisation (ex : terminaison du plus court ou du plus long média) sont définis pour exprimer la terminaison de la composition temporelle ;
- décrire le placement des objets médias sur l'écran pendant la présentation. Sur la fenêtre principale de présentation, SMIL spécifie des régions dont la position et la taille sont exprimées en valeur absolue ou en pourcentage de la taille de la fenêtre principale ;
- associer des hyperliens aux objets médias. La désignation de la destination d'un lien est effectuée en termes d'adresse URL.

SMIL permet de caractériser les exigences spatiales, structurelles ainsi que temporelles d'une présentation multimédia, qui doit être jouée sur un large éventail d'équipements, notamment de terminaux mobiles. Ainsi, les apports principaux de SMIL se situent au niveau de la portabilité du langage de spécification des DM et de la spécification de la navigation.

Dans SMIL, les ancres des liens hypermédias peuvent être définies activables pendant des intervalles de temps ou sur des sous-régions de la fenêtre où l'objet média est affiché.

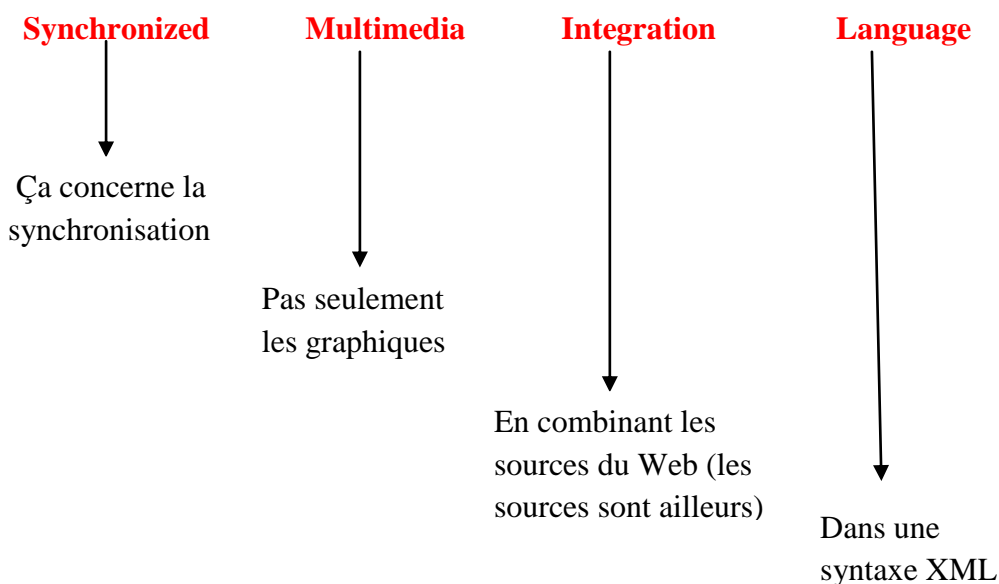
Plusieurs sociétés informatiques ont déjà annoncé leur support pour SMIL [Laya 97]. Ceci démontre bien le rôle important que joue SMIL dans le domaine de la spécification et de l'échange de DM.

Dans ce chapitre nous nous intéressons à l'étude du langage SMIL 2.0 : ses avantages et ses inconvénients. Nous présentons alors les approches de modélisation et de validation des documents SMIL.

### III.2 Définition de SMIL

SMIL qui est l'abréviation de *Synchronized Multimedia Integration Language* (**Figure III.1**) est un langage déclaratif recommandé par le W3C permettant la spécification des présentations multimédias interactives sur le Web, par l'intégration et la synchronisation des contenus multimédias diversifiés (image, son, vidéo, texte, animation, flux de vidéo (streaming)) .

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 Historique de SMIL

Le W3C, consortium mondial regroupant des industriels et des centres de recherche, dont l'objectif est de développer et de promouvoir le Web, propose un format de DM nommé SMIL [1]. L'objectif de ce format est de permettre l'écriture de documents Web dont les objets multimédia sont

synchronisés dans le temps. On peut le voir comme une extension du format HTML qui va permettre de rendre les documents sur le Web plus dynamiques sans pour autant nécessiter de la programmation.

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 DM 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).
- Permettre l'interaction événementielle avec l'utilisateur. Offrir des possibilités d'adaptation de contenu local avec l'élément **switch**.

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 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 2.0.

#### III.4 Pourquoi 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 :

#### **A. 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] ;
- SMOX Pad 1.2.1.2 [11] pour SMIL 2.0 ;

#### **B. Les lecteurs**

- QuickTime [12] pour SMIL 1.0 ;
- RealOne Player [13] pour SMIL 2.0 ;
- 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 [2]

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 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 2.0//EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/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"/>
<region id="region3" left="282" top="174" width="222" height="174"/>
<region id="region4" left="282" top="324" width="252" height="138"/>
</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">
```



```

<audio src="audio1"/>
</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.
- **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 :

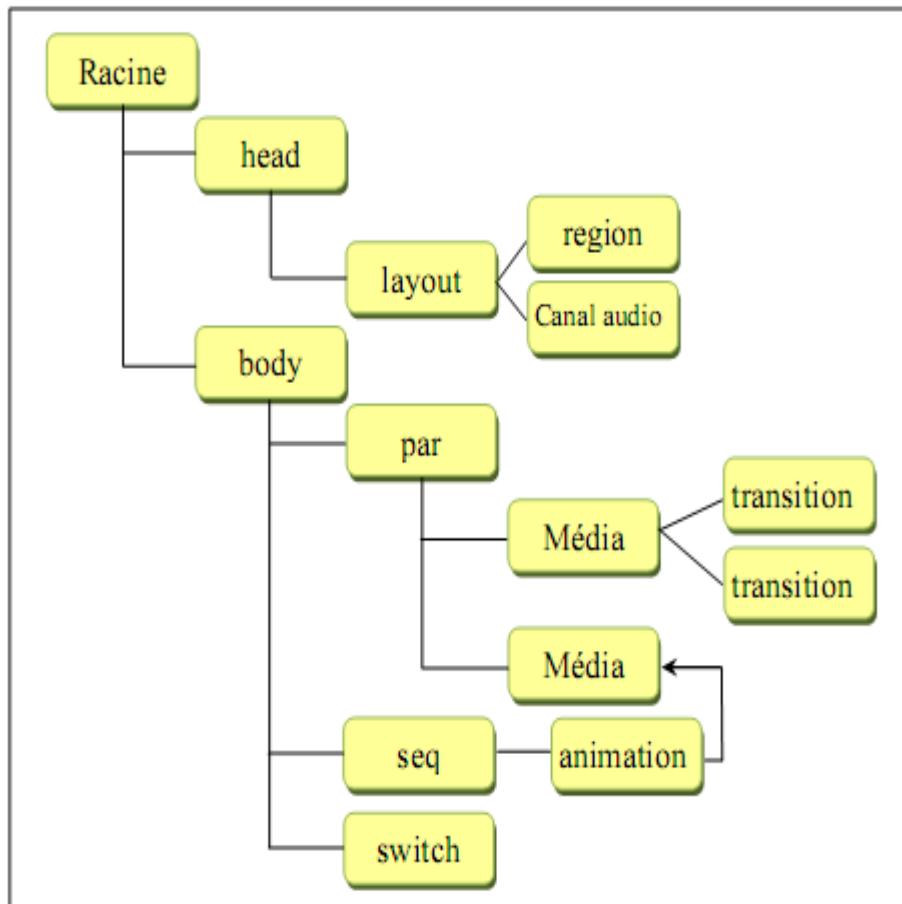


Figure III. 3 Structure d'un document SMIL

### 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 1.0 permettait aux auteurs de décrire des documents avec un vocabulaire très basique en utilisant l'élément **meta**. Le module **Méta-Information** de SMIL 2.0, 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 1.0.

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 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 Le 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

Élément média	Fonction et extensions
<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

Attribut	Fonction	Type de valeur
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 région
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	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 porteur de cet attribut est ignoré.	
System-language		
System-screen-size		

### III.5.5 Manipulation du temps

Ce module introduit de nouveaux attributs pour la manipulation avancée du comportement du temps, comme le contrôle de la *vitesse* ou le *régime* du temps pour un élément. Ces *manipulations du temps* sont particulièrement adaptées aux médias d'animation et aux médias non linéaires, ou médias discrets. Les types de média continus ne gèrent pas tous totalement les manipulations du temps. Par exemple, un flux vidéo MPEG 1 ne gère pas, en général, une lecture en arrière. Un mécanisme de repli est décrit pour ces types de média. Quatre attributs *accelerate*, *decelerate*, *autoReverse* et *speed* ajoutent une gestion des manipulations du temps aux modules de temporisation de SMIL, incluant le contrôle sur la vitesse d'un élément et la gestion de l'accélération et de la décélération. Une définition des mécanismes de repli raisonnables est fournie pour les lecteurs de média qui ne gèrent pas les manipulations de temps.

### III.5.6 Le temps et la 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.5.6.1 Les attributs de temporisation

**SMIL2.0** offre l'ensemble d'**attributs** existant dans **SMIL1.0** 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.

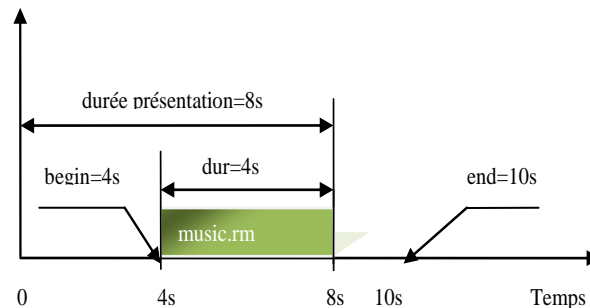
- **L'attribut begin** : Cet attribut sert à lancer l'objet multimédia à un moment précis de l'axe du temps de la présentation. Par conséquent, si l'attribut **begin** est utilisé dans un groupe, le temps qui lui sera affecté au début du groupe. La valeur de cet attribut peut *être déterministe ou non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

**Exemple** : Durée d'un élément multimédia définie par un temps de début explicite, une durée et un temps de fin explicite.

```
<audio src="music.rm" region="main" begin="4s" dur="4s" end="10s"/>
```

**Remarque** : Si la durée de présentation (*begin+dur*) est supérieure à *end* alors la durée de la présentation de l'élément est égale à la valeur de *end*, c'est-à-dire la présentation de l'élément s'arrête à la valeur de *end*.

Dans l'exemple de la **Figure III.3** ci-dessous, si **dur=8s** alors la durée de présentation est égale à **12s**, on a la durée de la présentation supérieure à **end**, dans ce cas, l'élément "music.rm" va s'achever à **10s**, et non à **12s**.



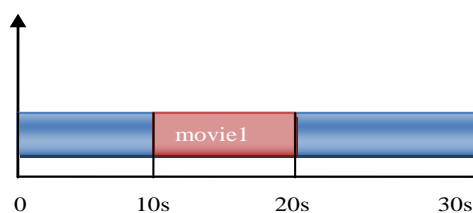
**Figure III. 4** Durée d'un élément multimédia

### III.5.6.2 Les attributs clipBegin et clipEnd

*ClipBegin* indique l'instant interne de l'élément **multimédia continu** à partir duquel sa diffusion commence. La syntaxe de cet attribut est semblable à celle de *begin*. Par contre, *clipEnd* indique l'instant interne de l'élément multimédia à partir duquel sa diffusion s'arrête, la syntaxe de cet attribut est semblable à celle de *end*.

Dans l'exemple de la **Figure III.4**, nous avons une vidéo "**video1.mpeg**" d'une durée de plus de **30s**, on ne veut jouer que le morceau "movie1" d'une durée de **10s** entre **10** et **20s**.

```
<video id="movie1" src="video1.mpeg" clipBegin="10s" clipEnd="20s"/>
```



**Figure III. 5** Illustration des attributs clipBegin, clipEnd

### III.5.7 Les attributs repeatCount et repeatDur et fill

SMIL a introduit l'attribut *repeat* qui est utilisé pour répéter un élément média ou tout un conteneur de temps. SMIL 2.0 introduit deux nouveaux attributs pour contrôler les fonctionnalités de répétition qui supplantent l'attribut *repeat* de SMIL 1.0 qui sont **repeatCount** et **repeatDur**. Ces attributs fournissent une sémantique qui correspond mieux aux cas d'utilisation typiques et offrent un meilleur contrôle de la durée des répétitions.

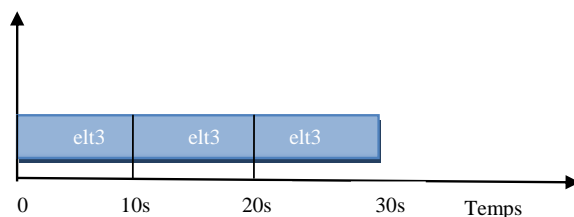
- **repeatCount** : indiquer le nombre d'itérations de la durée simple.
- **repeatDur** : indique la durée totale de la répétition d'un élément média ou conteneur.



Dans l'exemple de la **Figure III.5**, l'élément "elt3" est joué 3 fois de suite.

```

```



**Figure III. 6** Illustration des attributs repeatCount et repeatDur

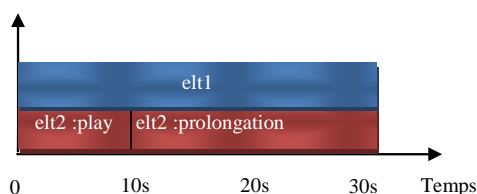
L'attribut **fill** : Permet à l'auteur de spécifier qu'un élément devrait être prolongé au-delà de sa durée active en *gelant* son état final. Il autorise les attributs suivants (cf. **Tableau III.3**) :

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

Valeur	Fonction
Remove	Spécifie que l'élément disparaîtra de l'écran.
Freeze	Spécifie que l'élément sera gelé. Le conteneur de temps parent de l'élément détermine combien de temps l'élément sera gelé.
Hold	Spécifie que l'élément est toujours gelé jusqu'à la fin de la durée de l'élément conteneur parent.
Transition	Spécifie que l'élément est retiré à la fin de la transition.cette valeur n'est autorisée que pour les éléments directement associés à des média.
Auto	Le comportement de l'élément dépend des attributs de temporisation.

L'exemple de la **Figure III.7** montre comment l'objet "elt2" est prolongé jusqu'à la durée de fin de "elt1".

```
<par>
  <video id="elt1" src="movie2.mpeg" dur="30s"/>
  < video id="elt2" src="vid1.mpeg" dur="10s" fill="freeze"/>
</par>
```



**Figure III. 7:** Illustration de l'attribut fill

### III.5.8 Les éléments de synchronisation

Appelés aussi conteneurs de temps, ils acceptent comme fils les éléments média, l'élément switch et aussi d'autre conteneurs de temps. Ces éléments sont :

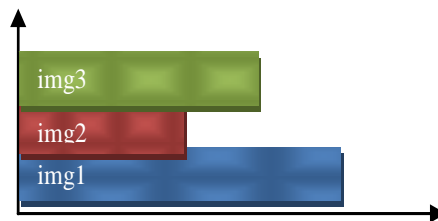
- L'élément **<seq>** joue les éléments enfants les uns à la suite des autres en séquence.
- L'élément **<excl>** joue les éléments enfants un à la fois mais sans imposer l'ordre.

- 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.8** permet à trois images de s'afficher en parallèle.

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



**Figure III. 8** 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.4** suivant :

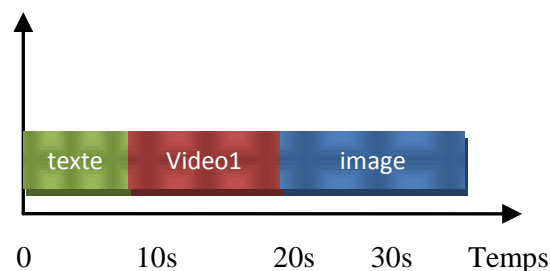
**Tableau III.4:** Les valeurs acceptées par l'attribut **endsync**

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 (durée)	La durée implicite du conteneur de temps se termine quand la durée intrinsèque du média se termine.
Id (valeurr)	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.9** suivant permet d'afficher séquentiellement trois éléments média.

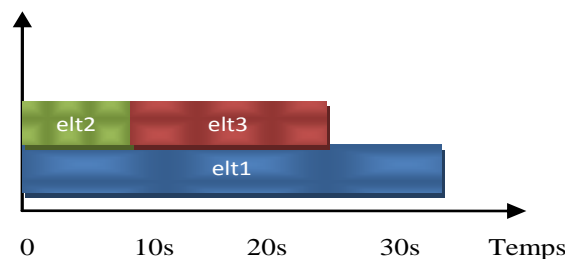
```
<seq>
  <text id="texte" src="txt.rt"/>
  <video id="video1" src="video1.mpeg"/>
  
</seq>
```



**Figure III. 9** 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.10** lance parallèlement la lecture de deux groupes d'éléments multimédia. Le premier groupe ne comprend qu'un élément, alors que le deuxième en comprend deux qui sont lus séquentiellement.

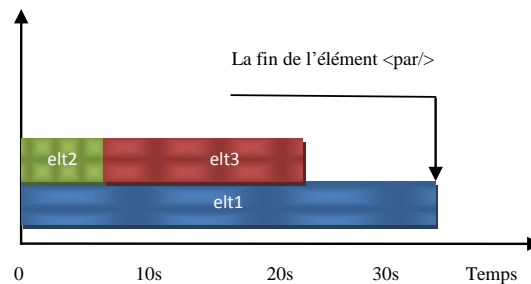
```
<par>
  <audio id="elt1" src="chanson.mp3"/>
  <seq>
    <video id="elt2" src="video1.rm" region="r1"/>
    <video id="elt3" src="video2.rm" region="r1"/>
  </seq>
</par>
```



**Figure III. 10** Exemple d'éléments joués en parallèle et en séquence

Dans l'exemple de la **Figure III.11**, 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. 11** Exemple de synchronisation en enchaînement

### III.6 Le module Linking

Le module de lien, définit des éléments et attributs du document **SMIL 2.0** 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/> 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 <switch>.

**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>
```

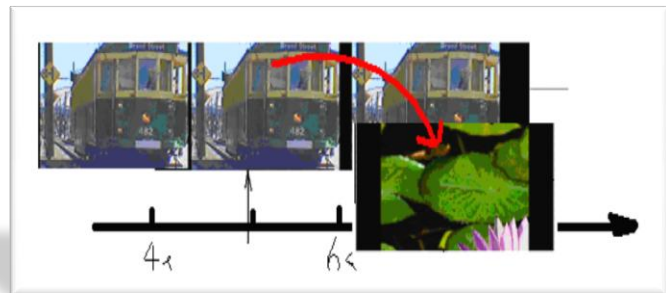
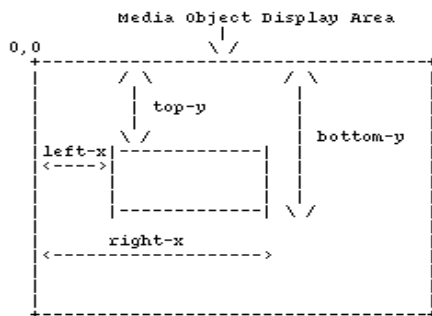
**B- L'élément <area/>** : La sémantique de l'élément **<area/>** dans SMIL 2.0 est la même que celle dans HTML du fait qu'elle peut spécifier qu'une partie spatiale d'un objet visuel peut être choisie pour déclencher l'apparition de la destination du lien. C'est-à-dire, elle rend la partie spatiale 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.12** 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. 12** Exemple d'une partie temporelle cliquable

### III.7 Le module Content Control

Le module de contrôle de contenu de SMIL 2.0 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 SMIL2.0 se partage en quatre modules :

- **Le module BasicContentControl** : Contient les **attributs de test** du système définis dans SMIL1.0. SMIL2.0 inclut la fonctionnalité de l'attribut de test de SMIL 1.0 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.5** suivant :

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

Attribut	valeur	Fonction
<code>SystemLanguage</code>	Liste de noms de langues (Fr, En,...)	Effectue un test sur la langue configurée par l'utilisateur.
<code>SystemBitrate</code>	Bites/seconds	Choix du type de connexion, contrôle le débit de transmission.
<code>SystemCaptions</code>	On ou off	Ignoré si l'utilisateur ne veut pas de sous-titre.
<code>SystemRequired</code>	Nom de l'extension	Reconnaissance ou non d'une extension.
<code>SystemScreenSize</code>	Taille: hauteur x largeur (en pixels)	Spécifie la taille minimale de l'écran.
<code>SystemScreenDepth</code>	Nombre de bits de codage 1, 4, 8, 16, 32	Spécifie la résolution minimale.
<code>SystemOverdupOrCaption</code>	<b>Caption</b> ou <b>overdub</b>	Sous-titrage ou non.

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 2.0, 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 2.0 contiennent les éléments et les attributs qui offrent la fonctionnalité de transition à SMIL 2.0. 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.

### III.10 Analyse des document SMIL

De nos jours, la notion de DM 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 DM 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 bas 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 2.0 [2] qui est une amélioration de SMIL 1.0 de telle sorte que la présentation devienne plus du 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 2.0 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 2.0 sont : Structure, Media, Content Control, Linking, Timing and Synchronization, Animation, Time Manipulations, Transition Effects et Metainformation.

### III.10.1 Structuration

La structuration d'un DM 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 tout 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 DM et aux différents détails qui le composent s'ajoutent des 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). SMIL2.0 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'événement 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 DM 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

La flexibilité du modèle temporel de SMIL 2.0 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

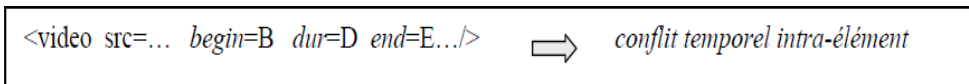
Une incohérence temporelle est la conséquence de la définition de relations de synchronisation temporelle qui ne peuvent être satisfaites simultanément au moment de la présentation. Les recherches menées en [Yang 00] ont montré que les incohérences temporelles en SMIL peuvent être définies comme des 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.13**: 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. 13** Conflit temporel intra-élément

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

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

### 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. 14**, 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. 14** Conflit temporel inter-éléments

Un autre exemple qui montre une incohérence temporelle inter-éléments 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 DM apparaissent comme une nouvelle génération d'applications informatiques dont la conception à grande échelle met en jeu leurs consistances. 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 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'exemple 1 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>
```

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 Les approches de modélisation et 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 DM veut tout d'abord s'assurer de la validé 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.

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 utilisé 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 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.

**Tableau III.7** : Comparaison entre les différentes approches de modélisation et de validation des documents SMIL

<b>Auteurs</b>	<b>Vérification temporelle</b>	<b>Formalisme utilisé</b>	<b>Localisation des erreurs</b>
<b>Little et Ghafor. 1990</b>	oui	RdP + logique d'intervalles temporels	oui
<b>Yang 2000</b>	oui	RdP	oui
<b>Jourdan et al. 2001</b>	oui	automate	non
<b>Chung et Anil 2003</b>	non	RdP	non
<b>Maazouz et al. 2006</b>	oui	RdP	non
<b>Sampaio et Courtiat. 2004</b>	oui	automate	non
<b>Bossi et Gaggi. 2007</b>	oui	Logique d'Hoare	oui
<b>Bouyakoub et Belkhir 2007</b>	oui	RdP	oui

### III.13 Conclusion

Dans ce chapitre nous avons présenté le langage SMIL 2.0 : sa syntaxe et sa sémantique, et nous avons discuté les différentes approches de l'état de l'art pour sa modélisation et sa validation.

Le langage SMIL 2.0 est un modèle de document basé sur le langage XML qui permet la représentation des présentations multimédia interactives. Le langage permet principalement de décrire l'organisation temporelle des objets média, les relations entre les objets utilisés, la disposition spatiale des objets ainsi que les liens qui peuvent être utilisés dans une présentation multimédia. Il décrit les fonctionnalités avancées, y compris les transitions, l'animation et les métadonnées intégrées, et un document SMIL doit alors être valide lors de sa présentation. Pour cela, une attention particulière est faite quant à la formalisation du langage SMIL en utilisant les RdP temporels pour pouvoir valider les incohérences temporelles d'un document multimédia avant sa présentation.

Dans le chapitre IV suivant, nous allons détailler la conception et la réalisation de notre outil de modélisation et de validation des documents SMIL 2.0 et qui est basée sur les réseaux de Petri temporelle de Merlin (t-RdP) [Merl 74].

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

### **IV.1 Introduction**

Depuis son apparition, le langage SMIL suscite un intérêt croissant, notamment dans le cadre de l'édition de DM synchronisés interactifs. Le standard SMIL, tel qu'il a été défini par le W3C, est un langage basé sur XML de description de DM 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 les réseaux de Petri de Merlin [Merl 74]. 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.

### **IV.2 Notre proposition**

Dans le domaine plus spécifique de la modélisation multimédia, les modèles existants semblent insuffisants en terme de conceptualisation et n'abordent pas tous les aspects nécessaires des présentations multimédia. Un langage majeur comme SMIL propose par exemple une synchronisation temporelle complète mais aussi complexe et dont la sémantique prête parfois à confusion. En outre, SMIL est un langage de scripts et ne supporte pas l'analyse au même degré que les systèmes à base des réseaux de Petri (RdP en abrégé).

Partant de la considération que les RdP offrent un des cadres formels les plus complets pour la modélisation de la synchronisation multimédia [Litt 90], [Sena 94], nous avons adopté ce formalisme comme modèle de base pour concevoir notre outil de modélisation et de validation des présentations SMIL.

Dans le cadre du multimédia, les RdP se prêtent bien à la modélisation des composants séquentiels et parallèles d'un DM, ainsi que leurs interactions. Cependant, un RdP classique ne peut pas modéliser la composante temporelle essentielle pour la modélisation de la synchronisation multimédia. Pour cela, des extensions temporelles du modèle RdP ont été proposées [Merl 74] [Khan 97] afin d'enrichir le modèle de base par l'introduction d'une dimension temporelle lui permettant ainsi de modéliser le temps.



Nous pensons donc que notre contribution dans ce chapitre peut apporter des éléments cohérents permettant de clarifier la spécification des présentations multimédia, que ce soit à travers le langage lui-même ou bien des notions du modèle temporel et de synchronisation, notions qui peuvent être appliquées à d'autres langages. Ceci permettrait d'exécuter le langage d'édition sur des players SMIL, de comparer son pouvoir d'expression et d'explicitier les points sensibles et difficiles de la sémantique de SMIL.

Notre outil est basée sur le modèle des réseaux de Petri temporels de Merlin [Merl 74] (t-RdP en abrégé) qui permet de modéliser et valider les documents SMIL. Ce dernier sera détaillé dans la **section IV.2.2**.

### IV.2.1 Les Réseaux de Petri simples

#### Définition d'un RdP simple

Les réseaux de Petri (RdP) est un outil mathématique et graphique qui a été introduits par Carl Adam Petri en 1962 [Petr 62] pour la modélisation des systèmes concurrents, asynchrones, distribués, parallèles et non déterministes. Différents facteurs contribuent à son succès et incluent sa nature graphique et la base mathématique solide sur laquelle il repose.

Les RdP présentent des avantages indéniables en matière d'analyse de la structure et du comportement des systèmes à évènements discrets.

Un RdP est un graphe orienté biparti valué  $\langle P, T, \text{Pre}, \text{Post} \rangle$  avec :

- $P$ , un ensemble fini de places,  $\{p_1, p_2, \dots, p_n\}$ .
- $T$ , un ensemble fini de transitions,  $\{t_1, t_2, \dots, t_m\}$ .

Ces deux ensembles formant les sommets du réseau, avec  $P \cap T = \emptyset$ .

- $\text{Pre} : P \times T \rightarrow \mathbb{N}$  est l'application d'incidence avant, correspondant aux arcs directs reliant les places aux transitions.
- $\text{Post} : P \times T \rightarrow \mathbb{N}$  est l'application d'incidence arrière, correspondant aux arcs directs reliant les transitions aux places.

Les places sont représentées par des cercles et les transitions par des barres horizontales (voir **Figure IV.1**).

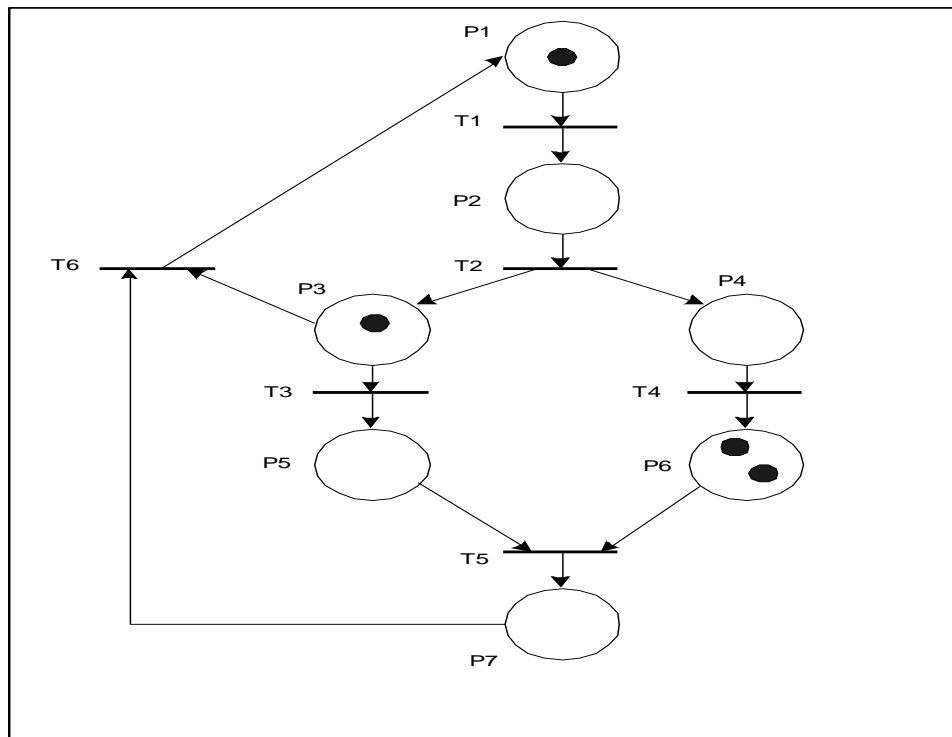


Figure IV.1 : Réseau de Petri marqué

Les places traduisent une action quelconque, les arcs montrent les différents chemins possibles entre les places et les transitions permettent la synchronisation de ces actions. Lorsque deux places sont reliées par des arcs à une même transition, celle-ci ne sera franchie que lorsque les deux actions auront été terminées.

Pour simuler le fonctionnement de tels réseaux, des marques ou jetons évoluent de places en places. Lorsqu'un jeton apparaît dans une place, l'action associée à celle-ci peut se dérouler. Lorsque l'action est terminée, la transition qui suit la place devient validée ou franchissable. Le franchissement s'effectue, et le jeton passe de la place d'entrée de la transition à la place de sortie de cette même transition. On peut parfois rencontrer la notion de conflit. Quand une place est suivie de deux transitions, il faut décider du chemin que l'on va emprunter par la suite. Cette notion de conflit peut être expliquée par le problème du partage des ressources.

#### IV.2.2 Les réseaux de Petri temporels

L'extension temporelle des RdP au temps s'exprime sous la forme d'un intervalle temporel associé soit aux places (i.e. les RdP p-temporel) [Khan 97], soit aux transitions (i.e. les RdP t-temporel) [Merl 74].

Comme les RdP t-temporels (t-RdP) de Merlin sont à la base de notre outil, nous avons jugé nécessaire de donner les informations suivantes :

### Définition d'un t-RdP

Les réseaux de Petri temporels (*Time Petri Nets*) ont été créés en même temps que les réseaux temporisés par Merlin dans sa thèse [Merl-74].

Les RdP t-temporels (t-RdP) est un outil mathématique adéquat pour la spécification et la vérification des systèmes contraints au temps.

Dans les RdP t-temporels, l'intervalle  $[a, b]$  associé à la transition  $t$  est relatif au moment où la transition devient validée, où  $a$  et  $b$  désignent respectivement les temps minimum et maximum de franchissement de la transition  $t$ . Supposons que  $t$  soit validée à l'instant  $\tau$ , alors elle peut être franchie dans l'intervalle matérialisé par les quantités  $(a + \tau)$  et  $(b + \tau)$ , sauf si elle est « désensibilisée » à cause du franchissement d'une autre transition avec laquelle elle était en conflit.

Un RdP t-temporel est un 6-uplet  $(P, T, \text{Pré}, \text{Post}, M_0, IS)$  tel que :

- $(P, T, \text{Pré}, \text{Post}, M_0)$  est RdP
- $IS: T \rightarrow \mathbb{Q}^+ \times (\mathbb{Q}^+ \cup \infty)$  avec  $\mathbb{Q}^+$ : est l'ensemble de nombre rationnels positifs

### Règle de fonctionnement d'un t-RdP

Le comportement d'un RdP t-temporel est caractérisé par la notion d'état qui est représentée par :

- Un marquage courant.
- Pour chaque transition validée, de son intervalle de tir.

L'intervalle de tir considéré sera alors qualifié d'intervalle dynamique. En effet, ce dernier peut différer de l'intervalle statique, comme l'illustre la **Figure IV. 2**.

Considérons le réseau suivant où le jeton vient d'arriver dans la place  $p_1$ . Pour la représentation fournie, où seule la transition  $t_1$  est sensibilisée, son intervalle dynamique correspondra à son intervalle statique [2, 6].

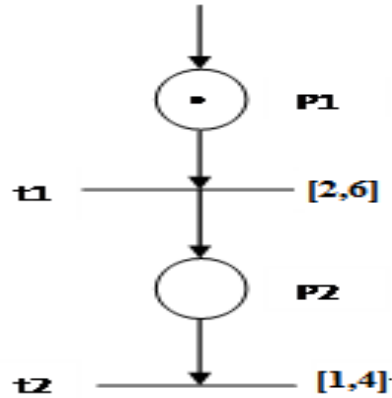


Figure IV.2 : Exemple d'un RDP t-temporel

Supposons maintenant qu'après  $\tau$  unités de temps ( $\tau < 6$ ) la marque soit toujours contenue dans  $p_1$ , l'intervalle dynamique associé à  $t_1$  correspondra alors à l'intervalle défini par  $[\max(0, 2 - \tau), 6 - \tau]$ .

L'état d'un RDP t-temporel est défini par une paire  $E = (M, I)$ , telle que :

- $M$ , est un marquage.
- $I$ , une application intervalle de tir, associant à chaque transition du réseau l'état initial

$E_0 = (M_0, I_0)$  tel que :

- $M_0$ : est le marquage initial.
- $I_0$ : est l'application associée à chaque transition est définie par :

$$I_0(t) = \begin{cases} IS(t) & \text{Si } \forall p \in P, M(p) \geq \text{Pre}(p, t) \\ \emptyset & \text{Sinon} \end{cases}$$

Une transition  $t$  est franchissable à un instant  $\tau$  depuis un état  $E = (M, I)$  si et seulement si les deux conditions suivantes sont satisfaites :

- 1- la transition  $t$  est validée par le marquage  $M$  au sens des RDP,
- 1)  $\tau$  est compris (bornes incluses) entre la date de tir au plus tôt de  $t$  et la plus petite des dates de tir au plus tard des autres transitions validées (dans l'état  $E$ ).

#### Algorithme de calcul de l'état suivant

Le tir d'une transition  $t$  franchissable depuis un état  $E = (M, I)$  à un instant  $\tau$ , conduit au nouvel état  $E' = (M', I')$ , déterminé par l'équation:

$$\forall p \in P, M'(p) = M(p) - \text{Pre}(p, t) + \text{Post}(p, t)$$

Les nouveaux intervalles de tir  $I'$  :

- Pour toutes les transitions  $t_i$  non sensibilisées par le marquage  $M'$ ,  $I'_i = \emptyset$ ,
- Pour toutes les transitions  $t_i$  sensibilisées par le marquage  $M'$ , on distingue :

- i. Pour toutes les transitions  $t_j$  sensibilisées par les marquages  $M$  et  $M'$  non en conflit avec la transition  $t$  :  $I' = [\max(0, a_j - \tau), b_j - \tau]$ ,  
Avec  $[a_j, b_j]$  l'intervalle statique associé à la transition  $t_j$ ,
- ii. Pour toutes les transitions  $t_k$  nouvellement sensibilisées par le marquage  $M'$ ,  
 $[a_k, b_k]$ .

### IV.2.3 Modélisation d'un document SMIL

Le processus de traduction de documents SMIL suppose que le document source est syntaxiquement correct (au sens XML). Dans notre travail, nous nous intéressons aux aspects temporels et hypermédia d'un document SMIL.

Trois éléments de synchronisation temporelle doivent être modélisés par le modèle t-RdP : les éléments de composition temporelle  $\langle seq \rangle$  et  $\langle par \rangle$ ,  $\langle excl \rangle$ , les éléments multimédias de base, et les attributs de synchronisation ( $begin$ ,  $end$ ,  $dur$ ,  $endsync$  pour le groupe  $\langle par \rangle$  et  $\langle seq \rangle$ ,  $\langle excl \rangle$ ). Dans le cadre de l'hypermédia, la navigation au sein d'un document SMIL est définie à l'aide des éléments  $a$  et  $anchor$  (ou  $area$  dans SMIL2.0).

Dans notre travail, nous avons modélisé un document SMIL ainsi que ses éléments:  $\langle seq \rangle$ ,  $\langle par \rangle$ ,  $\langle excl \rangle$  en s'inspirant de [Yang 00], [Maaz 06] et [Bouy 07] comme suit:

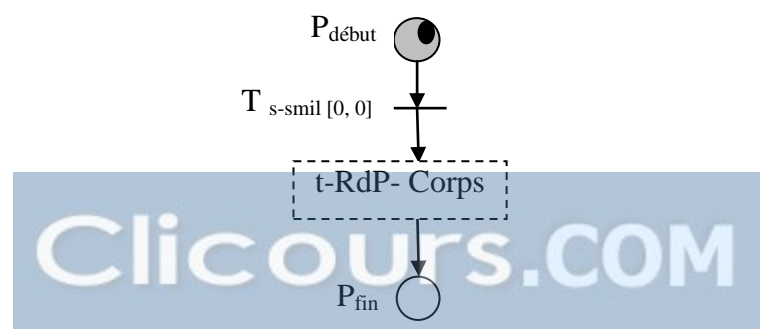
#### IV.2.3.1 Modélisation des éléments SMIL de base

Un document SMIL est modélisé par un t-RdP (**Figure IV.3**) qui contient :

- deux places : une place début et une place fin, et
- une transition de départ  $t_{s-smil}$  qui admet un intervalle de temps égal à  $[0, 0]$ .

La place début  $p$  est associée à la balise  $\langle smil \rangle$  et correspond au début de la présentation. Par contre, la place fin est associée à la balise  $\langle /smil \rangle$  et correspond à la fin de la présentation.

Dans le marquage initial  $M_0$ , la place début  $p$  est marquée par un jeton.



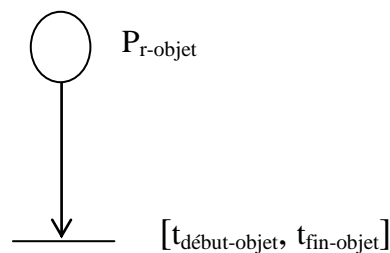
**Figure IV.3:** Modélisation d'un document SMIL par le t-RdP

Un document SMIL est la composition de plusieurs objets multimédias par l'intermédiaire d'opérateurs de synchronisation. On parle alors d'une modélisation de composants (**Figure IV.5**). Ces derniers encapsulent un t-RdP qui décrit le comportement temporel d'un objet multimédia. Ses interfaces d'entrée et de sortie sont respectivement un ensemble de places et une transition d'entrée et de sortie.

Notre modèle des t-RdP de la **Figure IV.4** est la représentation d'un objet média de base qui contient:

- une place  $P_{r\text{-objet}}$ .
- une transition avec l'intervalle  $[t_{\text{début-objet}}, t_{\text{fin-objet}}]$ , avec  $t_{\text{début-objet}}$  le temps de début d'exécution d'un objet et  $t_{\text{fin-objet}}$  le temps de fin d'exécution d'un objet.

la présence d'un jeton dans  $P_{r\text{-objet}}$  signifie que l'objet est en cours d'exécution entre  $t_{\text{début-objet}}$  et  $t_{\text{fin-objet}}$



**Figure IV.4:** Modélisation du média objet de base par le t-RdP

#### IV.2.3.2 Modélisation de l'opérateur de lecture en séquence <seq>

L'opérateur de séquence <seq> est composé de trois places : la place de début (interface d'entrée), la place de lecture, et la place de fin (voir **Figure IV.5**).

L'interface d'entrée  $T_{s\text{-seq}}$  est le lien du premier fils qui sera joué et l'interface de sortie est liée à la place  $P_{m\text{-seq}}$ .

L'interface de sortie  $T_{e\text{-seq}}$  contient l'intervalle  $SI_2 = [dur\text{-seq}, dur\text{-seq}]$ .

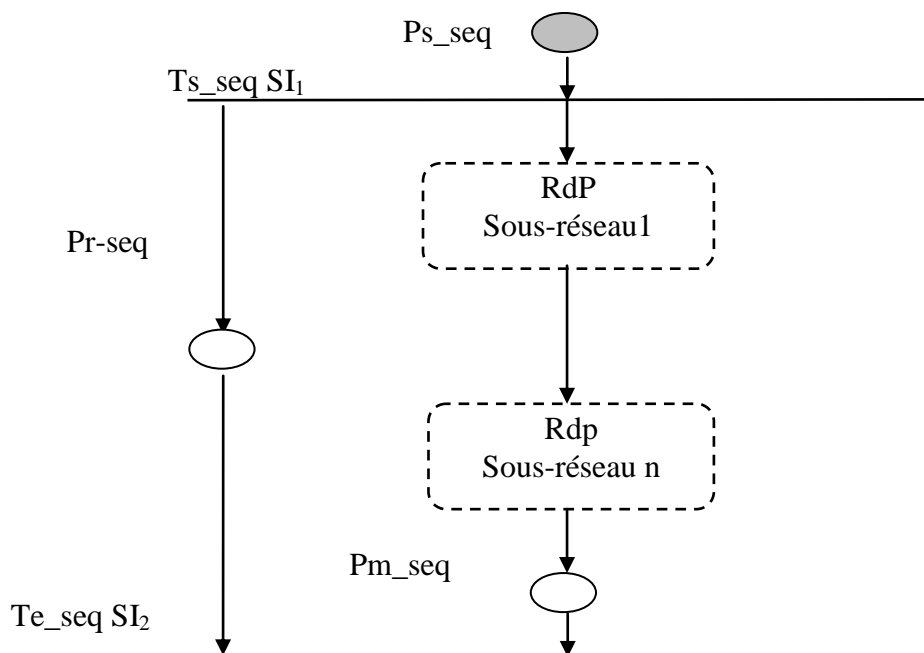


Figure IV.5: Modélisation de l'élément <seq> par le t-RdP

#### IV.2.3.3 Modélisation de l'opérateur de lecture en parallèle <par>

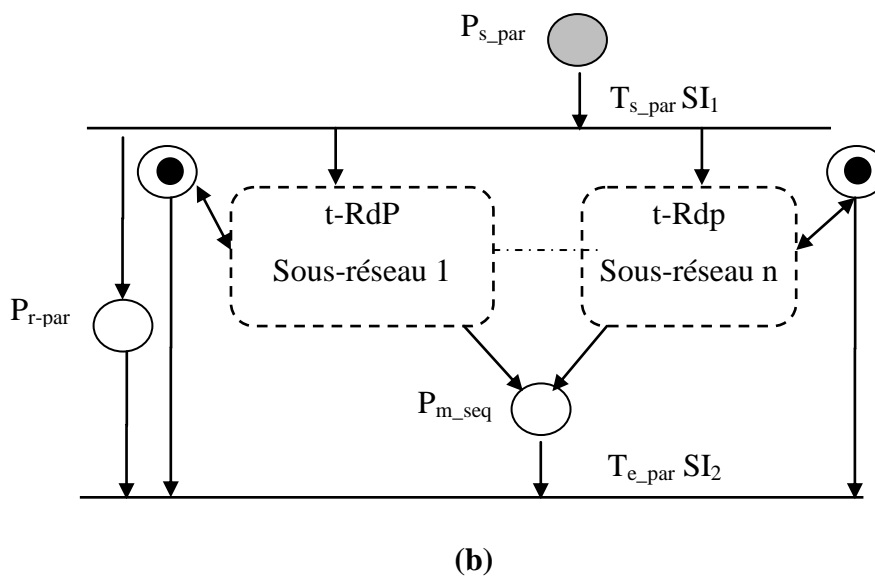
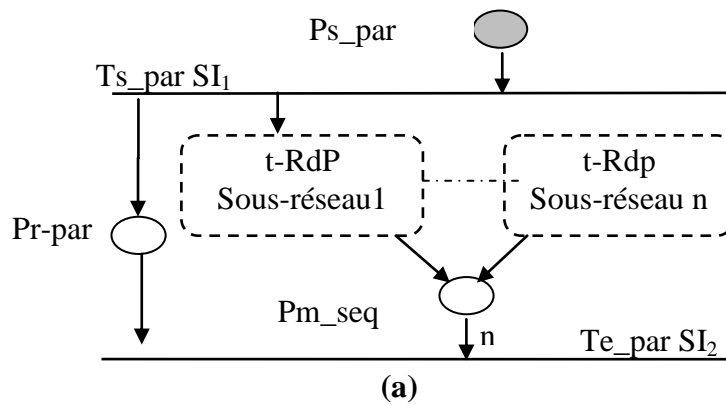
L'opérateur en parallèle <par> est composé des mêmes places et des transitions que celui de l'opérateur de séquence <seq>, toutefois l'interfaçage des médias est différent ; ces derniers s'attacheront tous à la même transition de départ ( $T_{\text{debut-par}}$ ) et se terminent de jouer dès le passage des jetons à la place fin ( $T_{\text{fin-par}}$ ). Cette dernière est liée par un arc dont la valuation de cet arc est égal au nombre de médias qui compose l'opérateur <par> (Figure IV.6).

La modélisation de l'élément **par** dépend de l'attribut « endsync » qui admet les trois sémantiques différentes suivantes :

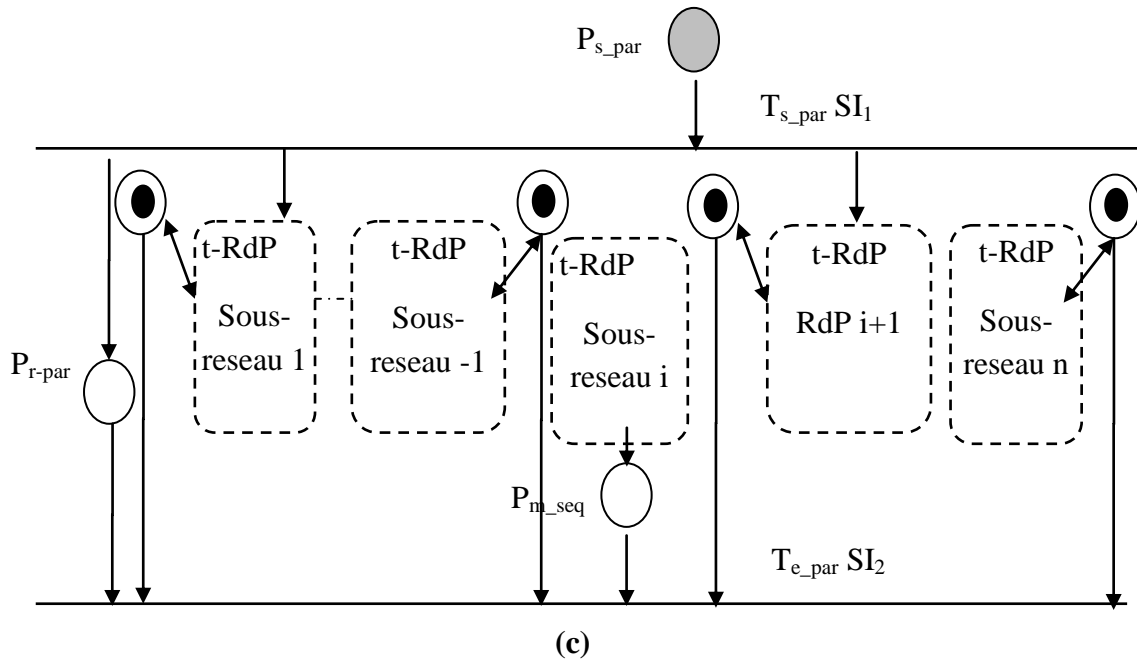
- Lorsque *endsync="last"* (comportement par défaut), l'élément <par> se termine lorsque tous ses éléments fils seront terminés. Par conséquent, le franchissement de la transition de fin de l'élément <par> doit se faire lorsque tous ses éléments fils seront terminés, c'est-à-dire dès le franchissement de la transition  $t_{\text{e-par}}$  qui doit avoir lieu après le franchissement de toutes les transitions des médias fils (Figure IV.6-a).

– Lorsque *endsync*="first", l'élément *<par>* se termine dès la fin d'un de ses fils. La transition de fin de l'élément *<par>* sera franchie dès la fin d'un de ses fils, où nous ajoutons une nouvelle place. C'est une place d'entrée/sortie qui est utilisée lorsque le premier fils se termine, la transition  $t_{e-par}$  sera activée et les autres médias s'arrêtent (**Figure IV.6-b**).

– Lorsque *endsync*="id-ref" où *id-ref* fait référence à un des fils de l'élément *<par>*. La fin de l'opérateur *<par>* est conditionnée par la fin de l'élément fils désigné par *id-ref*. Dans ce cas, le franchissement de la transition de fin du bloc *<par>* aura lieu dès la fin de l'élément "id-ref". Lorsque le fils désigné par *id-ref* se termine, tous les autres fils qui n'ont pas encore terminé doivent être arrêtés. Pour cela, nous ajoutons une place marquée pour chaque fils sauf pour l'enfant "id-ref". Dès que le fils spécifiée se termine, la transition  $t_{e-par}$  doit être tirée et le nombre des jetons dans toutes les places des autres fils sera augmenté (**Figure IV.6-c**).





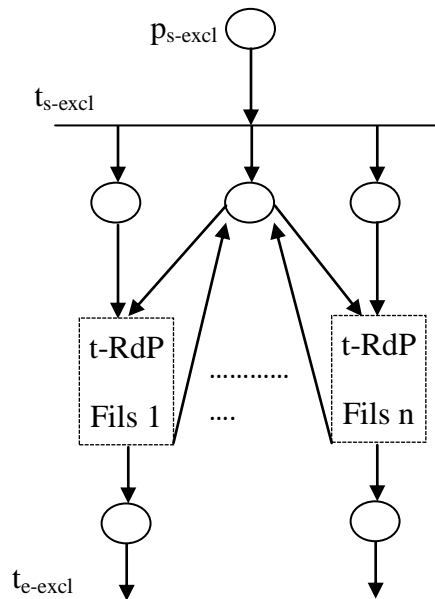


**Figure IV.6:** Modélisation de l'élément `<par>` à l'aide du t-RdP  
 (a) l'attribut (`endsync = 'last'`), (b) l'attribut (`endsync = 'first'`),  
 (c) l'attribut (`endsync = 'id-object'`).

#### IV.2.3.4 Modélisation de l'opérateur de lecture en exclusivité `<excl>`

Pour ce qui est de notre contribution relative aux éléments de SMIL, nous proposons les modélisations de l'opérateur `<excl>` et les objets des liens hypermédia `<a>` et `<area>`.

Pour l'opérateur `<excl>`, on définit une séquence non ordonnée des éléments dans lequel un seul objet média est joué à la fois (voir **Figure IV.7**).



**Figure IV.7 :** Modélisation de l'élément `<excl>` à l'aide du t-RdP

#### IV.2.3.5 Les objets des liens hypermédia

Un document SMIL peut définir des liens vers d'autres médias. Deux balises de lien hypermédia sont considérés dans notre approche : soient `<a>` et `<area>`. La balise `<a>` est le moyen le plus simple pour créer un lien hypermédia. La balise `<area>` comprend toutes les fonctionnalités de `<a>` avec la capacité de définir des liens multiples pour chaque clip.

La modélisation de L'élément `<a>` ou `<area>` est illustrée par:

- une place de lien (cercle en pointillé) qui associée une transitions avec un intervalle [ $t_{\text{début-objet-source}}$ ,  $t_{\text{fin-objet-source}}$ ].

- une seconde place qui associée a une transition contient un intervalle [0, fin] pour l'éléments `<a>` et un intervalle [début, fin] pour l'éléments `<area>`.

- un arc pointillé Véhicule l'information de contrôle renfermée dans les jetons de contrôle. (Jeton d'activation, Jeton de pause).

- *Les jetons d'état* : Les jetons sont de deux types:

- Les jetons d'activation: l'arrivée (ou la présence) d'un tel jeton dans une place quelconque active la place en question, et active l'objet associé à cette place.
- Les jetons de pause: la présence de ce jeton dans une place la met dans l'état suspendu, il suspend l'exécution de l'élément en question jusqu'au retour à l'état actif.

● Jeton d'activation

◆ Jeton de pause

Selon la valeur de l'attribut "show", les éléments d'une région peuvent avoir trois comportements :

- **Pause** : l'activation du lien suspend le composant source (met le jeton de pause) pendant l'exécution du composant cible, et permet de reprendre le composant source (met le jeton d'activation ) à partir de son point d'arrêt dès la fin de l'élément cible. Ce comportement est modélisé par la **Figure IV.8 a**.
- **Remplacer** : le composant cible remplace la source de composant; la source de composant est alors arrêtée. Ceci est modélisé par la **Figure IV.8 b** où l'activation de la place provoque l'arrêt de la présentation de l'élément source.

- **New** : le composant cible commence dans un nouveau contexte, sans influencer la source de composant. Ceci est modélisé par la **Figure IV.8 c** dans laquelle l'élément cible exécute de façon autonome et indépendante de la source de composant.

Dans tous les cas, le moment de l'exécution de l'élément cible n'est pas compté dans le temps de la présentation de l'élément source. Les éléments de liens n'ont aucune influence sur la synchronisation de l'élément source. Par exemple, si on prend le code SMIL de l'exemple 1 suivant :

Exemple 1 :

```
<a href="URL/video 2">
  <video src="video 1">
</a>
```

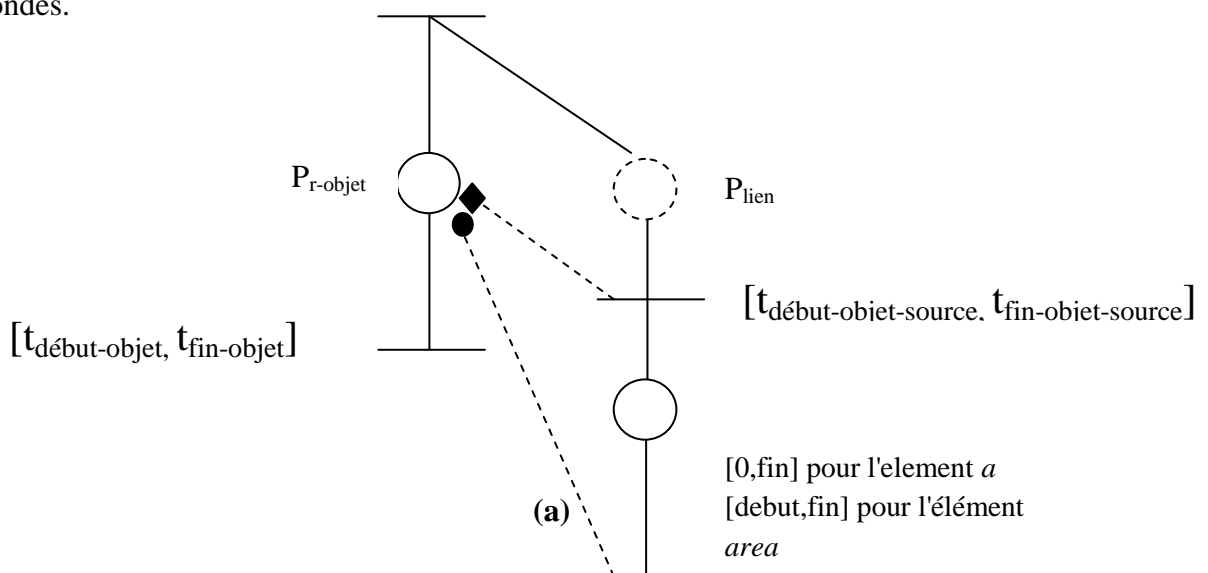
On voit que le clip source « video 1 » a des liens vers le clip cible « video 2 ». En effet, la « video 2 » prend la place de la « video1 » lorsque l'utilisateur clique sur « video1 ».

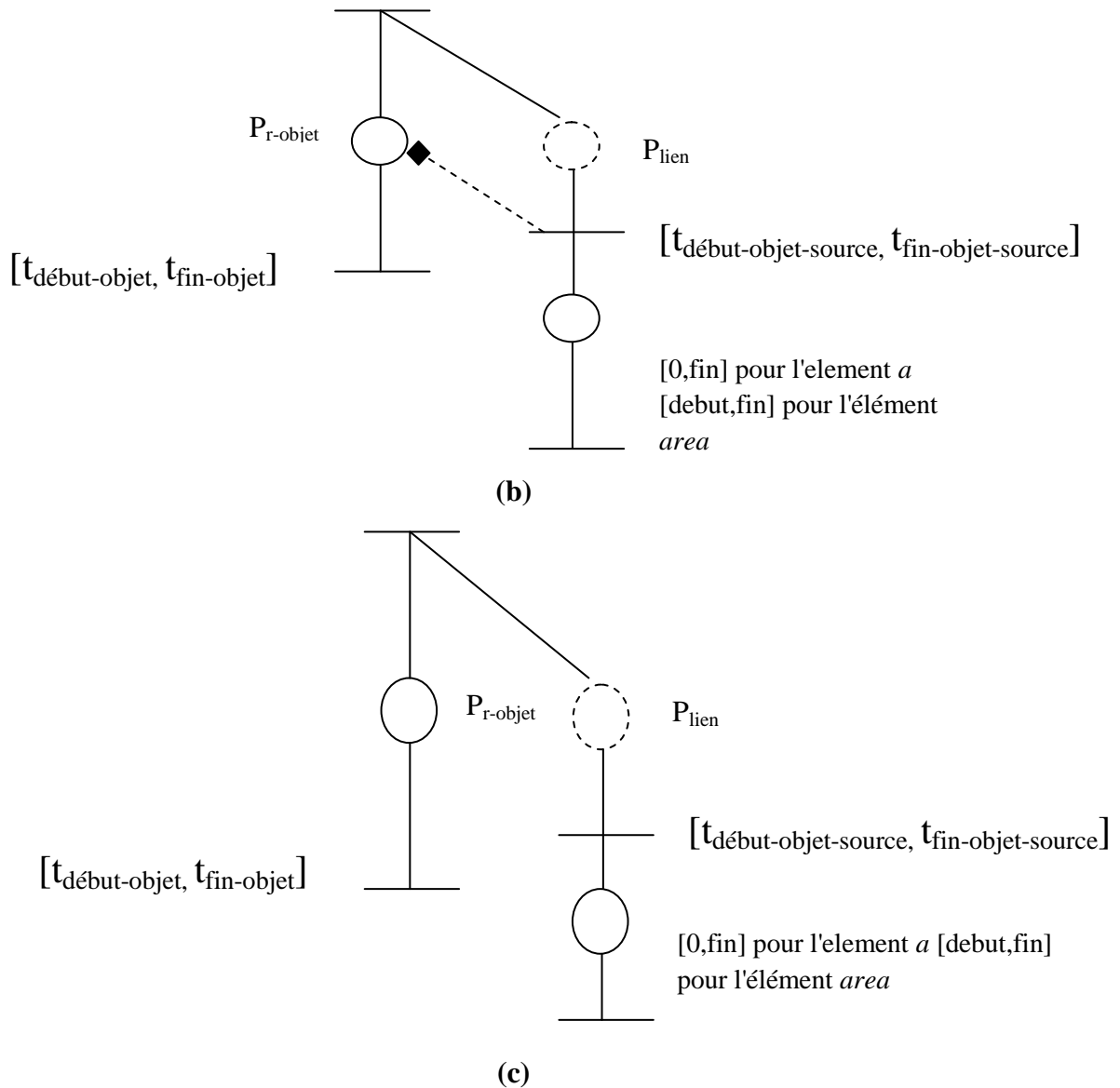
Pour créer un lien temporel, nous utilisons un marqueur <area> et on inclut des attributs temporels (début et de fin) qui précisent quand est ce que le lien est actif par rapport au début de la lecture du clip. Par exemple, si on prend le code SMIL de l'exemple 2 suivant :

Exemple 2 :

```
<video src="http://www.example.org/video1">
  <area href="http://www.example.org/AudioVideo" begin="0s" end="5s" ... />
  <area href="http://www.example.org/Style" begin="5s" end="10s" ... />
</video>
```

On peut créer deux liens temporels pour le clip «video1 ». Le premier lien est actif pendant les 5 premières secondes de lecture, tandis que le deuxième lien est actif pour les 5 prochaines secondes.





**Figure IV.8 :** Modélisation des éléments de liens hypermédia : (a): Show=Pause, (b): Show=Replace, (c): Show=New, à l'aide du t-RdP.

#### IV.2.3.6 Etude de cas de modélisation d'un document SMIL du cœur

A titre d'exemple, dans le cadre d'un service fourni par un hôpital, un médecin a la possibilité de choisir un service multimédia parmi plusieurs, par exemple la consultation des dossiers médicaux multimédias des malades. S'il s'agit d'un cardiaque, le médecin est capable, grâce à la synchronisation temporelle, de visualiser le dossier multimédia du malade (Figure IV.9) en se focalisant sur les organes affectés.

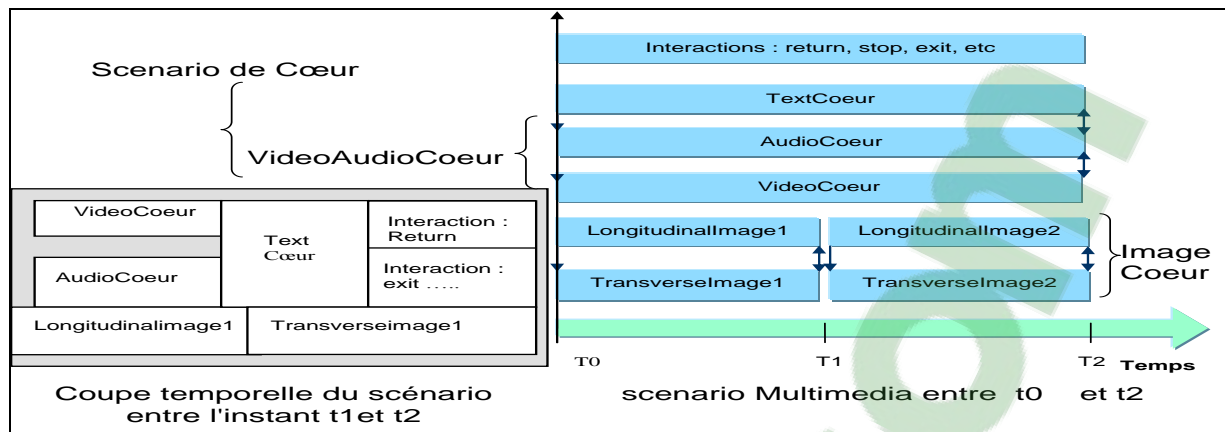


Figure IV.9: Exemple du DM interactif du cœur

Le dossier est composé d'une description textuelle du cœur (un texte décrivant le cœur) et son comportement, une séquence audio et vidéo décrivant les battements du cœur. La séquence audio et vidéo est synchronisée avec deux images, décrivant respectivement une coupe longitudinale et transversale du cœur. Ces deux coupes évoluent avec la séquence audio et vidéo du cœur. A tout moment, l'utilisateur peut arrêter, re-présenter, revenir au menu initial pour choisir des dossiers médicaux d'autres malades ou sortir complètement du scénario.

Dans notre travail, on a utilisé l'éditeur LimSee2 [7] pour générer le code SMIL 2.0 associé au DM interactif du cœur suivant :

```
<?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 )"/>
<layout type="text/smil-basic-layout">
<root-layout id="rootLayout" width="780" height="726" backgroundColor="white"/>
<region id="region1" backgroundColor="#FFFFFF" left="468" top="6" width="312"
```

```

height="318"/>
<region id="region2" backgroundColor="#FFFFFF" left="48" top="6" width="282"
  height="186"/>
<region id="region3" backgroundColor="#FFFFFF" left="42" top="198" width="420"
  height="324"/>
<region id="regiontext" backgroundColor="black" left="18" top="534" width="666"
  height="186"/>
<region id="region_ve" backgroundColor="#FFFFFF" left="486" top="330" width="276"
  height="186"/>
</layout>
</head>
<body id="body_mainSequence">
  <par dur="60s" endsync="last" id="par1">
    <seq dur="18s" id="seq1">
      
      
    </seq>
    <seq dur="18s" id="seq2">
      
      
    </seq>
    <textstream src="coeurtext.rt" begin="0s" dur="65s" id="text" region="regiontext" />
    <video begin="0s" dur="66s" end="67s" id="video" src="VideoCoeur.rv"
region="region3"/>
    <audio dur="69s" id="AudioCoeur" src="muzic.mp3"/>
    <a href="www.google.fr" dur="10s" id="lien" accesskey="e" show="new">
      <video dur="20s" id="video2" region="region_ve" src="VideoCoeur.rv"/>
    </a>
  </par>
</body>
</smil>

```

Dans notre exemple du DM interactif du cœur, chaque média est modélisé par un t-RdP (Figure IV.10). Nous avons associé une horloge globale pour chaque sous réseau modélisant les éléments <seq> ou <par> pour calculer le temps de tir des transitions du t-RdP.

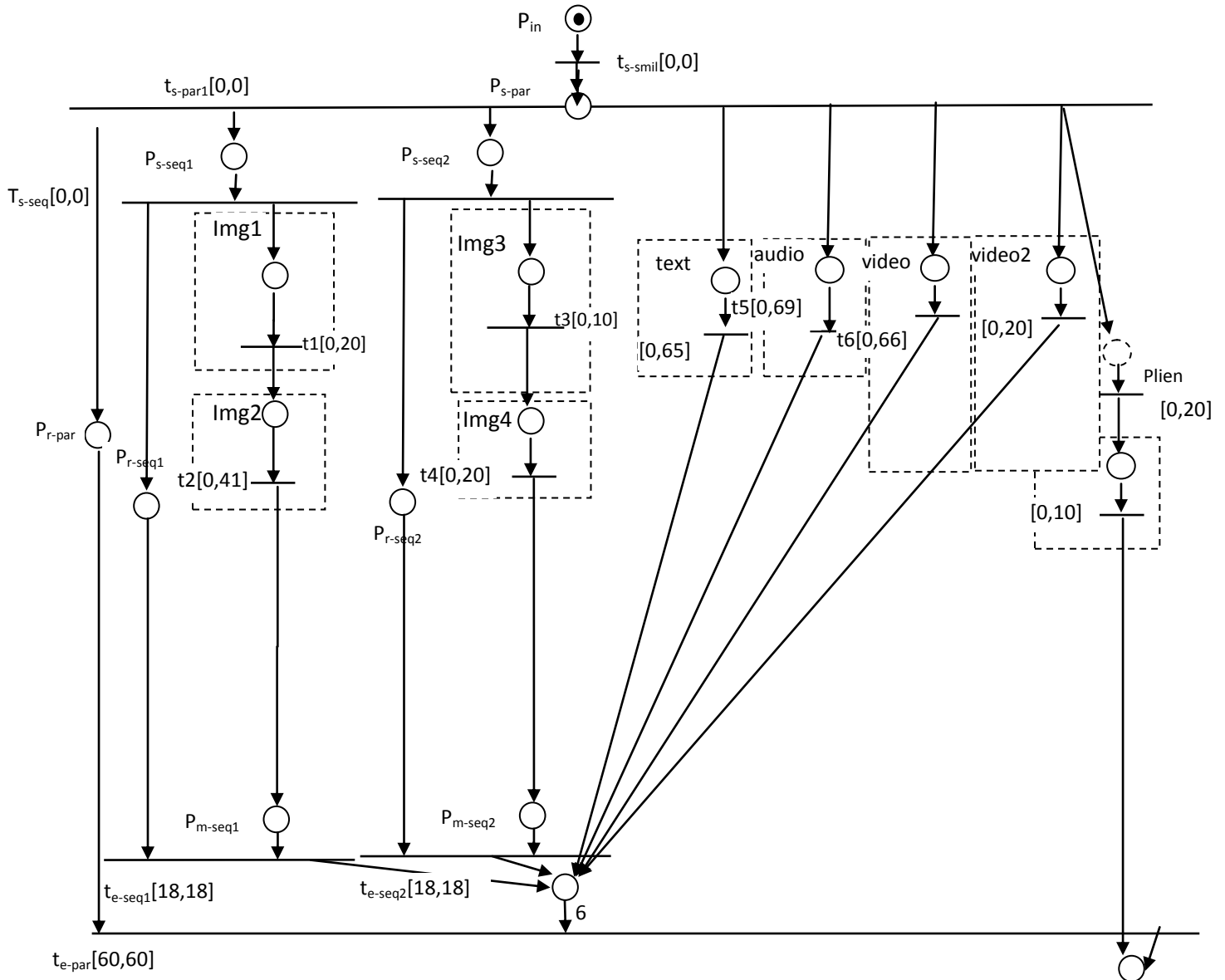


Figure IV.10 Exemple de modélisation du DM interactif du cœur à l'aide des t-RdP

### IV.3 Validation d'un document SMIL

La validation des documents SMIL peut se faire soit par simulation, soit par analyse des t-RdP. Dans notre travail, nous avons utilisé la simulation qui utilise l'algorithme du calcul de l'état suivant pour calculer le temps de tir des transitions du t-RdP.

Pour cela, on doit vérifier d'après [Yang 00] deux types d'incohérences dans un document SMIL modélisé par un t-RdP :

### 1- Incohérences intra-média

Chaque média est vérifiée à partir de ses attributs temporels begin, dur et end. Par exemple, si on prend le code SMIL du DM interactif du cœur de la **Figure IV.9**, on a le code SMIL du média vidéo suivant :

```
<video id= .... begin = "T_début" dur ="T_dur" end =" T_fin". .. / >
```

Pour vérifier la cohérence intra-média, on doit appliquer la règle de base suivante [Ayar 01] [Hosc 98] :

$$T\_début + T\_dur = T\_fin$$

Si la condition de cette règle n'est pas satisfaite alors une incohérence temporelle intra-média est détectée.

Par exemple, dans le cas du média vidéo du cœur dont le code SMIL est comme suit :

```
<video begin="0s" dur="66s" end="67s" id="video" src="VideoCoeur.rv"
region="region3"/>
```

On constate qu'il y a une erreur intra vidéo car :  $T\_début + T\_dur = 0s + 66s \neq T\_fin = 67s$

### 2- Incohérences inter-médias

La vérification de la cohérence inter-médias d'un document SMIL, consiste en l'analyse d'un ensemble de média respectant les contraintes temporelles de synchronisation. Elle est déduite par la comparaison entre la durée globale des opérateurs de synchronisation : <par>, <seq>, <excl>, et la durée de représentation des médias qui les composent.

#### Cas 1 de l'élément <Seq>

La durée de l'opérateur de synchronisation <seq> doit être supérieure ou égale à la somme des durées des médias qui les composent. Par exemple, si on prend le code SMIL du DM interactif du cœur (**Figure IV.9**), on a le code SMIL suivant :

```
<seq dur="18s" id="seq1">
  
  
</seq>
```

On constate qu'il y a une erreur inter-médias car la durée de l'élément <seq> qui égale à 18s est plus petite que la somme des durées des médias : image 1 (LongitudinalImage2.gif) et image 2 (LongitudinalImage1.gif) égale à 61s.



**Cas 2 de l'élément <Par>**

La durée de l'opérateur de synchronisation <Par> doit être supérieure ou égale à :

- ✓ la durée de l'élément média qui se termine en même temps que la dernière fin active de tous les éléments fils, dans le cas endsync="last"
- ✓ la durée du premier fils qui termine sa durée active, dans le cas endsync="first".
- ✓ la durée de l'élément média spécifié qui termine sa durée active, dans le cas endsync="id-ref".

Par exemple, si on prend le code SMIL du DM interactif du cœur (**Figure IV.9**), on constate qu'il y a une erreur inter-médias, car la durée de l'élément <par> qui égale à 60s est plus petite que la durée de l'élément média audio qui se termine en même temps que la dernière fin active de tous les éléments fils égale à 69s.

**Cas 3 de l'élément <excl>**

La durée de l'opérateur de synchronisation <excl> doit être supérieure ou égale à la somme des durées des médias qui les composent.

Par exemple, si on prend le code SMIL suivant :

```
<excl id="excl1" dur="10s">
  <priorityClass peers="defer">
    
    
    
  </priorityClass>
</excl>
```

On constate qu'il y a une erreur inter-médias car la durée de l'élément <excl> qui égale à 10s est plus petite que la somme des durées des médias images : imgexcl1, imgexcl2, imgexcl3 égale à 15s.

**3- Inconsistances**

L'exemple suivant montre un cas d'inconsistance lors de la lecture en parallèle des quatre médias : 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 la lecture du média TransverseImage2 est supérieur au temps de fin de la lecture du média AudioCoeur.

```

<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=" AudioCoeur" />
  
</par>

```

Dans l'exemple on remarque que le média TransverseImage2 n'est jamais joué.

#### IV.4 Expérimentation et évaluation

Pour mettre en œuvre notre travail conceptuel, nous avons conçu et réalisé un outil de modélisation et de validation du langage SMIL dont l'architecture générale est comme suit :

##### IV.4.1 Architecture générale de l'outil

L'architecture générale de l'outil réalisé dans ce mémoire est esquissée dans la **Figure IV.11** ci-dessous :

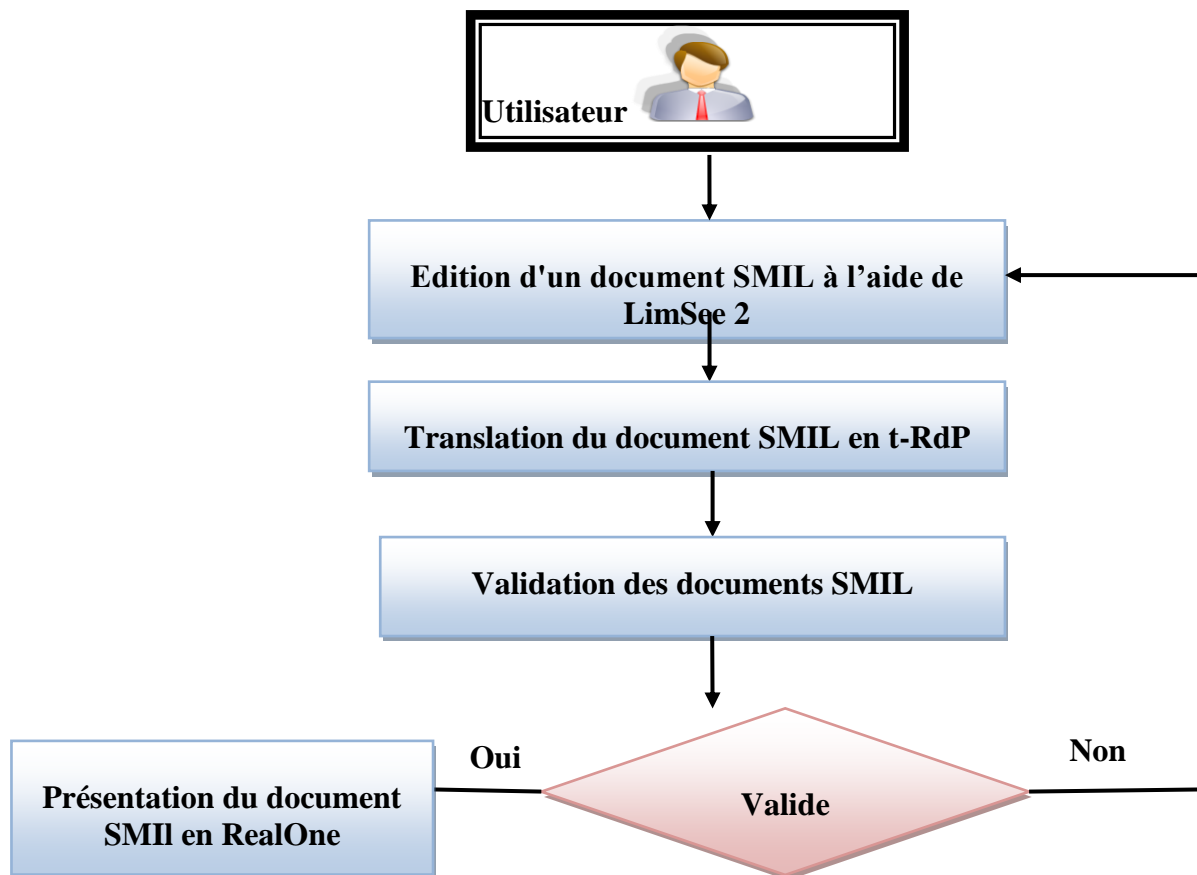


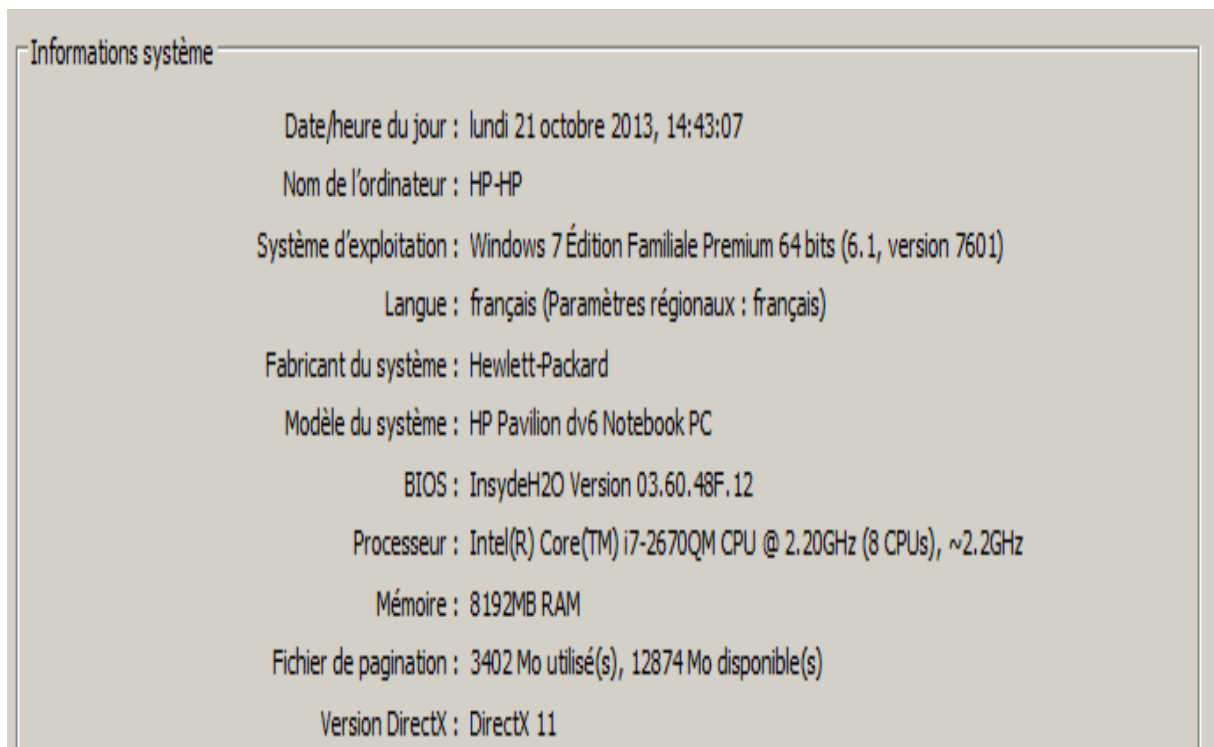
Figure IV.11 : Architecture générale de l'outil

Pour réaliser notre outil de modélisation et de validation de SMIL, nous avons utilisé l'environnement de travail suivant :

#### **IV.4.2 Environnement de travail**

##### **IV.4.2.1 Environnement matériel**

Pour l'implémentation de notre outil on a utilisé un Laptop de type HP dont la configuration est comme suit :



##### **IV.4.2.2 Environnement logiciel**

Pour réaliser notre outil de modélisation et de validation des documents SMIL, nous avons utilisé le langage de programmation Delphi. Ce dernier permet la création simple des éléments qui pourraient être intégrés dans l'IDE et bien d'autres. Le langage de programmation Delphi, appelé aussi Pascal Objet est originaire d'une extension du langage de programmation Pascal qui a été conçu pour une utilisation dans le domaine de la programmation orientée objet. Actuellement Embarcadero Delphi, anciennement connu sous CodeGear Delphi et Borland Delphi, est un IDE pour le développement d'applications Microsoft Windows.

Il a d'abord été développé par Borland comme un outil d'application pour Windows en 1995 à 2008 et plus tard il a été adopté par les Technologies Embarcadero de 2009 jusqu'à

aujourd'hui. Parmi les avantages de Delphi, on peut citer d'une part, son code qui est lisible en raison de la clarté de sa syntaxe, et ses premières versions avaient une très grande documentation, et d'autre part, sa prise en charge de l'API Windows entier.

#### **IV.4.3 Implémentation des modules de l'outil**

L'outil réalisé (voir architecture générale de la **Figure IV.11**) est mis en œuvre sur la plateforme HP Intel ® Core I7 en utilisant le langage de programmation Delphi, et fonctionne dans un environnement de fenêtres. Les modules de l'outil réalisé sont détaillés comme suit :

##### **Module 1 : Edition d'un document SMIL**

Pour éditer le code SMIL, on utilise soit les éditeurs de texte (WordPad ou Bloc-Note) ou bien l'éditeur LimSee 2.0 [ 7 ] selon les connaissances de l'auteur en SMIL (Voir le code SMIL de la **section IV.2.3.6**)

##### **Module 2 : Translation du document SMIL en t-RdP**

La translation automatique d'un document SMIL en t-RdP est réalisée en appliquant un ensemble de règles (voir les **sections IV.2.3.1, IV.2.3.2, IV.2.3.4, IV.2.3.5**) comme la modélisation des opérateurs <par> et <seq> et les attributs : begin, dur et end.

##### **a- Fragmentation et extraction des mots clés**

Supposant que le document SMIL (**section IV.2.3.6**) est syntaxiquement correct, on doit le parcourir pour extraire les mots clés qui seront stockés dans une liste (**Figure IV.12** ) ainsi que sa procédure d'implémentation (**Figure IV.13**). C'est à dire, on doit rechercher les mots clés prédéfinis tels que : <body>, <seq>, <par>, <img, <vid, begin=, dur=, </seq>, ...etc.

par2	0s	40s	90	last	par		
seq1	0s	20s	94		seq		
img_1	2s	10s	98			12s	
img_2	0s	15s	106			15s	
par_1	0s	10s	114	last	par		
img_3	2s	10s	119			10s	

Figure IV.12 : Extraction des mots clés

```

Unit1  Unit2
-----
procedure ouvrir();
begin
  lm:=Form1.RichEdit1.Lines.Count;
  for j:=0 to lm do
  begin
    mot:=Form1.RichEdit1.Lines[j];
    if mot<>' ' then
    begin
      i:=1;
      while mot[i]=' ' do
      begin
        i:=i+1;
      end;
      if mot[i]='<' then
      begin
        Form1.ListBox1.Items.Add('<');
        i:=i+1;
      end;
      tb:='';
      while i<Length(mot) do
      begin
        tb:=tb+mot[i];
      end;
    end;
  end;
end;

```

Figure IV.13: Procédure d'extraction des mots clés

Dans notre application, on a utilisé la structure de données des piles pour la construction du t-RdP définie comme suit :

```
Type pile=^Element;
      Element=record
          typ, nom, avant, apres, tm: string;
          x, y, niveau, poid: integer;
          suivant: pile;
      end;
```

Où les champs sont définis comme suit:

- typ: contient une place ou une transition
- nom: le nom d'une place ou transition
- x, y, niveau : désigne la position de la place ou la transition dans le graphe
- poid: désigne le poids de l'arc
- avant et apres: désignent les places qui sont avant ou apres une transition( par suite les transitions qui sont avant ou apres une place).
- tm: signifier soit le temps (begin ou dur pour les transition et le nombre de jeton pour les places)

#### **b- Translation automatique de SMIL en t-RdP**

Après l'extraction des mots clés, on construit le t-RdP correspondant au code SMIL en appliquant un ensemble de règles déjà décrites les **sections IV.2.3.1, IV.2.3.2, IV.2.3.4, IV.2.3.5.**

Enfin, on décompose le t-RdP global en sous réseaux en utilisant l'algorithme du calcul de l'état suivant pour calculer le temps de tir des transitions du t-RdP de chaque sous réseau.

Pour cela, nous avons les implémentations des différents opérateurs de synchronisation : <seq>, <par> et <excl> et des différents médias : texte, image, audio et vidéo suivantes :

Pour l'implémentation de l'opérateur <par>, on a le programme Delphi suivant (**Figure IV.14**) :

```

Unit1 | Unit2 |
if form1.ListBox1.Items.Strings[i]='par' then
begin
if trouvl=false then
begin
ptin:=ptin+1;
typ:='p';nom:='Pin'+inttostr(ptin);avant:='';apres:='Tin'+inttostr(ptin);
elem^.apres:=nom;
tm:='0';x:=plig; y:=pcol;
poid:=1;
empile(elem,T);
typ:='t';nom:='Tin'+inttostr(ptin);avant:='Pin'+inttostr(ptin);apres:='';
tm:='0';x:=plig; y:=y+pcol;
poid:=1;
empile(elem,T);
end;
trouvl:=false;
nom:=elem^.nom;
typ:='p';
avant:=elem^.avant;apres:=elem^.apres;
tm:=inttostr(k1);x:=elem^.x; y:=elem^.y;
poid:=100000;
empile(elem,T);
/////////////////

```

Figure IV.14: Implémentation de l'opérateur &lt;par&gt;

Pour l'implémentation de l'opérateur <seq>, on a le programme Delphi suivant (Figure IV.15) :

```

1 | Unit2 |
if form1.ListBox1.Items.Strings[i]='seq' then
begin
if trouvl=false then
begin
ptin:=ptin+1;
typ:='p';nom:='Pin'+inttostr(ptin);avant:='';apres:='Tin'+inttostr(ptin);
elem^.apres:=nom;
tm:='0';x:=plig; y:=pcol;
poid:=1;
empile(elem,T);
typ:='t';nom:='Tin'+inttostr(ptin);avant:='Pin'+inttostr(ptin);apres:='';
tm:='0';x:=plig; y:=y+pcol;
poid:=1;
empile(elem,T);
end;
trouvl:=false;
nom:=elem^.nom;
typ:='p';
avant:=elem^.avant;apres:=elem^.apres;
tm:=inttostr(k1);x:=elem^.x; y:=elem^.y;
poid:=100000;

```

Figure IV.15: Implémentation de l'opérateur &lt;seq&gt;

Pour l'implémentation des médias, on a le programme Delphi suivant (**Figure IV.16**) :

```
Unit2 | smil |
end;
// ref, animation, audio, img, text, et video
if ((form1.ListBox1.Items.Strings[i]='img')
or (form1.ListBox1.Items.Strings[i]='audio')
or (form1.ListBox1.Items.Strings[i]='ref')
or (form1.ListBox1.Items.Strings[i]='text')
or (form1.ListBox1.Items.Strings[i]='animation')
or (form1.ListBox1.Items.Strings[i]='video')
or (form1.ListBox1.Items.Strings[i]='textstream')) then
begin
/////////
if trouvu1=false then
begin
ptin:=ptin+1;
typ:='p';nom:='Pin'+inttostr(ptin);avant:='';apres:='Tin'+inttostr(ptin);
elem^.apres:=nom;
tm:='0';x:=plig; y:=pcol;
poid:=1;
empile(elem,T);
//elem^.poid:=0;
typ:='t';nom:='Tin'+inttostr(ptin);avant:='Pin'+inttostr(ptin);apres:='';
tm:='0';x:=plig; y:=y+pcol;
poid:=1;
```

**Figure IV.16** : Implémentation des médias

### **Module 3 : Vérification de cohérence temporelle**

Dans ce module, on doit vérifier les incohérences temporelles intra et inter-médias suivantes :

#### **a- Vérification de cohérence temporelle intra-média**

Cette vérification se fait en comparant les attributs begin, dur et end. Une situation de conflit temporel sera détectée dès que  $B + D \neq E$  (avec B le temps de début, D le temps de durée et E le temps de fin d'un média). (voir procédure de la **Figure IV.17**).





```

Unit1 Unit2
procedure correction();
var mot1:string;
    k1:integer;
begin
    calcul();

    while pos('<',Form1.RichEdit1.Lines[lm])=0 do lm:=lm-1;
    //tr:=false;
    repeat
    begin
    lm:=lm+1;
    mm1:=0;
    if pos('>',Form1.RichEdit1.Lines[lm])>0 then mm1:=1;
    if pos('/>',Form1.RichEdit1.Lines[lm])>0 then mm1:=2;
    tr:=false;
    if pos('begin',Form1.RichEdit1.Lines[lm])>0 then
    begin
    i:=pos('begin',Form1.RichEdit1.Lines[lm])-1;
    k1:=1;
    typ:='';
    for k1:=1 to i do
    typ:=typ+' ';
    typ:=typ+'begin="'+form1.Edit5.Text+'s" ';
    Form1.RichEdit1.Lines[lm]:=typ;
    end;

Unit1 Unit2
if deb+dur<>fin then
begin
dit:=true;
form1.label4.caption:='incoherence intraelement';
i:=1;
while i<=Form1.RichEdit1.Lines.Count do
begin
p:='';
p:=p+'id="'+ form1.ensGrid.Cells[0,ens];
if pos(p,Form1.RichEdit1.Lines[i])>0 then
begin
form1.Edit1.Text:=Form1.RichEdit1.Lines[i];
form1.RichEdit1.Lines.Insert(i,'eeeeee');
form1.RichEdit1.SelLength:=length(Form1.RichEdit1.Lines[i+1]);
form1.RichEdit1.SelAttributes.Color := clred;
form1.RichEdit1.Lines.Delete(i);
lm:=i;
j:=5;
form1.Edit5.Text:=inttostr(deb);
form1.Edit6.Text:=inttostr(dur);
break;
end;
i:=i+1;
end;

```

Figure IV.17 : Procédure de détection des incohérences temporelles intra-média

### b- Vérification de cohérence temporelle inter-médias

Cette vérification se fait par la comparaisant du temps de tir des transitions du t-RdP (voir procédure de la Figure IV.18). Pour cela, on les a cas suivants :

**Cas de l'élément <seq>**: le temps de tir de la transition  $T_{e-seq1}$  (Figure IV.10) diminué du temps de tir de la transition  $t2$  doit être supérieur ou égal au temps de tir de la transition  $t2$ .

Par exemple: pour le temps de tir de la transition  $T_{e-seq}$  égale 79s et le temps de tir de la transition  $t7$  égale 61on remarque que :  $79\text{ s} - 61\text{ s} < 61\text{ s}$

**Dans le cas de l'élément <par>**: le temps de tir de la transition  $T_{e-par}$  diminué du temps de tir:

- de la transition de l'élément média qui se termine en même temps que la dernière fin active de tous les éléments fils, dans le cas endsync="last"

- de la transition du premier fils qui termine sa durée active, dans le cas endsync="first".

- de la transition de l'élément média spécifié qui termine sa durée active, dans le cas endsync="id-ref".

Doit être supérieur ou égal au temps de tir :

- de la transition de l'élément média qui se termine en même temps que la dernière fin active de tous les éléments fils, dans le cas endsync="last"

- de la transition du premier fils qui termine sa durée active, dans le cas endsync="first".

- de la transition de l'élément média spécifié qui termine sa durée active, dans le cas endsync="id-ref".

Dans la **Figure IV.10** on a plusieurs incohérences inter-médias par exemple :

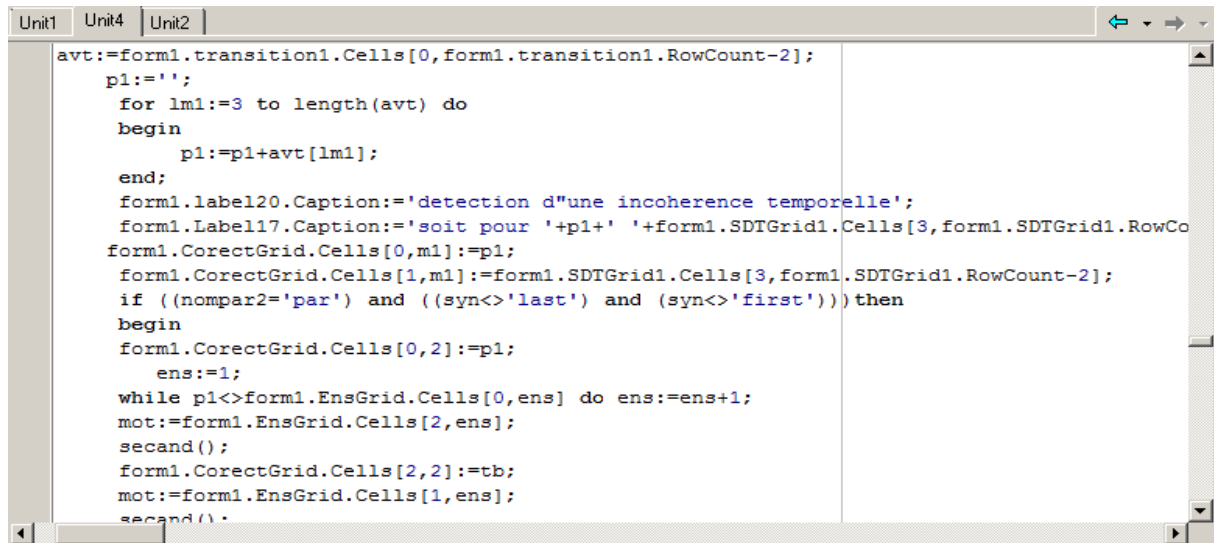
Le temps de tir de la transition  $t_5$  du média audio (égal à 69s) est plus grand que le temps de tir de la transition  $t_{e-par}$  (égal à 129s) diminué du temps de tir de la transition  $t_5$  (égal à 69s).

```

Unit1 | Unit2 |
procedure TForm1.Button29Click(Sender: TObject);
var mm:boolean;
    il,som:integer;
begin
    //////////// correction des incohorences
    trouv:=true;
    vericalc();
    if trouv=true then
    begin
        form1.RichEdit1.Lines.Insert(mm1,'e');
        form1.RichEdit1.SelLength:=length(form1.RichEdit1.Lines[mm1+1]);
        form1.RichEdit1.SelAttributes.Color := clBlack;
        form1.RichEdit1.Lines.Delete(mm1);

    correctioninter();
    i:=positf;
    j:=positf+2;
    while j<(form1.positGrid.RowCount-1) do
    begin
        form1.positGrid.Cells[1,j]:=inttostr(strtoint(form1.positGrid.Cells[1,j])-npil+2);
        j:=j+1;
    end;
    while positf<=(form1.positGrid.RowCount-3) do
    begin

```



```

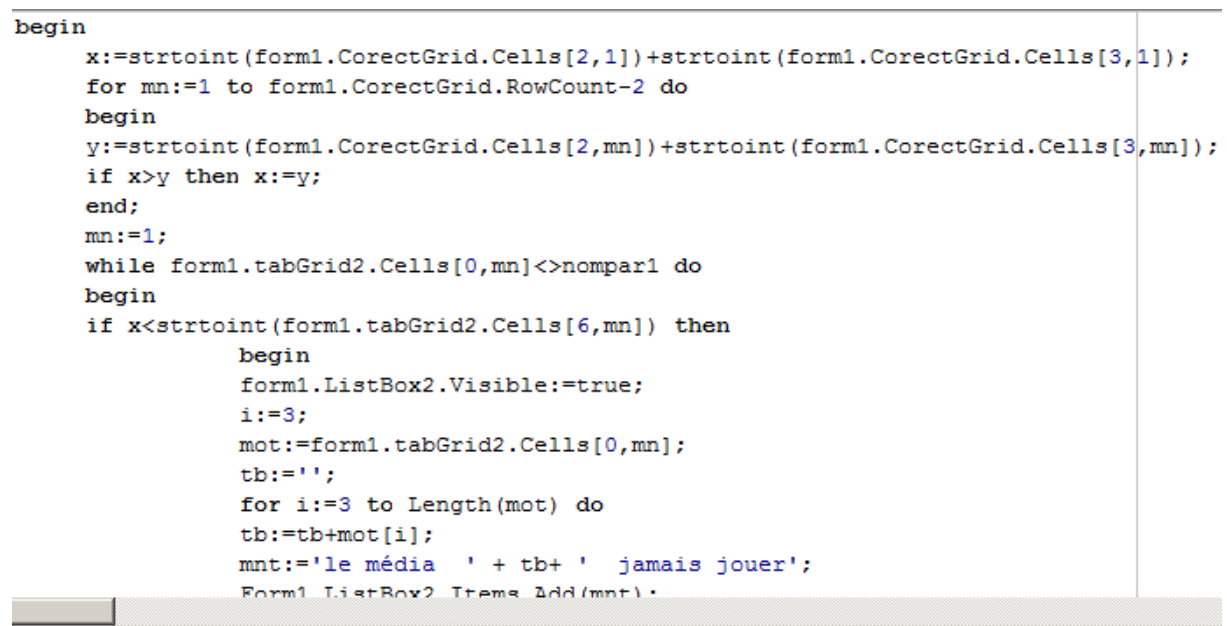
Unit1  Unit4  Unit2
avt:=form1.transition1.Cells[0,form1.transition1.RowCount-2];
p1:='';
for lm1:=3 to length(avt) do
begin
  p1:=p1+avt[lm1];
end;
form1.label20.Caption:='detection d"une incoherence temporelle';
form1.Label17.Caption:='soit pour '+p1+' '+form1.SDTGrid1.Cells[3,form1.SDTGrid1.RowCo
form1.CorectGrid.Cells[0,m1]:=p1;
form1.CorectGrid.Cells[1,m1]:=form1.SDTGrid1.Cells[3,form1.SDTGrid1.RowCount-2];
if ((nompar2='par') and ((syn<>'last') and (syn<>'first')) then
begin
  form1.CorectGrid.Cells[0,2]:=p1;
  ens:=1;
  while p1<>form1.EnsGrid.Cells[0,ens] do ens:=ens+1;
  mot:=form1.EnsGrid.Cells[2,ens];
  second();
  form1.CorectGrid.Cells[2,2]:=tb;
  mot:=form1.EnsGrid.Cells[1,ens];
  second();

```

Figure IV.18 : Procédure de détection des incohérences temporelles inter-médias

**c- Inconsistance:**

Si une transition du t-RdP modélisant un document SMIL n'est jamais franchie alors le média lié a cette transition est jamais joué.



```

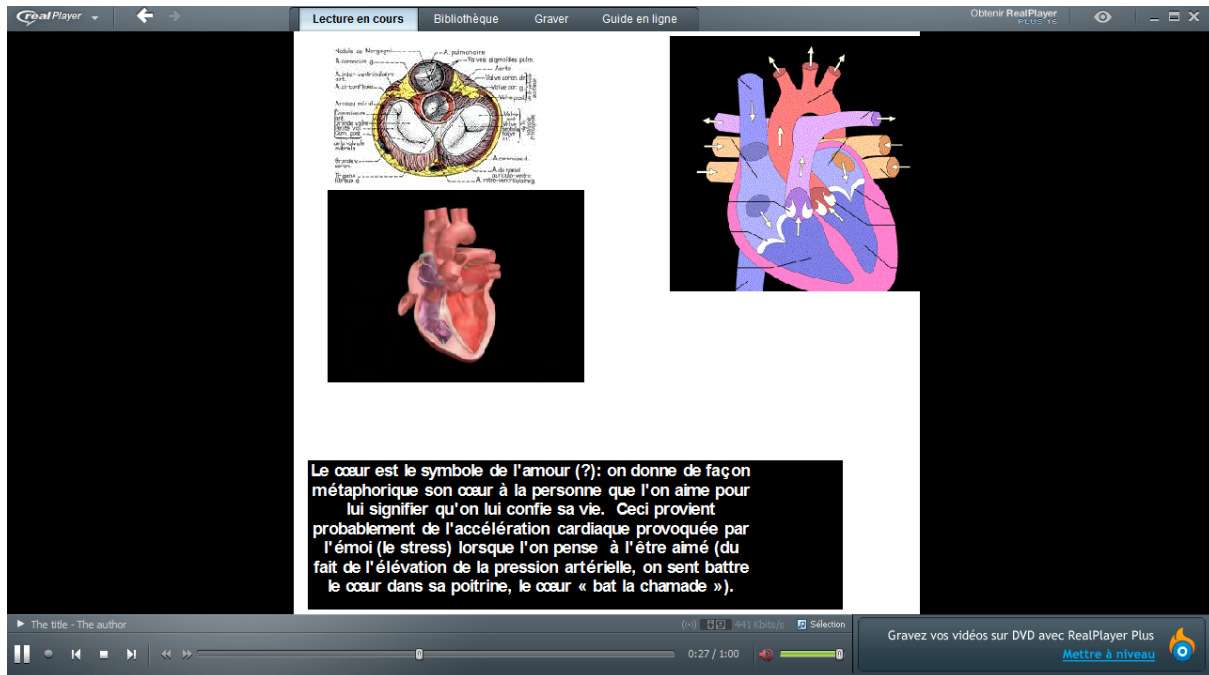
begin
  x:=strtoint(form1.CorectGrid.Cells[2,1])+strtoint(form1.CorectGrid.Cells[3,1]);
  for mn:=1 to form1.CorectGrid.RowCount-2 do
  begin
    y:=strtoint(form1.CorectGrid.Cells[2,mn])+strtoint(form1.CorectGrid.Cells[3,mn]);
    if x>y then x:=y;
  end;
  mn:=1;
  while form1.tabGrid2.Cells[0,mn]<>nompar1 do
  begin
    if x<strtoint(form1.tabGrid2.Cells[6,mn]) then
    begin
      form1.ListBox2.Visible:=true;
      i:=3;
      mot:=form1.tabGrid2.Cells[0,mn];
      tb:='';
      for i:=3 to Length(mot) do
      tb:=tb+mot[i];
      mnt:='le média ' + tb+ ' jamais jouer';
      Form1.ListBox2.Items.Add(mnt);
    end;
  end;

```

Figure IV.19 : Procédure de détection des inconsistances

**Module 4 : Présentation du document SMIL**

Dans ce module, on doit présenter le document SMIL valide en jouant les médias qui le composent à l'aide d'un lecteur multimédia (player) comme : RealOne, QuickTime, VLC, etc.



**Figure IV.20:** Exemple d'une présentation du document SMIL du Cœur à l'aide du player RealOne

#### IV.5 Conclusion

Dans ce chapitre, nous avons détaillé notre outil de modélisation et de validation des présentations des documents SMIL, qui est basée sur le modèle des RdP temporels (t-RdP).

Les t-RdP est un outil graphique et mathématique qui permet de formaliser la sémantique du langage SMIL, et de le valider ainsi par simulation ou par analyse des propriétés structurelles et comportementales. Pour expérimenter notre travail conceptuel, nous avons conçu et réalisé un outil de modélisation et de validation des documents SMIL qui permet :

- de traduire un code SMIL 2.0 en réseaux de Petri t-temporels (t-RdP)
- de vérifier les incohérences temporelles intra et inter-médias et un cas d'inconsistance.
- de présenter un document SMIL 2.0 valide, en utilisant un player tel que : RealOne.

### Conclusion & Perspectives

Grâce aux avancements technologiques et notamment informatiques (puissance de calcul des ordinateurs, capacité de stockage, nouveaux supports de communication, etc.), les possibilités d'interactions entre l'homme et la machine ne cessent de s'accroître. Les présentations multimédia en sont l'illustration parfaite

De nos jours, la notion de présentation multimédia est devenue de plus en plus répandue et utilisée dans les différents domaines, par exemple l'enseignement, la publicité, les présentations commerciales, etc. Un système de présentation multimédia est un système informatique intégré caractérisé par sa capacité à traiter des informations exprimées dans plusieurs média, comme le son, la vidéo, l'image, le graphique, le texte, etc.

Cette avancée technologique au niveau de la puissance des ordinateurs, des périphériques et des réseaux a abouti à la création d'un grand nombre d'applications multimédia, comme le télé-enseignement, la visioconférence, la vidéo à la demande, la téléphonie IP etc.

De plus, la disponibilité de périphériques nécessaires pour le traitement des données multimédia (comme les cartes vidéo ou audio haute qualité et les lecteurs CD), facilite la mise en œuvre d'applications multimédia. Enfin, l'évolution rapide des réseaux à haut débit rend possible l'échange de l'information multimédia entre des sites interconnectés, notamment sur le web. En effet, le réseau Internet, qui fournit à travers le Web un accès à l'information hypertexte, offre maintenant la possibilité de synchroniser les objets média dans les documents hypertextes. De son côté, le système de communication télévisuelle, qui fournissait jusqu'à présent des informations audiovisuelles passives, commence à offrir des moyens d'interaction et d'enrichissement de l'information diffusée.

Parmi les standards qui illustrent ces nouveaux services, on note les langages SMIL, HTML+Time, SVG, etc. Cette évolution fait exploser les besoins en documents multimédia (DM), et par conséquent, pousse le développement d'outils de création, de production et de diffusion de ces types de documents.

SMIL est un langage de balisage, permettant de synchroniser des fichiers de nature différente (*images, sons, textes, vidéos, animations, flux de texte*) pour en faire des objets multimédia aptes à être intégrés dans une page Web. Ce langage s'appuie avant tout sur le XML (*eXtensible Markup Language*), il est même entièrement basé sur celui-ci.

SMIL est un langage d'intégration de média (au sens « d'élément de contenu ») pour réaliser des présentations audiovisuelles interactives accessibles sur le web. Le terme «intégration» concerne non seulement le placement spatial des objets dans l'écran mais surtout leur agencement temporel défini sous forme d'un scénario qui exprime leurs synchronisations et enchaînements. A ces deux fonctions de base, placement spatial et placement temporel, est associé un ensemble d'éléments du langage qui couvre toute une palette de besoins complémentaires tels que spécification de transitions, de liens hypermédia, d'animations et de contrôle fin du temps.

Les DM apparaissent comme une nouvelle génération d'applications informatiques dont la conception à grande échelle met en jeu leurs cohérences. 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 le valider en vérifiant sa cohérence. Ce qui fait l'objet du travail de ce mémoire.

Dans ce mémoire, nous avons réalisé un outil de modélisation et de validation des documents SMIL à l'aide des RdP temporels de Merlin (t-RdP). En effet, le modèle des t-RdP permet de valider un document SMIL soit par simulation, soit par analyse des propriétés structurelles ou comportementales. Dans ce mémoire, nous avons étudié les éléments de SMIL comme : les médias et les opérateurs <seq>, <par> et <exc>.

Notre travail est loin d'être terminé, nous pensons réaliser à l'avenir les perspectives suivantes :

- Vérification de la cohérence spatiale d'un document SMIL
- Etendre le pouvoir d'expression du modèle t-RdP afin de prendre en charge l'ensemble des éléments de SMIL ( l'opérateur d'exclusivité <excl>, attribut repeat, élément switch, ... )

**Bibliographie**

- [Adib 00] 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.
- [AYA 01] Ayars, J. Recommandation du W3C, "Synchronized Multimedia Integration Language (SMIL 2.0)", en ligne à "<http://www.w3.org/TR/2001/REC-smil20-20010807/>", 07 août 2001.
- [Babe 93] K. Böhm, K. Aberer. An Object-Oriented Database Application for HyTime Document Storage. Technical report, GMD-IPSI, 1993
- [Bart 99] Barták R., Constraint Programming - What is behind?. in J. Figwer (editor) Proceedings of the Workshop on Constraint Programming in Decision and Control, Juin 1999, Pologne.
- [Bayk 97] Baykan A., Fox M., Spatial layout by disjunctive constraint satisfaction. AIEDAM 11, 245–262, January 1997.
- [Benn 96] B. Bennett. The application of qualitative spatial reasoning to GIS. In R.J. Abraham, editor, Proc First Int. Conf. on GeoComputation, volume I, pages 44–47, Leeds, 1996.
- [Bert 00] Bertino E., Member S., Ferrari E; and Stolf M., MPGS: An Interactive Tool for the Specification and Generation of Multimedia Presentations, IEEE Transactions On Knowledge And Data Engineering, Vol.12, No.1, January/February 2000.
- [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.
- [Bloc 00] I. Bloch, 2000. Spatial Representation of Spatial Relationships Knowledge. 7th International Conference on Principles of Knowledge Representation and Reasoning KR 2000, Breckenridge, CO, pp. 247-258.
- [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.
- [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.
- [Burn 94] L. Burnard. TEI EDW25 What is SGML and does it help? SIGIR'94, Tutorial Material: Text Encoding and Information Interchange, 1994.
- [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.
- [Chris 92] Christian Freksa. Using orientation information for qualitative spatial reasoning. In A. U. Frank, I. Campari, and U. Formentini, editors, Theories and methods of spatio-temporal reasoning in geographic space, pages 162 – 178. Springer; Berlin, 1992.
- [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.
- [Davi 92] David A. Randell, Zhan Cui, and Anthony Cohn. A spatial logic based on regions and connection. In Bernhard Nebel, Charles Rich, and William Swartout, editors, KR'92. Principles of Knowledge Representation and Reasoning : Proceedings of the Third International Conference, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.
- [Deem 00] K.V. Deemter, R. Power. Authoring Multimedia Documents using WYSIWYM Editing. Association for Computational Linguistics, 2000



- [Euze 94] Jérôme Euzenat. Granularité dans les représentations spatio-temporelles. April 1994. RR - 2242, INRIA.
- [Gaff 02] F. Gaffiot. Le Grand Gaffio, Dictionnaire Français-Latin. Hachette, 2000.
- [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.
- [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.
- [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>
- [Huiq 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.
- [Jedi 05] A. JEDIDI. "Modélisation générique de documents multimédia par des Métadonnées : Mécanismes D’annotation et D’interrogation". Thèse de Doctorat, Université TOULOUSE III - PAUL SABATIER U.F.R Mathématique Informatique Gestion, 06 Juillet 2005.
- [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.
- [Khan 97] W. Khansa, « Réseaux de Petri P-temporels: contribution à l'étude des Systèmes à Evénements Discrets ». Thèse de Doctorat de l'Université de Savoie, France, mars 1997.

- [Kirm 08] K Irmgarde. "Donner accès à des documents multimédia atypiques : projets web, animations 3D, installations interactives", Institut International du Multimédia (IIM) du pôle universitaire Léonard de Vinci, France, 2008
- [Labe 07] I. LABED. " Méthodes et Outils pour la Construction de Scènes Multimédia Distribuées". Thèse de Doctorat, Université Mentouri – Constantine- Algérie, 08 Décembre 2007.
- [Lamp 86] LAMPORT L., LATEX: a document preparation system, Addison-Wesley, New York, 1986.
- [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'Édition et de Présentation de Documents Structurés Multimédia. Thèse de Doctorat, Université Joseph Fourier – Grenoble 1. 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.
- [Mare 09] Maredj Azze-Eddine. Spécification et vérification des relations temporelles et spatiales dans un document multimédia interactif. Thèse de doctorat, Université des sciences et de la technologie Houari Boumedienne, 2009.
- [Mare 10] Maredj Azze-Eddine, Tonkin Nourredine, Sadallah Madjid. «MediaStudio: Un système d'édition et de présentation de documents multimédia interactifs.» revue de l'information Scientifique et technique (RIST) , Vol. 18, No 2, pages 25-47, 2010
- [Mari 01] Marriott K., Moulder P., Stuckey P. and Borning A. Solving disjunctive constraints for interactive graphical applications. International Conference on Principles and Practice of Constraint Programming, CP01. Paphos, Cyprus, Nov. 2001.
- [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)

- [Merl 74] MERLIN P. M., « A study of the recoverability of computing systems », PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.
- [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.
- [Newm 03] R.M. Newman and E.I. Gaura. "Formal design of SMIL presentations", In Proc 21st annual international conference on Documentation SIGDOC'2003, 2003.
- [Owen 98] C. B. Owen. Multiple Media Correlation: Theory and Applications . Rapport Technique : PCS-TR98-335, Dartmouth College, Computer Science, Hanover, NH, Juin 1998.
- [Papa 94] Papadias Dimitris et Sellis Timos. "Qualitative Representation of Spatial Knowledge in Two-Dimensional Space." Special Issue on Spatial Databases, Vol. 3(4), pages 479-516,1994
- [Petr 62] C.A. Petri : Communication with automata, Supplement 1 to technical report RADCTR-65-377, Vol 1, New-York 1966, Translated from "Kommunikation mit Automaten" PhD Bonn 1962
- [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)
- [Sann 94] Sannella M., Constraint Satisfaction and Debugging for Interactive User Interfaces. PhD dissertation, August 1994. Revised version published as UW Tech Report 94-09-10
- [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.

- [Séba 06] Sébastien Laborie Jérôme Euzenat Nabil Layaida, 2006. Effective spatial adaptation of SMIL documents. In RFIA 2006, France.
- [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
- [Stem 97] STEMLER (L.K.). – Educational characteristics of multimedia: a literature review. Journal of Educational Multimedia and Hypermedia, vol. 6, n° 3/4, p. 229-359 (1997).
- [Step 04] Stephen J.H. Yang and Al. 2004. "Modeling and analysis of spatio temporal Behavior of multimedia in SMIL", Int Computer Symposium, Taipei, Taiwan
- [Vanb 90] VanbeekPeter, "Reasoning about qualitative temporal information", Proceedings of AAAI, Boston-USA, pages 728-734, 1990
- [Vila 86] M. Vilain and H. Kautz. Constraint propagation algorithms for temporalreasoning. In Proceedings of the Fifth National Conference on Artificial Intelligence, pages 377–382,1986.
- [Vila 86] Vilain Marc et Henry Kautz. "Constraint propagation algorithms for temporal reasoning", In Proceedings of AAAI-86 Philadelphia-USA, pages 132 –144, 1986.
- [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.
- [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>.

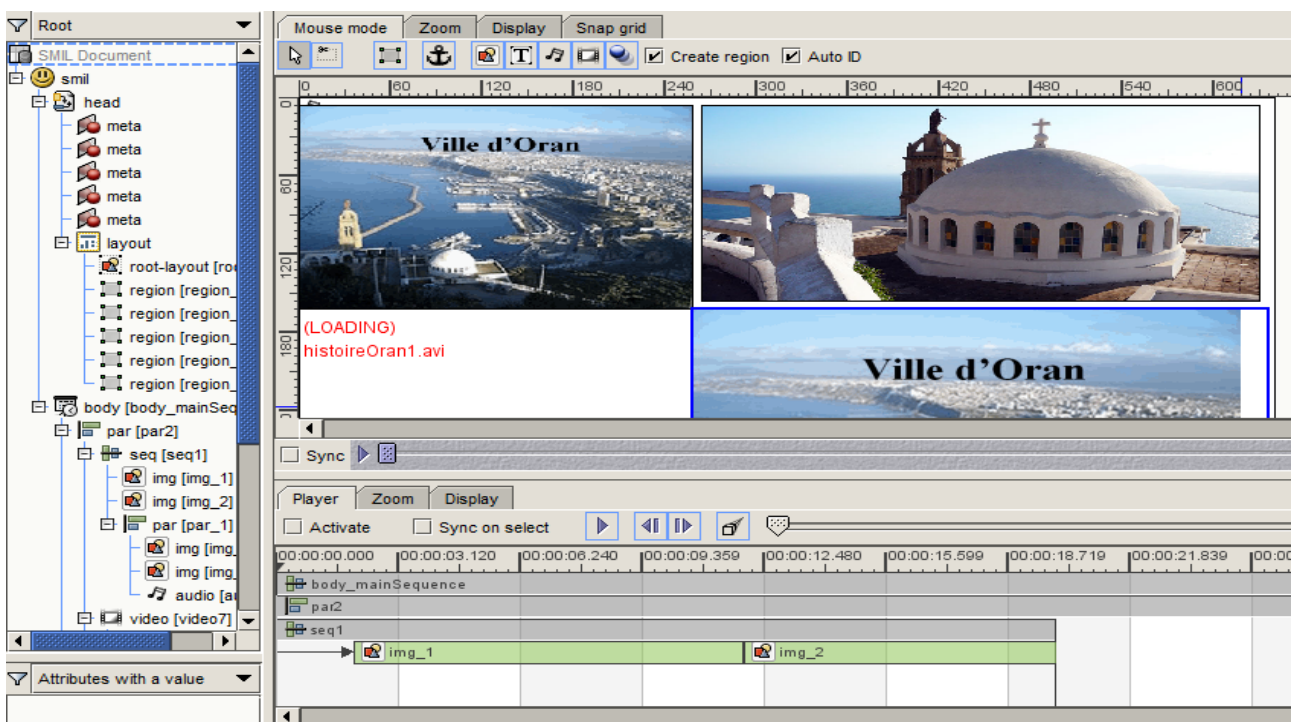
Dans cette annexe A, nous allons expérimenter notre outil de modélisation et de validation d'un document SMIL à travers une étude de cas de la ville d'Oran définie comme suit :

**Etude cas :** Cas du DM interactif de la ville d'Oran

Dans le cadre d'un service proposé par une agence de tourisme qui consiste à donner des Informations sur le *tourisme* à *Oran* : actualité de la ville, les hôtels de la ville, le musée Zabana, le théâtre d'Oran, Santa Cruz, les plages oranaises etc., un touriste peut choisir un endroit à visiter parmi plusieurs fichiers multimédias. Par exemple, lors de la consultation d'un fichier concernant la ville d'Oran, un touriste est en mesure de visualiser la vidéo de la ville grâce à la synchronisation temporelle, tout en se concentrant sur les places intéressantes.

Le DM interactif de la ville d'Oran est composé d'une description textuelle (un texte décrivant l'histoire de la ville d'Oran) et une séquence des médias : audio, image et vidéo décrivant un endroit spécifique de la ville d'Oran. A tout moment, le touriste peut arrêter, re-présenter, revenir au début ou sortir complètement du document.

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



**Figure A.1 :** Edition du DM interactif de la ville d'Oran

## Annexe A

---

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 )"/>
  <layout type="text/smil-basic-layout">
    <root-layout id="rootLayout" width="640" height="636" backgroundColor="white"/>
    <region id="region_go" backgroundColor="#FFFFFF" left="0" top="6" width="636"
      height="156"/>
    <region id="region_x" backgroundColor="#FFFFFF" left="258" top="162" width="378"
      height="228"/>
    <region id="region_nqund" left="0" top="162" width="258" height="228" fit="fill"/>
    <region id="region_wafg" left="6" top="396" width="630" height="228"/>
  </layout>
</head>
<body
  id="body_mainSequence">
  <par dur="40s" endsync="last" id="par2">
    <seq dur="20s" id="seq1">
```

```

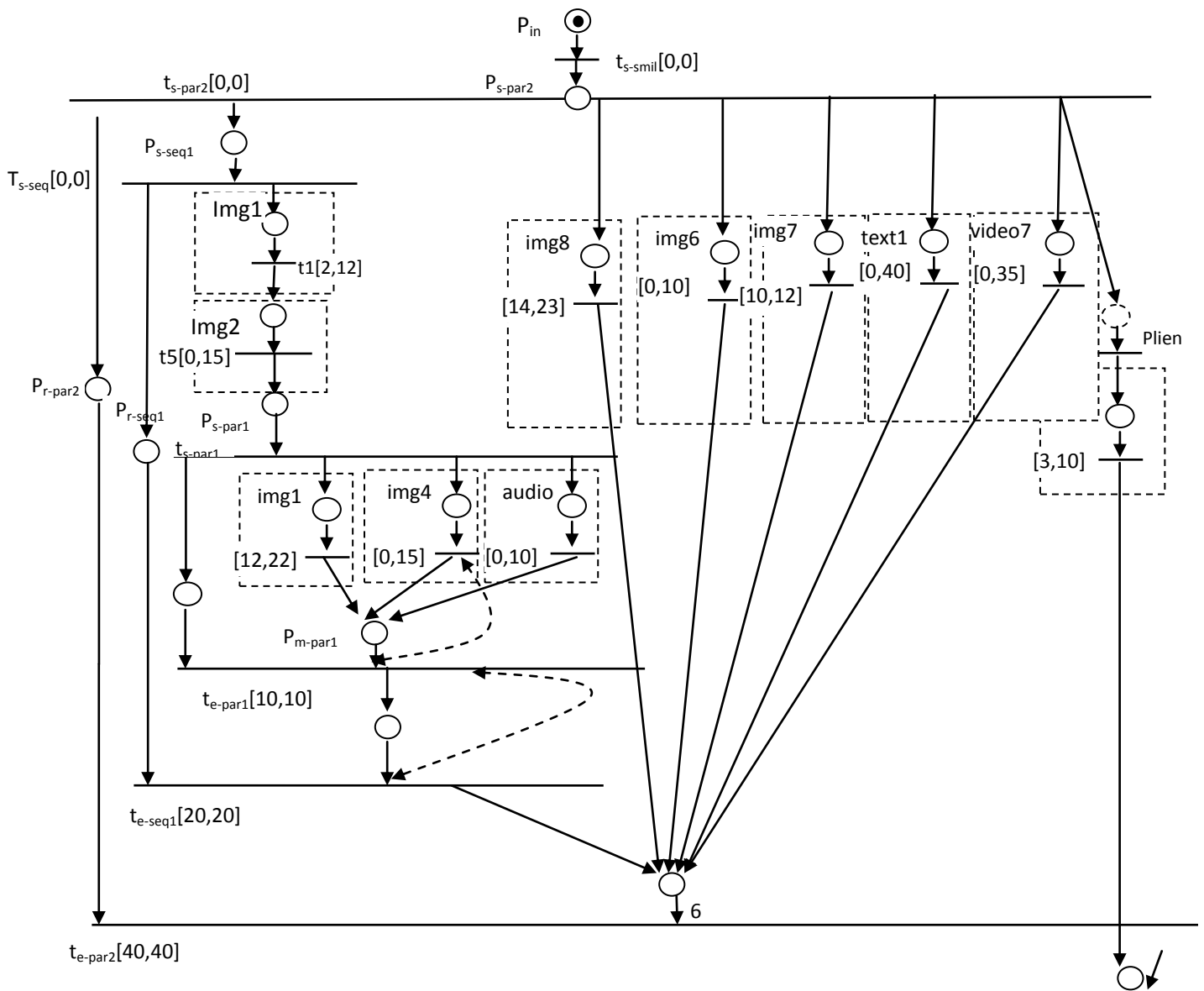

<par dur="10s" id="par1" endsync="first">
    
    
    <audio dur="10s" id="audio" src="muzic.mp3"/>
</par>
</seq>


<text dur="40s" id="text1" src="histoire'doran.rt" region="region_wafg"/>

<video dur="35s" id="video7" src="histoireOran1.avi" region="region_nqund" >
    <area begin="3s" end="10s" id="area1" href="#ref_1" accesskey="2" show="new" />
</video>
</par>
</body>
</smil>
```

Notre outil réalisé traduit automatiquement le code SMIL généré par LimSee 2 en le modèle des t-RdP (voir **Figure A.2**). Nous avons associé une horloge globale pour chaque sous réseau modélisant les éléments <seq> ou <par> pour calculer le temps de tir des transitions du t-RdP.





**Figure A.2:** Exemple d'une modélisation t-RdP correspondante au DM interactif de la ville d'Oran

Le modèle des t-RdP traduisant le code SMIL du DM interactif de la ville d'Oran est affiché comme dans la **Figure A.3** suivante :

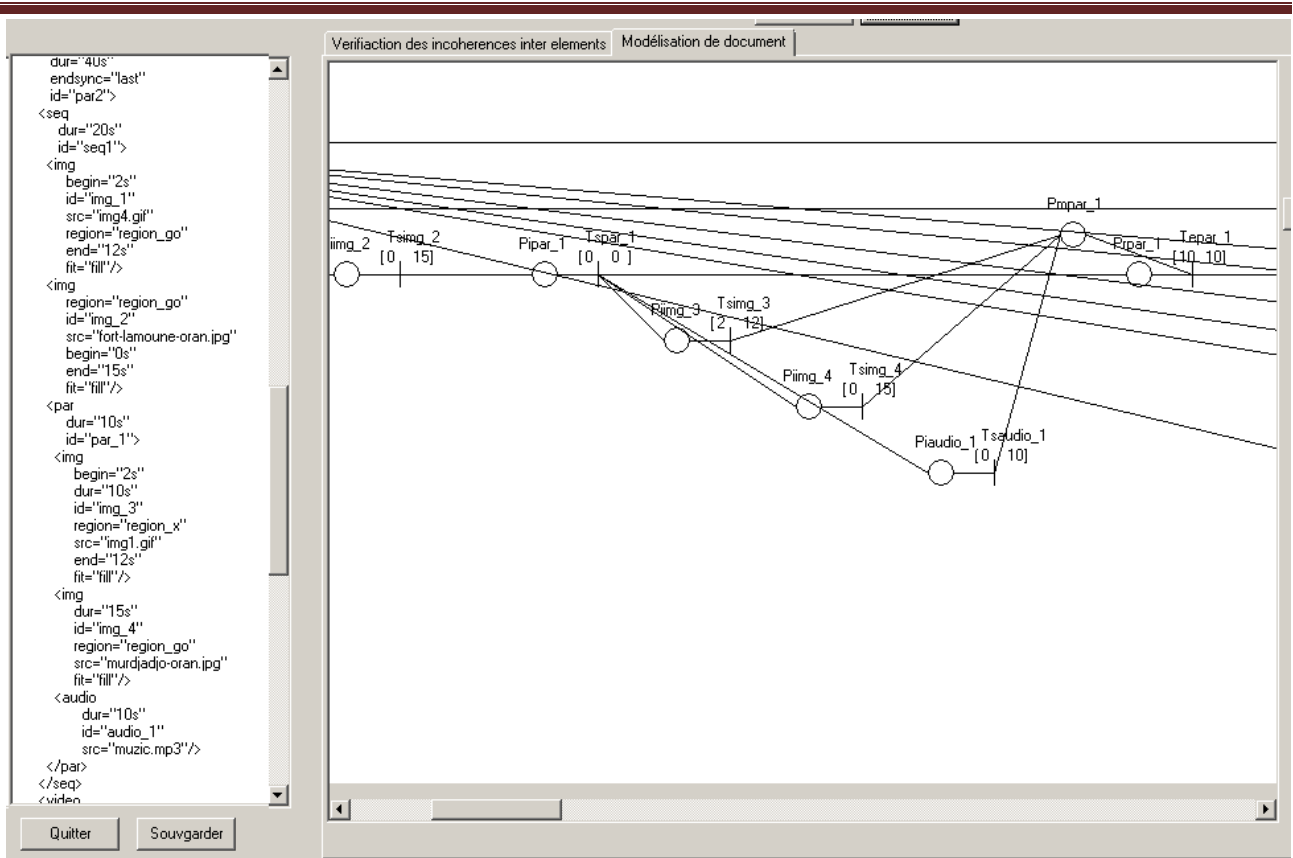


Figure A. 3: Le modèle t-RdP correspondant au DM interactif de la ville d’Oran

Des exemples de vérification et de correction d’incohérences temporelles intra-média sont montrés dans les **Figure A.4** et **Figure A.5**

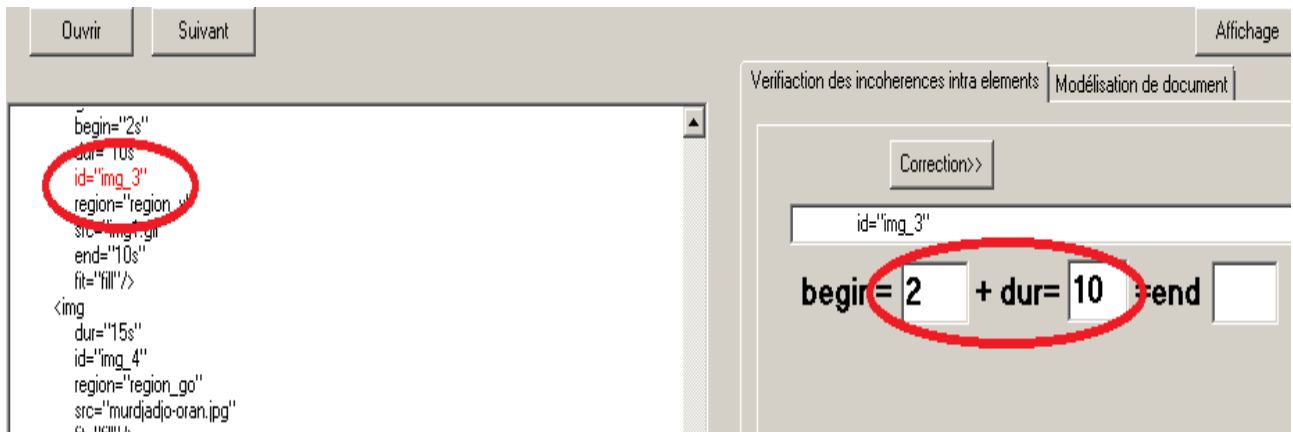


Figure A.4: Cas d’une incohérence intra-média

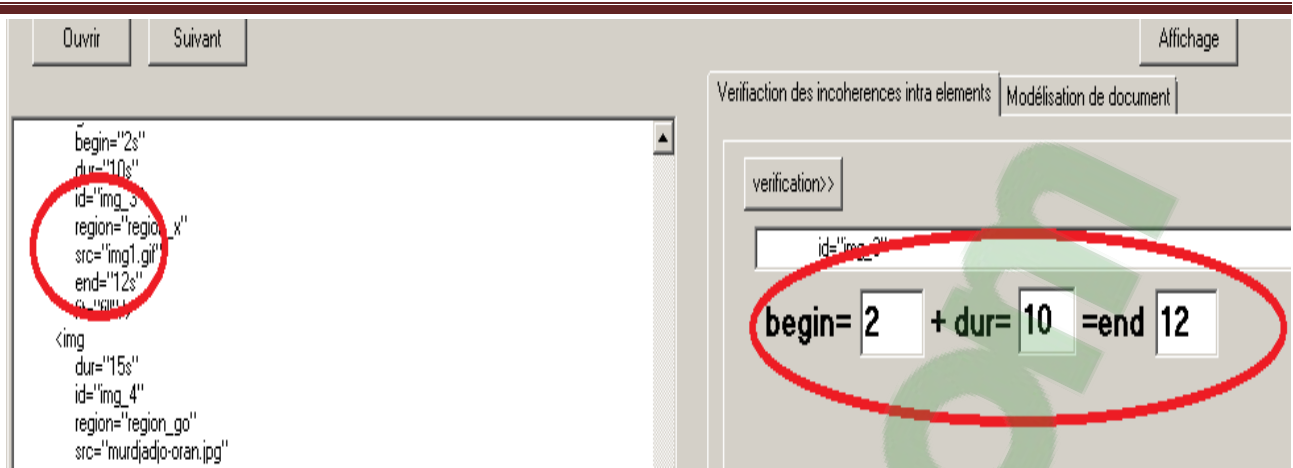


Figure A.5: Cas d'une incohérence intra média validé

Des exemples de vérification et de correction d'incohérences temporelles inter-médias des opérateurs : <seq> et <par> sont montrés dans les Figure A.6 et Figure A.7, Figure A.8 et Figure A.9 suivantes :

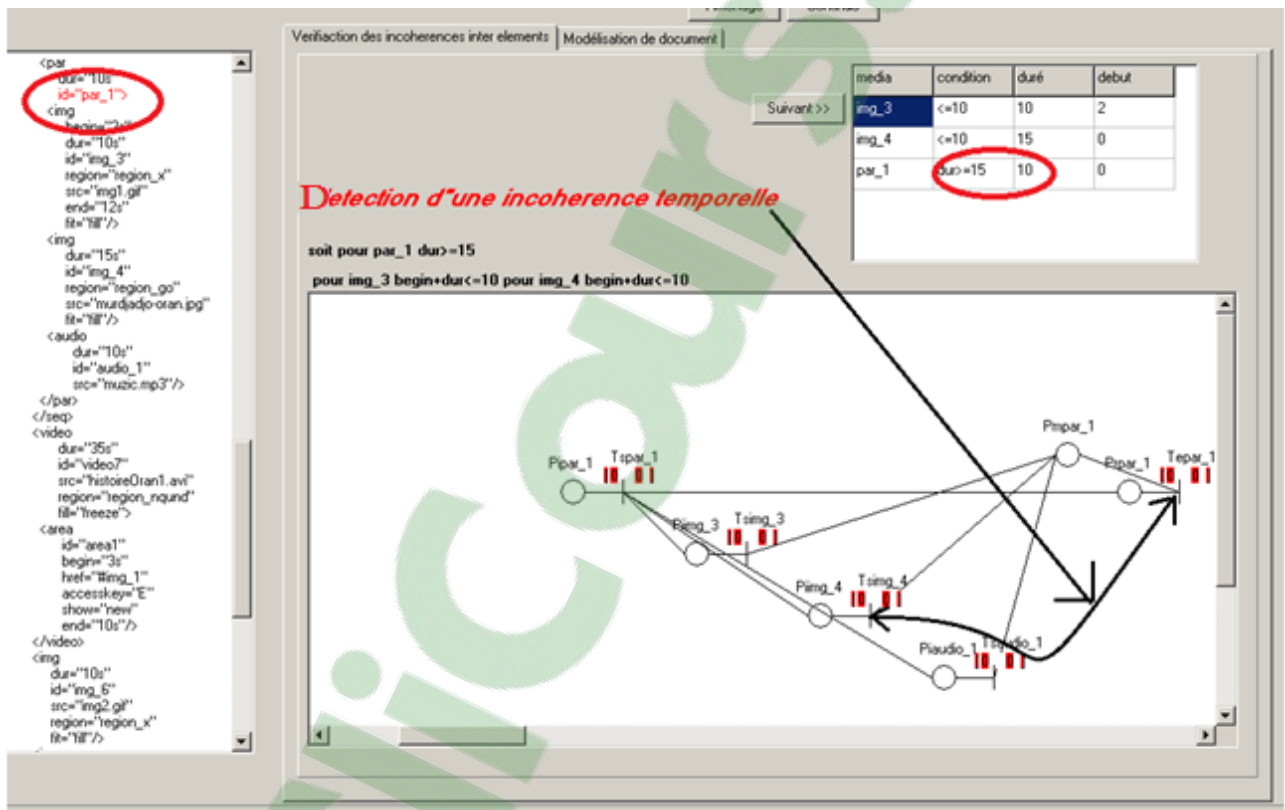


Figure A.6: Cas d'une incohérence temporelle inter-médias de l'élément <par>

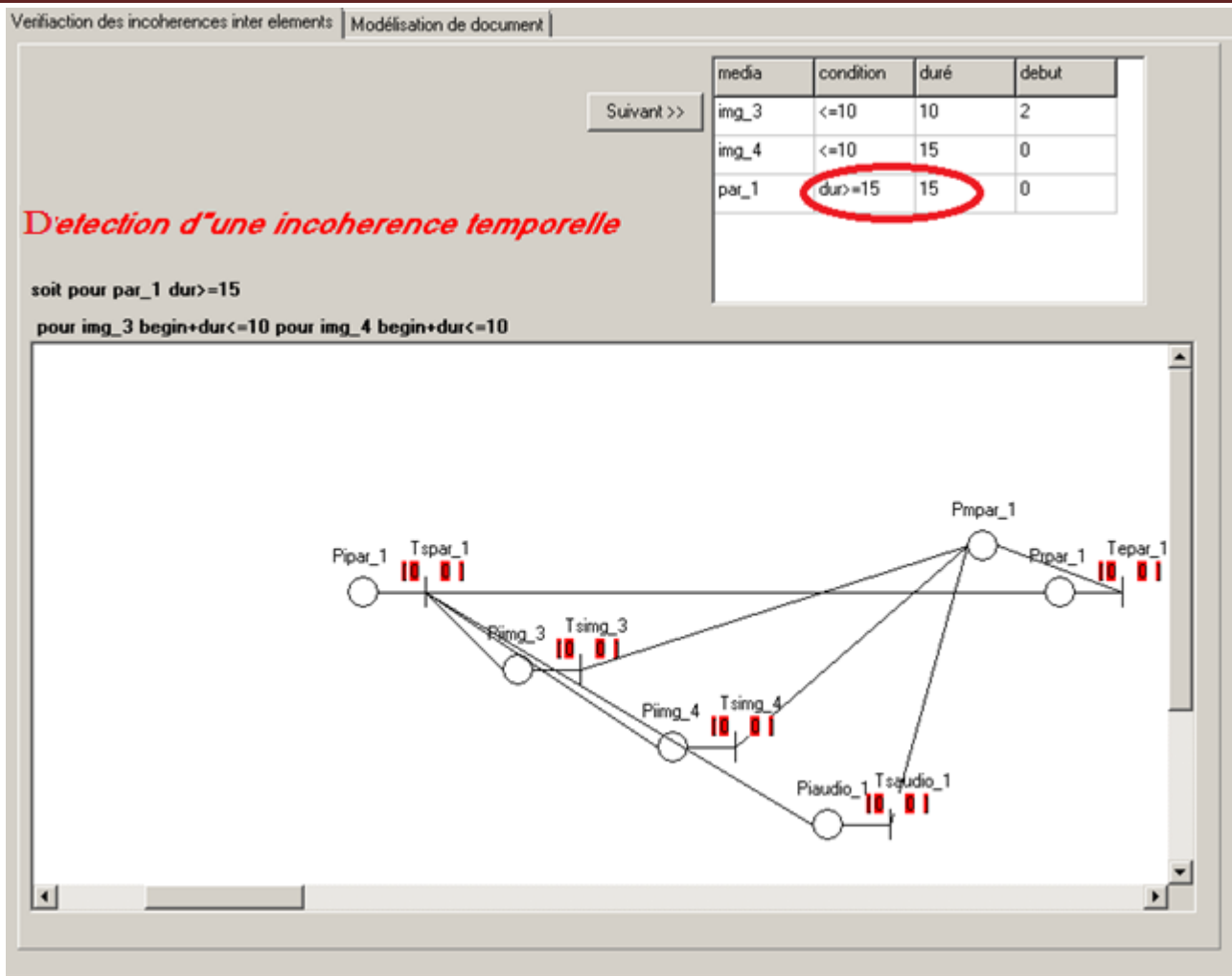


Figure A.7: Cas de validation de l'incohérence temporelle inter-médias de l'élément &lt;par&gt;

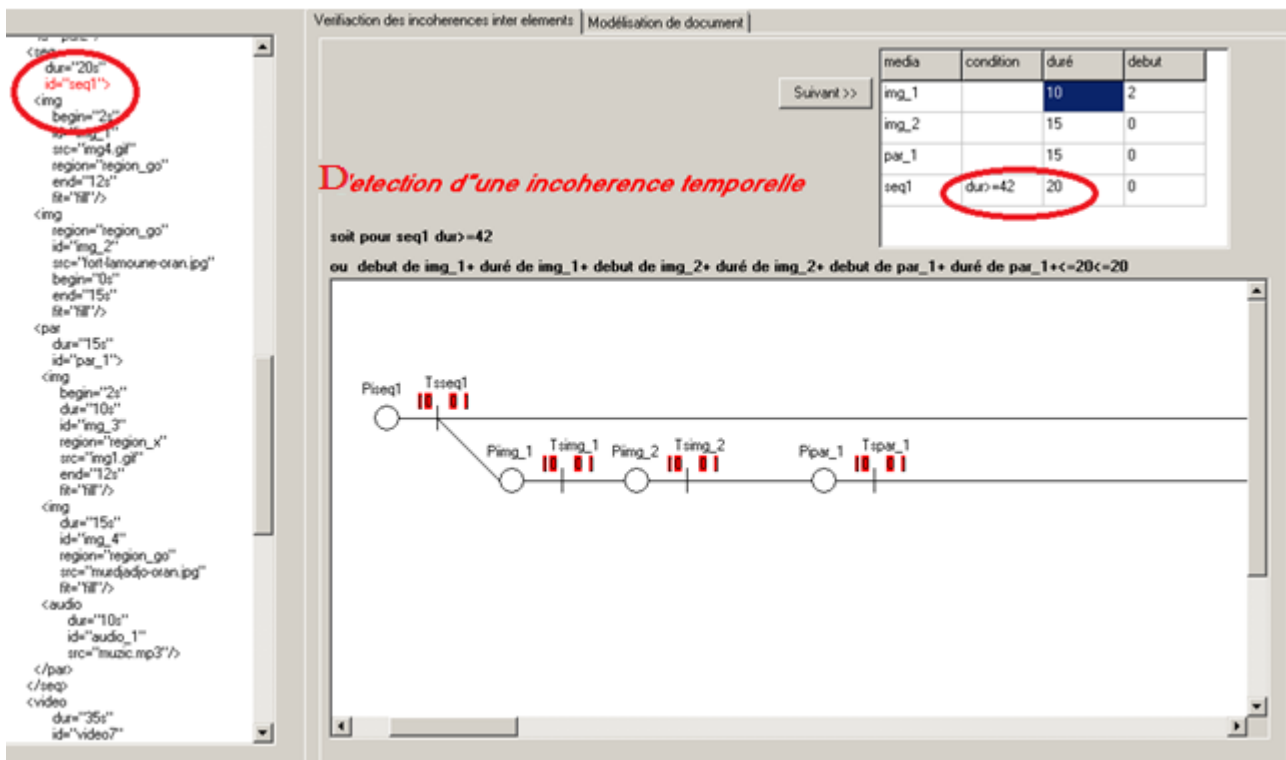


Figure A.8: Cas d'une incohérence temporelle inter-médias de l'élément &lt;seq&gt;

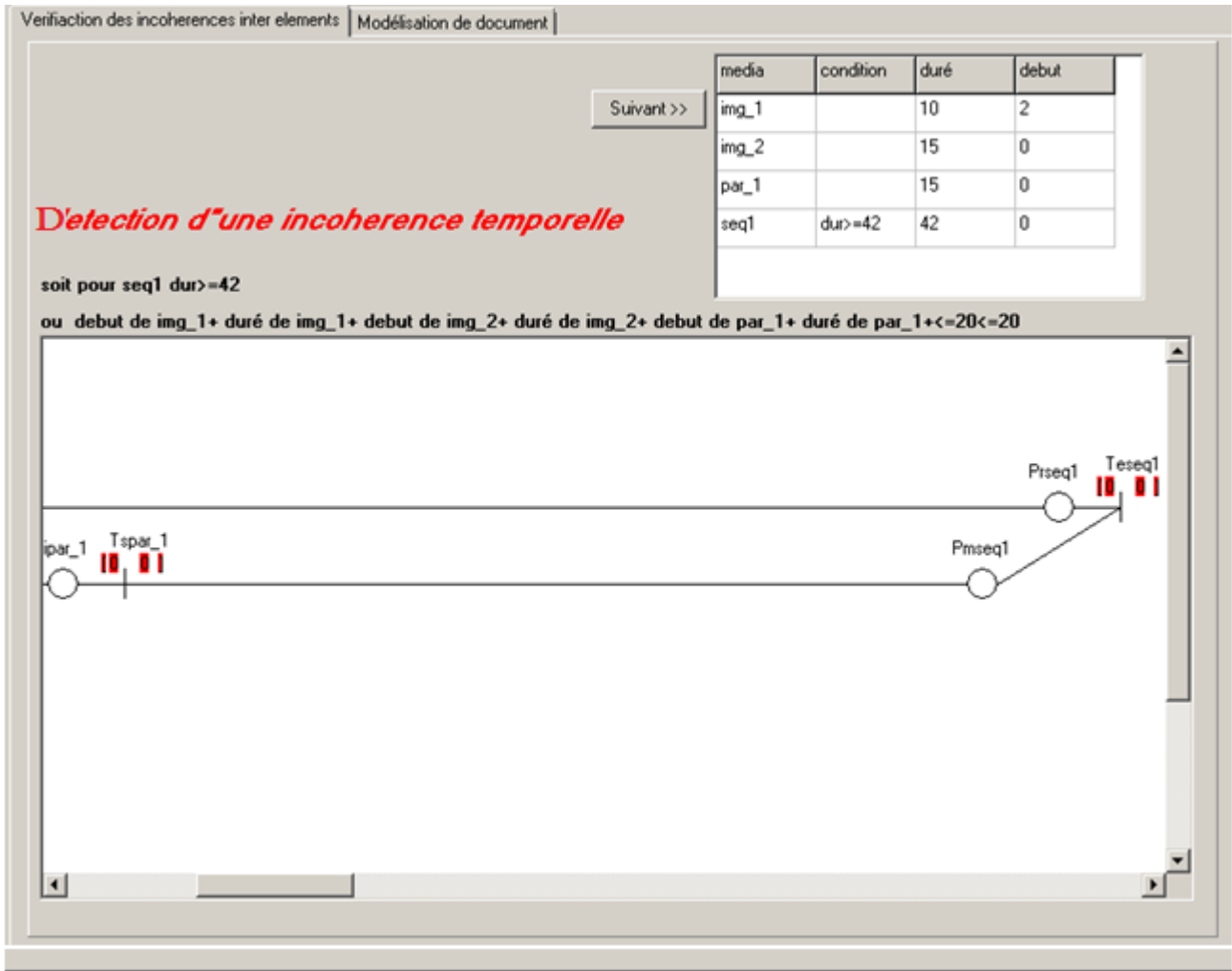


Figure A.9: Cas de validation de l'incohérence temporelle inter-médias de l'élément <seq>

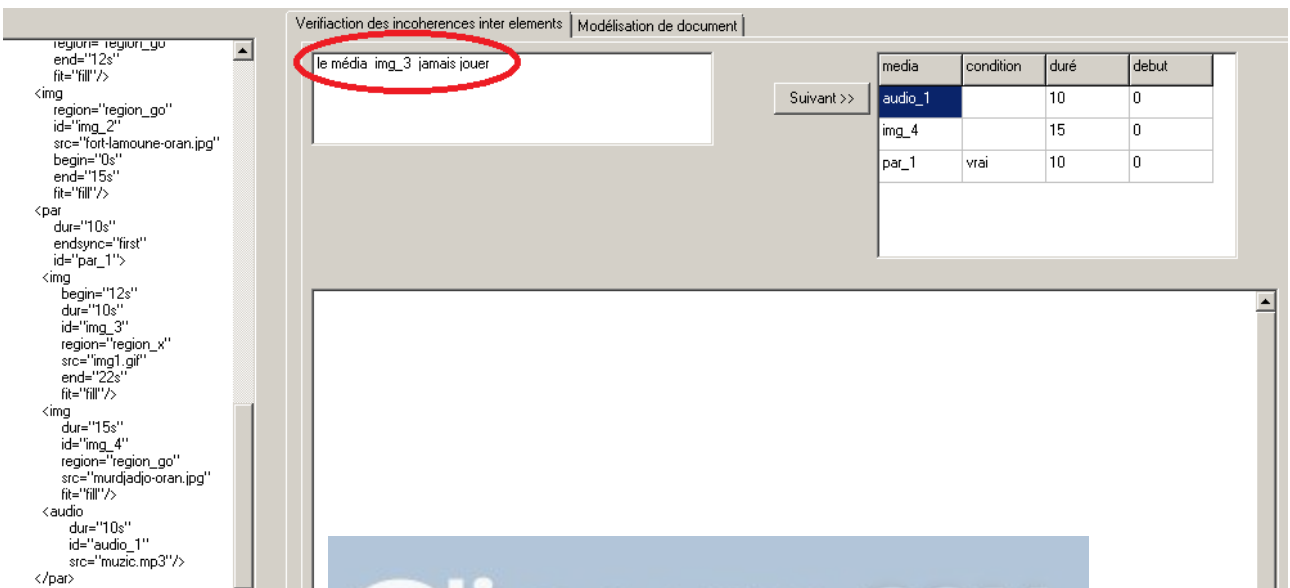


Figure A.10: Cas d'une inconsistance (un média qui n'est jamais joué).

## Annexe A

Une fois notre DM interactif de la ville d'Oran est valide, notre outil permet de le présenter aux touristes qui s'intéressent à visiter Oran (voir **Figure A.11**).



**Figure A.11:** Présentation du document SMIL de la ville d'Oran à l'aide de RealOne

## Résumé

Grâce à la puissance des ordinateurs et au débit des communications actuels, il est possible de combiner différents médias (son, vidéo, texte, images, animations ...) pour produire des documents plus sophistiqués communément appelés Documents Multimédia (DM en abrégé). Ces documents sont devenus de plus en plus répandus et utilisés dans différents domaines. Ils sont caractérisés par l'intégration de la synchronisation de plusieurs média (audio, vidéo, texte, image, ...).

Le langage SMIL (*Synchronized Multimedia Integration Language*) qui est un standard du World Wide Web Consortium (W3C) suscite un intérêt croissant, notamment dans le cadre de l'édition de DM synchronisés. L'aspect temporel est donc un point crucial pour de telles présentations. Ce langage permet aussi d'intégrer d'autres informations d'ordre spatial, hypermédia et des Méta 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. Ce mémoire utilise les réseaux de Pétri temporels de Merlin pour la modélisation et la validation des documents SMIL ce qui contribue à leur production et diffusion.

### Mots clés :

Documents Multimédias (DM); SMIL; Réseaux De Pétri Temporels; Formalisation; Validation; Aspect Temporel; Aspect Spatial; XML; Limsee; Incohérence Temporelle.