

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE  
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE



## THESE

*Présentée pour l'obtention du diplôme de DOCTORAT*

*En : INFORMATIQUE*

*Spécialité : Intelligence artificielle et Bases de Données avancées*

*Par : MAREDJ Azze-Eddine*

*Sujet*

**Spécification et validation des relations temporelles et spatiales dans un document multimédia interactif.**

Soutenue publiquement, le 16 Décembre 2009, devant le jury composé de :

<b>M. BADACHE Nadjib</b>	<i>Professeur, USTHB</i>	<i>Président</i>
<b>Mme. ALIMAZIGHI Zaia</b>	<i>Professeur, USTHB</i>	<i>Directrice de thèse</i>
<b>M. BOUFAIDA Mahmoud</b>	<i>Professeur, U. Constantine</i>	<i>Examineur</i>
<b>Mme BOUFAIDA Zizette</b>	<i>Professeur, U. Constantine</i>	<i>Examineur</i>
<b>M. BELKHIR Abdelkader</b>	<i>Maitre de conférences, USTHB</i>	<i>Examineur</i>
<b>M. KAZAR Okba</b>	<i>Maitre de conférences, U. Biskra</i>	<i>Examineur</i>

## ***Remerciements***

Je tiens à remercier Mme Ali Mazighi Zaia, doyenne de la faculté d'électronique et Professeur à l'USTHB, pour m'avoir encadré et conseillé tout au long de ces années de thèse.

Je remercie M. Badache Nadjib, Directeur du CERIST et Professeur à l'USTHB, pour m'avoir fait l'honneur de présider le jury de cette thèse.

Je remercie vivement Madame Boufaïdi Zizette et Messieurs Boufaïdi Mahfoud, Kazar Okba et Belkhir Abdelkader d'avoir accepté d'être les examinateurs de ce travail.

Je remercie également le Centre de Recherche sur l'Information Scientifique et Technique (CERIST) pour le soutien qu'il m'a accordé durant ces années de thèse.

Des remerciements particuliers vont à MM Sadallah Madjid et Tonkin Nouredine pour leur aide.

Mes remerciements vont également à toute personne m'ayant soutenu et encouragé pendant les moments difficiles.

# Table des matières

## Chapitre 1

### Introduction

1. Introduction .....	7
2. Motivation et objectifs.....	8
3. Plan de la thèse .....	9

## Chapitre 2

### Introduction aux systèmes d'édition et de présentation de documents multimédia

1. Introduction .....	11
2. Caractéristiques d'un SEPDMI .....	12
2.1 Hétérogénéité des médias .....	13
2.2 Composition multimédia .....	13
2.3 Intégration de données multimédia hétérogènes .....	15
2.4 Synchronisation multimédia.....	15
2.4.1 Synchronisation intra-média.....	15
2.4.2 Synchronisation inter-médias .....	16
2.4.3 Synchronisation avec l'environnement.....	17
2.5 Emplacement spatial des médias .....	17
2.6 Recouvrement spatial des médias.....	17
2.7 Indéterminisme de la présentation multimédia.....	18
3. Besoins de l'auteur et du lecteur .....	18
3.1 Besoins de l'auteur .....	18
3.2 Besoins du lecteur.....	19
4. Fonctions des systèmes multimédia .....	20
5. Étude des systèmes multimédia.....	20
5.1 Les approches opérationnelles.....	20
5.1.1 Axe temporel absolu.....	20
5.1.2 Langages de scripts .....	22
5.1.3 Structures de graphes.....	25
5.2 Les approches à base de contraintes.....	31
6. Conclusion.....	32

## Chapitre 3

### Modèle de documents multimédia

1. Introduction .....	34
2. Concepts du modèle .....	34
2.1 Le média .....	34
2.1.1 Modélisation logique d'un média.....	34
2.1.2 Les attributs d'un média .....	35
2.2 Le document Multimédia .....	38

2.2.1	Le document Primaire .....	38
2.2.2	Le parcours de lecture .....	38
2.2.2.1	Le document dérivé .....	38
2.2.2.2	La scène .....	38
2.2.2.3	Le parcours de lecture.....	39
3.	Modèle de document multimédia .....	40
3.1	La dimension logique .....	40
3.2	La dimension spatiale .....	41
3.2.1	La taille de l'emplacement .....	41
3.2.2	Les informations de l'emplacement .....	41
3.3	La dimension temporelle .....	41
3.4	La dimension hypermédia .....	42
3.4.1	Navigation .....	42
3.4.1.1	Navigation par liens.....	42
3.4.1.2	Navigation libre .....	43
3.4.2	Fonctions de manipulation .....	43
3.4.2.1	Fonctions de manipulation d'un document primaire ou d'un parcours.....	43
3.4.2.2	Fonctions de manipulation d'un média .....	43
4.	Conclusion.....	43

## **Chapitre 4**

### **Modèles de spécification des relations temporelles et spatiales**

1.	Introduction .....	44
2.	Scénario d'un document multimédia.....	44
2.1	Définitions .....	44
2.2	Compositions d'un scénario .....	45
3.	Critères d'évaluation d'un modèle de spécification .....	45
4.	Modèle temporel.....	46
4.1	Modélisation de l'information temporelle de base .....	46
4.2	Domaine de validité temporelle.....	46
4.3	Représentation du déroulement temporel d'un scénario.....	47
4.4	Expression des relations temporelles.....	47
4.4.1	Modèles à base d'instants .....	47
4.4.2	Modèle à base d'intervalles .....	48
4.4.3	Choix entre algèbres d'intervalles et d'instants.....	49
4.5	Modèle temporel proposé .....	49
4.5.1	Le modèle proposé .....	50
4.5.1.1	Degré d'expressivité du modèle .....	50
4.5.1.2	Simplicité d'utilisation et intuitivité .....	53
4.5.1.3	L'indéterminisme.....	53
5.	Modèle spatial .....	54
5.1	Modèles de présentation spatiale.....	54
5.1.1	Les modèles orientés intervalles.....	54
5.1.2	Les modèles orientés régions .....	54
5.1.2.1	Relations topologiques.....	54
5.1.2.2	Relations directionnelles.....	55
5.1.2.3	Discussion.....	55
5.2	Modèle spatial proposé.....	56

5.2.1	Degré d'expressivité du modèle .....	56
5.2.1.1	Définition de la distance .....	56
5.2.1.2	Choix des relations.....	57
5.2.2	Précision des emplacements.....	60
5.2.3	Simplicité d'utilisation et intuitivité.....	61
5.2.4	Spécification du recouvrement désiré .....	61
6.	Conclusion.....	62

## **Chapitre 5**

### **Analyse et synthèse des spécifications**

1.	Introduction .....	64
2.	Présentation des systèmes de contraintes .....	64
2.1	Contrainte .....	64
2.2	Système de contraintes .....	64
2.3	Représentation graphique .....	65
3.	Caractéristiques recherchées pour un résolveur dans le domaine de l'EPDMI.....	65
4.	Les différentes approches de résolution des systèmes de contraintes .....	67
4.1	Les approches locales .....	67
4.2	Les approches globales.....	69
4.2.1	Algorithme de backtrack [10].....	69
4.3	Approches basées sur le simplexe .....	70
5.	Le résolveur Cassowary .....	71
5.1	Principe de fonctionnement.....	71
5.1.1	La phase de prétraitement.....	71
5.1.2	La phase de résolution.....	72
5.2	Prise en comptes des critères pour le domaine de l'EPDMI .....	73
5.2.1	La gestion des équations et des inéquations.....	73
5.2.2	l'incrémentalité.....	73
5.2.3	Le maintien de la solution .....	74
5.2.4	Contrôle du choix de la solution.....	74
5.2.5	La complétude .....	75
5.2.6	La gestion des cycles.....	75
5.2.7	La directionnalité.....	75
5.2.8	L'intégration des trois aspects.....	75
5.2.9	Les performances temporelles.....	75
5.3	La fonction objective.....	75
6.	Conclusion.....	75

## **Chapitre 6**

### **Gestion du recouvrement spatial**

1.	Introduction .....	77
2.	Les approches de gestion du recouvrement spatial .....	77
2.1	Approche basée sur les rectangles libres.....	78
2.2	Approche basée sur les contraintes disjonctives .....	78
2.3	Discussion .....	80
3.	Approche proposée.....	80
3.1	Opérations pouvant provoquer un recouvrement .....	81

3.2	Les variables du système de contraintes.....	81
3.3	Recouvrement désiré .....	81
3.4	Tolérance d'un recouvrement.....	81
3.5	Réduction des dimensions d'un média.....	82
3.6	Modification des attributs temporels d'un média.....	83
4.	Adaptation de l'approche disjonctive.....	83
4.1	La levée de la condition de non recouvrement initial.....	83
4.2	Génération sélective des systèmes de contraintes disjonctives .....	84
5.	Conclusion.....	84

## Chapitre 7

### Architecture et mise en œuvre

1.	Introduction .....	85
2.	Architecture de MediaStudio.....	85
2.1	Principe de conception .....	85
2.2	Architecture générale du système.....	86
3.	Conditions et règles d'édition.....	88
3.1	Paramétrage et contrôle a priori .....	88
3.1.1	Paramétrage par défaut du système .....	88
3.1.2	Les contrôles a priori.....	88
3.2	Nombre de relations entre les médias.....	88
3.3	Concept d'activation/désactivation .....	89
3.4	Règles de construction d'un parcours de lecture.....	89
3.5	Les règles d'édition .....	90
3.5.1	L'insertion .....	90
3.5.2	La Création.....	90
3.5.3	La modification .....	90
3.5.4	Suppression .....	91
3.5.5	Gestion des documents.....	92
4.	Représentation des contraintes dans MediaStudio .....	92
4.1	Traduction des informations en contraintes .....	93
4.1.1	Traduction des relations temporelles.....	93
4.1.2	Traduction des relations spatiales.....	94
4.1.3	Traduction des informations intrinsèques .....	95
4.1.3.1	Contraintes temporelles intrinsèques.....	95
4.1.3.2	Contraintes spatiales intrinsèques.....	95
5.	Mise en œuvre .....	96
5.1	Interface utilisateur de MediaStudio .....	96
5.2	L'édition .....	97
5.3	Vérification de la cohérence.....	98
5.3.1	Le résolveur cassowary .....	98
5.3.1.1	Procédures principales du résolveur.....	98
5.3.2	Vérification de la cohérence des relations.....	99
5.3.3	Vérification de la cohérence des parcours de lecture .....	100
5.4	Gestion du recouvrement spatial .....	101
5.4.1	Procédures de mise en œuvre .....	101
6.	Conclusion.....	106

## Chapitre 8

### Systeme de présentation multimédia

1. Introduction .....	107
2. Caractéristiques d'une présentation multimédia .....	107
2.1. Hétérogénéité des médias .....	107
2.2. Dynamicité de la présentation multimédia .....	107
2.3. Synchronisation multimédia .....	107
2.3.1. Synchronisation inter médias .....	108
2.3.2. Synchronisation environnementale .....	108
3. Fonctions du système de présentation multimédia .....	108
3.1. Architecture du système de présentation .....	108
3.2. L'ordonnanceur .....	109
3.2.1. La structure d'exécution .....	109
3.2.2. Fonctionnement de l'ordonnanceur .....	111
3.2.3. La gestion de la synchronisation .....	111
3.3. Gestionnaire des médias .....	112
3.4. Gestionnaire des interactions .....	113
3.4.1. Le contexte de présentation .....	114
3.4.2. Manipulation du contexte de présentation .....	114
3.4.3. Navigation .....	114
3.4.3.1. Navigation par liens .....	114
3.4.3.2. Navigation libre .....	115
3.4.4. Manipulation d'un média .....	115
4. Gestion de l'indéterminisme .....	115
4.1. À l'arrêt complet d'un média .....	116
4.2. Lors de manipulation d'un média .....	116
5. Conclusion .....	117

## Chapitre 9

### Conclusion

1. Rappel des objectifs .....	118
2. Démarche suivie et bilan scientifique .....	119
3. Bilan et évaluation de la réalisation .....	120
4. Perspectives .....	120
4.1 Aspect recherche .....	121
4.2 Aspect développement .....	121

<b>Bibliographie</b> .....	122
----------------------------	-----

### Annexes

Annexe 1 .....	130
Annexe 2 .....	135

# Chapitre 1

## Introduction

### 1. Introduction

Les percées technologiques en matière de multimédia ont permis d'accroître les possibilités d'interaction entre l'homme et la machine. La manipulation digitale de vidéos, de sons et d'images sur 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 des vidéos et du son. Ces nouveaux types de documents électroniques sont communément appelés *documents multimédia* [57].

L'édition et la présentation des documents électroniques ont jusqu'ici été effectuées 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. L'intégration de cette nouvelle dimension dans la structure globale d'un document a défini un nouveau type de présentation spécifique aux documents multimédia appelé *présentation multimédia* [56]. Par conséquent, un ensemble de besoins est à prendre en charge depuis l'édition jusqu'à la présentation :

- Des mécanismes de spécification et de vérification de la cohérence de l'information temporelle doivent être définis.
- La dimension spatiale doit être reconsidérée, elle ne se limite plus aux seuls aspects typographiques. L'auteur doit pouvoir spécifier les emplacements qu'il désire pour ses médias. Des mécanismes de spécification et de vérification de la cohérence de l'information spatiale sont donc nécessaires.
- L'information temporelle et spatiale peut conduire à des recouvrements spatiaux qui peuvent altérer considérablement la qualité de service d'une présentation. Une gestion automatisée du recouvrement spatial doit être considérée.
- Des techniques de gestion de l'indéterminisme d'une présentation (le décalage temporel entre la présentation et le scénario spécifié par l'auteur) doivent être définies.
- Lors de la présentation, la dimension temporelle confère aux documents multimédia une certaine complexité de perception. Des fonctions de navigations spécifiques sont donc nécessaires.

Les standards et les modèles employés pour représenter les documents classiques sont devenus inadapés pour représenter de tels documents. De ce fait, de nouveaux standards comme HyTime, MHEG, HyperODA et SMIL émergent pour essayer de les compléter. Mais les standards ne suffisent pas, ceci s'explique en grande partie par la difficulté de leur utilisation ou au manque de réalisations pour les valider [51].

Le travail présenté dans cette thèse est une contribution au domaine de l'édition et de la présentation des documents multimédia interactifs.

## 2. Motivation et objectifs

Dans les standards et les systèmes proposés pour l'édition et la présentation de documents multimédia le contrôle du déroulement de la présentation d'un document multimédia est fondé sur une approche événementielle. Cette approche s'appuie sur la spécification des événements et les actions qui y sont associées. Dans ce cas, la construction d'une présentation se fait au moyen de langages de programmation, selon une approche impérative, pour définir des enchaînements temporels et spatiaux [57]. Cette approche comporte plusieurs inconvénients majeurs, comme l'inadaptation de la programmation à la nature incrémentale du processus d'édition, la difficulté de la maintenance des documents ainsi produits, la faible portabilité des documents, et enfin, les problèmes qu'ont les auteurs non-informaticiens pour les maîtriser.

Dans les standards et les systèmes actuels, le document génère une présentation unique, l'adaptation du contenu n'est pas considérée, alors qu'il existe actuellement des domaines, tel que la conception de cours interactifs, où cette caractéristique peut s'avérer très utile.

La totalité des standards et des systèmes considèrent le média dans sa globalité, c'est-à-dire comme une entité indissociable, alors qu'on devrait pouvoir utiliser une partie d'un média sans recourir à sa création.

L'indéterminisme qui peut engendrer des situations de désynchronisation voire des blocages est très peu considéré. Par exemple, il est difficile de contrôler l'interaction de l'utilisateur dans ces documents. Il est donc nécessaire que le système de présentation adopte des techniques qui peuvent le gérer.

Les techniques de navigation utilisées dans les systèmes hypermédia actuels restent très insuffisantes pour profiter des informations temporelles des documents. Des fonctions de navigation temporelles avancées doivent être fournies afin de mieux exploiter la présentation d'un document.

Enfin, les travaux menés dans le domaine traitent essentiellement de la dimension temporelle, alors que la conception de tels systèmes devrait considérer d'emblée les quatre dimensions afin de mutualiser au mieux leur conception et leur implémentation.

Ces motivations nous ont conduites à aborder notre travail de thèse dans un cadre applicatif où il est fondamental de confronter les propositions théoriques avec la réalité de l'application. C'est pourquoi, nous avons tenté de mener tout au long de cette thèse une activité équilibrée entre théorie et application. Deux types de résultats sont donc attendus :

1. Des résultats théoriques porteront sur la proposition d'un modèle logique de documents multimédia interactifs intégrant les quatre dimensions et permettant la définition de parcours de lecture selon le profil. Sur la proposition de deux modèles à base de relations pour la spécification des informations temporelles et spatiales qui allient facilité d'utilisation et puissance d'expressivité. Sur la proposition d'une approche automatisée de gestion du recouvrement spatial. Sur le choix d'une approche intégrée pour l'analyse et la synthèse des informations temporelles et spatiales et la gestion du recouvrement. Et enfin, sur la proposition d'un ensemble de mécanismes pour la navigation.

2. Des résultats pratiques, sous la forme d'un prototype d'édition et de présentation de documents multimédia interactifs, qui permet de créer et de présenter des documents d'une complexité significative.

### **3. Plan de la thèse**

Ce mémoire de thèse est organisé en deux grandes parties : la première, constituée du deuxième chapitre, fixe le contexte de notre travail en faisant l'analyse des besoins et des applications existantes dans le domaine de l'édition et de la présentation de documents multimédia interactifs; la deuxième, qui comprend le reste des chapitres, constitue notre réponse aux problèmes soulevés ci-dessus. Nous détaillons dans la suite le contenu de chacun de ces chapitres :

#### **Chapitre 2**

Le chapitre 2 introduit les notions de base sur les systèmes d'édition et de présentation multimédia. Cette étude aborde dans une première partie les caractéristiques d'une présentation multimédia que l'on trouve dans la littérature auxquelles s'ajoutent la reconsidération de la dimension spatiale et le problème du recouvrement spatial, desquelles un ensemble de besoins de l'auteur et du lecteur sont déterminés. La deuxième partie présente les fonctions de base qui doivent être fournies par un système d'édition et de présentation afin de satisfaire les besoins identifiés. La dernière partie présente une étude critique des approches de spécification et de présentation multimédia à travers les standards/systèmes HyTime, MHEG, PREMO, SMIL et MADEUS de laquelle nous avons opté pour une approche à base de relations pour la spécification des informations temporelles et spatiales, et une approche à base de contraintes pour la vérification de la cohérence et la génération de solutions de placements temporel et spatial des média.

#### **Chapitre 3**

Dans ce chapitre, nous décrivons le modèle de document proposé. Cette description porte sur les quatre dimensions d'un document multimédia interactif (temporelle, spatiale, logique et hypermédia), les informations retenues pour chaque dimension et les principes qui régissent la construction de parcours de lecture.

#### **Chapitre 4**

Après l'étude des approches de spécification à base de relations existantes dans la littérature et la présentation des caractéristiques servant à l'évaluation d'un modèle temporel et d'un modèle spatial dans notre domaine d'étude, nous décrivons les modèles temporel et spatial proposés sur la base de ces dernières.

#### **Chapitre 5**

Ce chapitre aborde dans une première partie la définition des systèmes de contraintes, une présentation des critères nécessaires pour un solveur de contraintes dans le domaine étudié et une étude critique de quelques approches de résolution des systèmes de contraintes. Dans sa deuxième partie, nous présentons le solveur Cassowary, que nous avons retenu sur la base de ces critères.

#### **Chapitre 6**

Une fois le problème du recouvrement spatial des média et ses conséquences sur la qualité de service de la présentation présenté, nous ferons dans ce chapitre une proposition d'une approche pour une gestion automatisée de ce problème.

## Chapitre 7

Dans ce chapitre, nous décrivons le système d'édition et de présentation MediaStudio développé dans le cadre de cette thèse. Cette description porte sur son principe de conception, son architecture générale et sa mise en œuvre à travers l'ensemble des fonctions retenues qui gèrent la création et la présentation d'un document/parcours de lecture multimédia interactifs.

## Chapitre 8

Ce chapitre est consacré à la description du système de présentation adopté dans MediaStudio. Nous détaillerons notamment l'ordonnanceur qui est au cœur du système de présentation, la gestion des interactions utilisateur à travers les fonctions de navigations et de manipulation de documents et la gestion de l'indéterminisme qui en découle.

## Chapitre 9

La conclusion résume les points essentiels de ce travail, fait le bilan de la réalisation du système MediaStudio et présente les perspectives de recherche et développement suggérés par ce travail.

# Chapitre 2

## Introduction aux systèmes d'édition et de présentation de documents multimédia

### 1. Introduction

Savoir manipuler de l'information multimédia est de nos jours une qualité de plus en plus recherchée. À un point tel, que le terme multimédia est devenu une notion courante et employée de façon très intuitive et très générale [57]. Plusieurs définitions ont été proposées dans la littérature pour définir le terme système multimédia ou application multimédia. Toutefois, la définition proposée par Blakowski nous paraît la plus appropriée compte tenu qu'elle s'appuie sur trois critères qui permettent de considérer les différents systèmes par rapport au qualifiant multimédia :

- le nombre de médias manipulés dans l'application, comme l'audio, la vidéo, le texte, et les interactions de l'utilisateur,
- la nature temporelle des médias supportés (continus comme la vidéo, statique comme le texte, etc.),
- et le niveau d'intégration de ces différents médias au sein de l'application.

En combinant ces trois critères, on obtient alors la définition suivante : un système est qualifié de multimédia s'il supporte le traitement intégré de plusieurs médias dont au moins un est de nature temporisée [12]. Nous appellerons par la suite *document multimédia*, tout document interactif répondant à cette définition.

Actuellement, de tels systèmes sont de plus en plus répandus et utilisés dans différents domaines, par exemple l'enseignement, la visioconférence, les présentations commerciales, etc. Toutefois, contrairement aux documents statiques, le processus de création de documents multimédia est un processus complexe et difficile à spécifier. Il devient donc indispensable que l'auteur soit aidé dans cette tâche d'édition par un logiciel. Ces logiciels sont appelés *système d'édition et de présentation de documents multimédia interactifs* (SEPDMI).

Les fonctions principales d'un tel système consistent à éditer et à présenter les médias de façon synchronisée selon un scénario préalablement spécifié. Garantir la synchronisation est un problème qui est au cœur de nombreux travaux de recherche. Ainsi, des études sont effectuées sur les modes de synchronisation dans les documents multimédia, notamment au niveau de la spécification (comme les études présentées dans [108], [12] et [57] sur les modèles temporels qui peuvent être utilisés pour spécifier la synchronisation entre les médias) et au niveau de la présentation (comme les études présentées dans [36] sur la gestion des paramètres qui affectent la qualité d'une présentation, dans [98] sur l'effet de la désynchronisation sur la perception du lecteur, et dans [97] sur les services dont la présentation multimédia a besoin au niveau système d'exploitation).

De plus, la dimension spatiale doit être reconsidérée, elle ne se limite plus aux seuls aspects typographiques ou de mise en page d'un média. L'auteur doit pouvoir spécifier les emplacements qu'il désire pour ses médias. Un modèle de spécification de relations spatiales et des mécanismes

de vérification de leur cohérence sont donc nécessaires. Très peu de travaux considèrent ce besoin, ils se limitent généralement à une spécification absolue de l'emplacement d'un média ou à quelques relations spatiales dont la puissance d'expressivité reste relativement faible.

Par ailleurs, comme les médias s'affichent chacun pendant son temps de présentation et qu'ils peuvent occuper un même emplacement, le risque de recouvrement spatial des médias est omniprésent. Ceci peut altérer considérablement la qualité de service de la présentation. De là, il faut en tenir compte en mettant en place une procédure automatisée pour sa gestion.

L'objectif de ce chapitre est d'introduire d'abord les notions de base qui caractérisent un SEPDMI avec un intérêt particulier pour les problèmes de la synchronisation temporelle, de l'emplacement spatial et du recouvrement spatial. Dans la section 3, nous recensons les besoins de l'auteur et du lecteur dans de tels systèmes. Nous présentons, dans la section 4, les fonctions attendues d'un SEPDMI garantissant les caractéristiques définies et prenant en charge les besoins retenus dans les sections précédentes. Dans la section 5, nous étudions les principes de quelques standards et systèmes et les façons dont ils traitent les fonctions précédemment identifiées. Enfin, nous concluons avec l'ensemble des constats établis et les grandes orientations retenues qui constitueront les lignes directrices de notre contribution au domaine des SEPDMI.

## **2. Caractéristiques d'un SEPDMI**

Dans cette section, nous présentons les principales propriétés qui caractérisent un SEPDMI.

### **2.1 Hétérogénéité des médias**

Les médias qui font partie d'un document multimédia ont des caractéristiques hétérogènes. Celles-ci peuvent être classées selon un certain nombre de critères dont les plus importants sont [57] :

- Le mode de perception de l'information : Il spécifie la manière selon laquelle l'utilisateur perçoit les médias. Ces derniers peuvent ainsi être audibles (comme un message audio ou un clip musical) ou visibles (comme un texte, une image ou une vidéo).
- Le comportement temporel : Il caractérise la façon selon laquelle les médias se comportent temporellement pendant leur présentation. On peut identifier les comportements suivants :
  - comportement discret, pour les médias n'ayant pas de notion de durée temporelle intrinsèque comme une image fixe ou un texte ;
  - comportement continu, pour les médias constitués d'un flot continu de données et caractérisés par une vitesse spécifique de restitution, comme une vidéo ou une audio ;
  - comportement indéterministe, pour les médias ayant des durées qui ne peuvent être déterminées qu'à la fin de leur présentation. C'est par exemple le cas d'un bouton d'interaction utilisateur, puisque l'on ne peut savoir à quel moment l'utilisateur va appuyer sur le bouton.
- Le format de codage : Différents formats de codage ont été définis pour stocker les médias. Ils servent fondamentalement à :
  - compresser le contenu d'un média afin de limiter l'espace de stockage utilisé, comme les formats MPEG [58], AVI et QuickTime pour la vidéo, les formats AU, AIFF, MPEG et WAV pour l'audio, et les formats GIF, JPEG [109], PNG et Bitmap pour les images ;

- associer aux médias un ensemble d'attributs qui spécifient le style de présentation pour ces médias. Par exemple les formats WORD, LATEX et Postscript permettent de spécifier entre autres la couleur, la police et la taille de caractères d'un média texte.
- L'adaptabilité : Ce critère définit la capacité des médias à répondre aux exigences de la qualité du service rendu. Hormis le texte, les médias tels que la vidéo, l'image et le son peuvent être adaptables (selon le codage utilisé) puisqu'ils tolèrent des dégradations dans leur qualité de présentation en fonction des ressources système disponibles.

## 2.2 Composition multimédia

Une présentation d'un document multimédia n'est pas une simple présentation de médias indépendants. C'est un processus qui traite un ensemble de médias organisés logiquement, temporellement et spatialement.

La spécification de l'organisation logique correspond à la décomposition d'un système d'information en composants [56]. Ainsi, l'auteur détermine le sujet à présenter et commence à définir ses médias, la façon selon laquelle ils seront synchronisés et leurs emplacements respectifs. L'opération de décomposition est récursive, c'est-à-dire que chaque composant peut être également décomposé en d'autres composants. L'opération de décomposition se termine lorsque le sujet est décomposé en médias de base (comme un texte, une image, une vidéo, une audio, etc.). Le résultat de cette opération donne une structure logique hiérarchique pour la présentation. Par exemple, un sujet comme la présentation d'un centre de recherche peut être décomposé en trois composants : une présentation générale, la recherche et développement et les services (voir Figure 1). La présentation générale peut être décomposée en un titre "*Présentation Générale*" et un "*message de bienvenue*", lui-même composé d'un message audio du directeur "*message du directeur*" et de sa photo "*Photo du directeur*" qui se présentent en parallèle pendant 20 secondes. Juste après le message de bienvenue, une vidéo "*Historique*" d'une durée de 60 secondes commence. À la fin de cette vidéo, un texte "*Missions centre*" est affiché pendant 30 secondes. Le titre "*Présentation Générale*" se termine au même temps que le texte "*Missions centre*". La partie recherche et développement commence 5 secondes après la "*Présentation Générale*", les divisions de recherche sont présentées en séquence, chacune pendant 600 secondes. La présentation d'une division commence par l'affichage du titre "*Présentation de la division*" et du texte "*missions*" d'une durée de 50 secondes. Juste après, le texte "*Projets*" est affiché. 10 secondes après le début de ce dernier, un commentaire audio "*Présentation des projets*" d'une durée de 60 secondes est joué. Il se termine au même temps que le texte "*Liste Projets*". Juste après, le texte "*Produits*" est affiché. 10 secondes après, une vidéo "*Démos des produits*" d'une durée de 320 secondes et qui se termine au même temps que le texte "*Produits*" est jouée. À la fin, les équipes de la division sont présentées en séquence. Chacune est composée d'un texte "*intitulé*", d'une photo des éléments de l'équipe et de leurs CVs qui s'affichent en parallèle pendant 30 secondes. La présentation de la division se termine par le texte "*Publications*" qui s'affiche pendant 30 secondes et qui se termine au même temps que le titre "*Présentation de la division*". Juste après, le composant "*Prestations de services*" est présenté pendant 300 secondes.

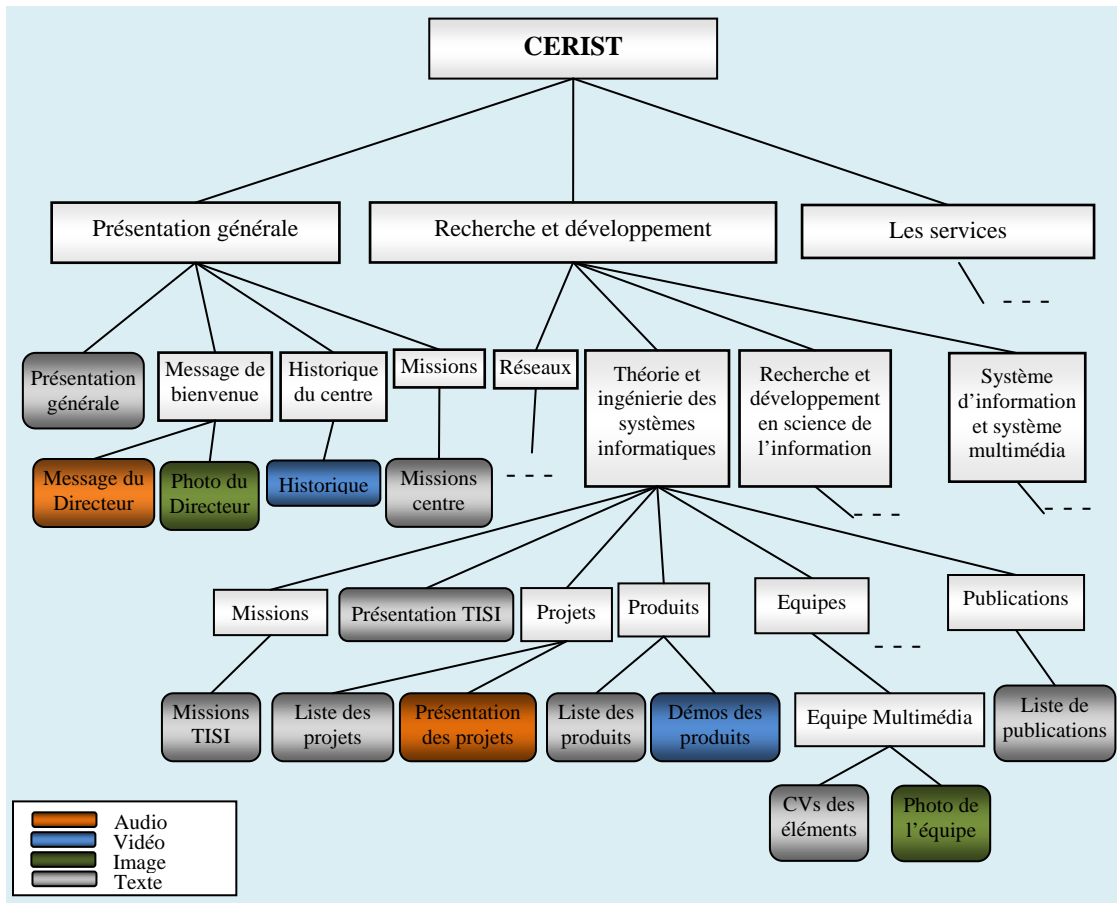


Fig 1. Structure logique d'une présentation

Du fait de la nature temporelle des documents multimédia, un besoin de perception globale du scénario apparaît tant pour l'auteur que pour le lecteur. Ce besoin peut être couvert soit par des techniques de visualisation de l'axe du temps (illustré dans la Figure 2) qui donne l'ordonnancement temporel des médias les uns par rapport aux autres (comme dans CMIFed [18], Macromedia Director [61] et Madeus [57]), soit par des fonctions de navigation en activant des hyperliens lors de l'exécution.

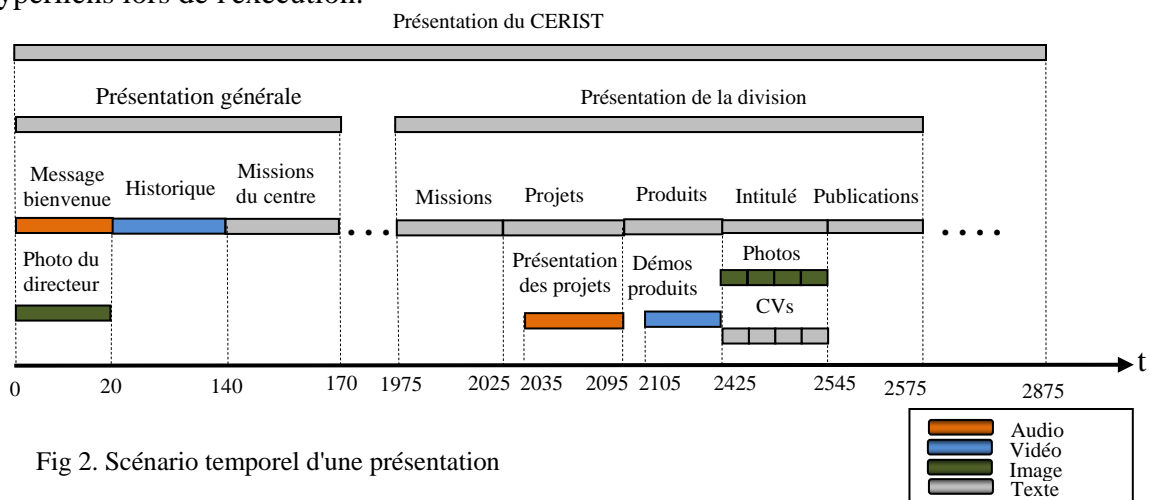


Fig 2. Scénario temporel d'une présentation

## 2.3 Intégration de données multimédia hétérogènes

Dans un SEPDMI, il est primordial de rendre homogène la manipulation des médias et les interactions utilisateur au sein d'une même structure de document. Cette intégration pose un certain nombre des problèmes liés principalement aux natures très diverses de ces médias (temporisés, statiques, indéterministes).

## 2.4 Synchronisation multimédia

La synchronisation est une caractéristique importante des SEPDMI qui concrétise la sémantique d'une présentation multimédia conçue par un auteur. Dans le cas des médias continus, le facteur temps apparaît comme une dimension essentielle de l'information. Une synchronisation peut être définie entre :

- des composants d'un même média. Par exemple, les images d'une vidéo ou des échantillons d'une audio qui doivent être présentés séquentiellement sont séparés par une durée prédéterminée ;
- des médias différents. Ils peuvent être :
  - soit indépendants. Par exemple, un texte s'affiche en même temps qu'une vidéo commence,
  - soit dépendants. Par exemple, une vidéo en parallèle avec son commentaire audio sont dépendants parce qu'ils ont été capturés simultanément. Une synchronisation interne aux deux médias doit alors avoir lieu pour assurer une cohérence entre l'état d'avancement de la vidéo et de son commentaire audio ;
- l'utilisateur et l'application multimédia. Par exemple une demande effectuée par un utilisateur pour démarrer, faire une pause, reprendre ou arrêter une présentation multimédia.

Dans la suite de cette section, nous détaillons les formes de synchronisation identifiées : la synchronisation intra-média, la synchronisation inter-médias et celle avec l'environnement.

### 2.4.1 Synchronisation intra-média

Ce type de synchronisation est inhérent au média continu qui est considéré comme une suite ordonnée d'unités de présentation ayant des relations temporelles implicites entre elles [12]. Pour ce type de média, les informations de synchronisation sont stockées dans le média lui-même lors de sa capture. Par exemple la vitesse de présentation d'une séquence vidéo (25 images/seconde) ou d'une séquence audio (8000 ou 44100 échantillons/seconde) spécifie la durée de présentation de chaque composant dans cette séquence. Ainsi, une image vidéo est présentée pour une durée de 40 millisecondes (voir Figure 3 ). Néanmoins, l'auteur d'un document multimédia peut spécifier une vitesse de présentation autre que la vitesse normale pour jouer le média plus rapidement ou plus lentement.

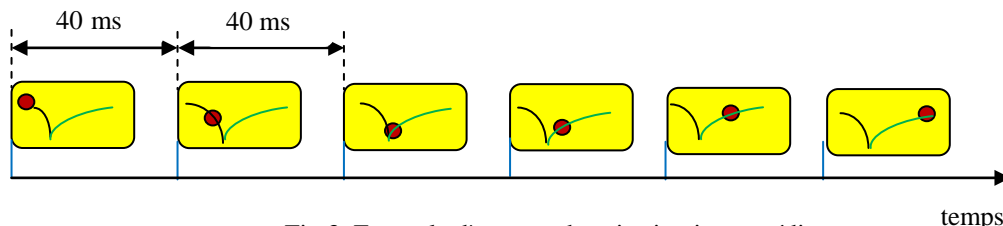


Fig 3. Exemple d'une synchronisation intra-média

## 2.4.2 Synchronisation inter-médias

La synchronisation inter-médias peut être *synthétique* ou *naturelle* [57]. La *synchronisation synthétique* est celle spécifiée explicitement par l'auteur entre différents médias afin de décrire un scénario souhaité. Elle est définie sous forme de relations temporelles entre les instants de début et/ou les instants de fin de différents médias (voir Figure 4). Ce type de synchronisation est appelé la *synchronisation à gros grain*.



Fig 4. Synchronisation inter-médias

Par contre, la *synchronisation naturelle* inter-médias est celle qui synchronise deux types de médias, souvent une vidéo avec une audio, où les paramètres de synchronisation sont déterminés lors de la capture des médias. Par exemple, si nous enregistrons une vidéo et son commentaire audio avec des vitesses respectives de 25 images/seconde et 8000 échantillons/seconde, une synchronisation naturelle est alors établie dont le paramètre est d'afficher une image vidéo tous les 320 échantillons audio. Ce type de synchronisation est appelé la *synchronisation fine* ou la *synchronisation de lèvres (lip-sync)*.

La synchronisation temporelle constitue un axe de recherche très important dans le domaine du multimédia. Dans le domaine de l'intelligence artificielle, Allen [1] a proposé un ensemble d'opérateurs qualitatifs binaires qui peut exprimer l'ordonnancement temporel entre des tâches dans une application de planification. Plusieurs adaptations de cet ensemble d'opérateurs ont été faites afin d'exprimer la synchronisation temporelle multimédia, comme dans OCPN [59] et Madeus [57]. Par ailleurs, Schloss et al. [89] ont proposé une algèbre d'événements pour construire des structures temporelles spécifiant la synchronisation entre les médias. Dans un but de standardisation, le *World Wide Web Consortium (W3C)* a proposé une spécification d'un standard nommé SMIL (*Synchronized Multimedia Integration Language*) [106] qui permet de spécifier la synchronisation sous la forme d'un arbre d'opérateurs.

Pour conclure, on peut classer les approches de spécification de la synchronisation en trois grandes familles :

1. les approches opérationnelles à base de programmation événementielle (comme Lingo [61] ou MHEG [74]),
2. les approches opérationnelles à base de structures d'arbres ou de graphes (comme CMIFed [101], HTSPN [91] ou SMIL [94][95]),
3. et les approches à base de contraintes d'intervalles ou d'instantes (comme FireFly [17], ISIS [96] ou Madeus [56]).

La comparaison entre ces approches peut être faite selon trois critères : la puissance expressive qui caractérise la possibilité de spécifier des schémas de synchronisation plus ou moins complexes, la facilité de spécifier un scénario temporel d'un document multimédia et la facilité de le modifier.

### **2.4.3 Synchronisation avec l'environnement**

Ce type de synchronisation permet à une application multimédia d'effectuer des actions de présentation en réponse à l'arrivée d'un événement venant de l'extérieur. Par exemples, les interactions utilisateur pour contrôler le déroulement de la présentation (comme démarrer, stopper, retour-arrière, etc.) ou les fonctions de navigation pour effectuer des sauts temporels dans le document.

## **2.5 Emplacement spatial des médias**

Dans un document statique aucune information ne spécifie l'ordre dans lequel doivent apparaître les éléments et c'est le plus souvent l'ordre de déclaration des éléments qui induit l'ordre spatial, c'est-à-dire la présentation du document. De plus, la mise en page est généralement spécifiée séparément et se limite aux informations typographiques.

Le choix d'un emplacement spatial, au même titre que l'information temporelle, est un élément déterminant dans la sémantique d'un document multimédia. Deux familles de modèles spatiaux existent dans la littérature : les modèles absolus où l'emplacement est spécifié avec des coordonnées et les modèles relatifs où il est spécifié avec des relations spatiales entre les médias. C'est derniers sont les plus utilisés dans les documents multimédia, et sont classés en approche par intervalles où les positions relatives sont exprimées à l'aide des relations d'intervalles d'Allen, et les approches par régions où la position absolue d'un média est donnée par la position de son repère et sa position relative est donnée par des relations spatiales. Les modèles les plus connus sont ceux proposés dans [30][80][81][86].

La comparaison entre ces approches peut être faite selon trois critères : la puissance d'expressivité qui caractérise la possibilité de spécifier des emplacements plus ou moins complexes, la précision de l'emplacement et la facilité de spécifier et de modifier un emplacement.

## **2.6 Recouvrement spatial des médias**

La qualité de service d'une présentation multimédia dépend en grande partie de la gestion du recouvrement spatial des médias qui le composent. Contrairement aux documents statiques où les parties de l'écran sont occupées par les mêmes médias pendant toute la durée de la présentation, dans un document multimédia ceux-ci peuvent se partager un même espace pendant leurs temps de présentation. Nous parlons de recouvrement spatial, lorsque deux ou plusieurs médias d'un document occupent, à un même instant de la présentation, entièrement ou partiellement, le même espace de présentation. Il en résulte alors une altération de la qualité de service de la présentation. Dans les systèmes actuels, cette gestion est laissée à la charge de l'auteur qui se retrouve contraint, à chaque fois qu'il procède à une opération d'édition, de vérifier la présentation de son document pour voir s'il n'y a pas de recouvrements et de revenir à la phase d'édition pour pouvoir l'éliminer. Un système multimédia doit assurer une gestion automatisée du recouvrement spatial des médias.

## 2.7 Indéterminisme de la présentation multimédia

L'indéterminisme d'une présentation multimédia est le décalage temporel entre la présentation et le scénario spécifié par l'auteur. Cette divergence peut avoir plusieurs sources, la plus importante étant la présentation d'un ou plusieurs médias incontrôlables. Un média est incontrôlable dans deux cas : soit c'est un média indéterministe comme un bouton d'interaction pour lequel nous ne savons pas à priori à quel instant l'utilisateur va l'activer, soit c'est un média continu ou discret pour lequel la durée de présentation diffère de celle attendue en raison des délais subis par le média pendant son accès distant et/ou pendant son traitement [57].

Au niveau de la synchronisation spécifiée par le scénario, l'indéterminisme introduit donc des décalages et, par conséquent, une désynchronisation entre les médias. Celle-ci peut dégrader la perception humaine du scénario. Des études expérimentales [98] ont été effectuées sur les seuils de décalage acceptables par la perception humaine. Le tableau de la Figure 5 montre les valeurs des seuils de décalage entre les différents types de médias. Ces valeurs peuvent être utilisées pour mesurer la qualité de service offerte par le système de présentation.

Média	Mode de couplage	Décalage toléré
animation	corrélés	+/- 120 ms
audio	synchronisation des lèvres	+/- 80 ms
image	annotation superposée	+/- 240 ms
	annotation non superposée	+/- 500 ms
texte	annotation superposée	+/-240 ms
	annotation non superposée	+/- 500 ms
audio	fortement couplés (stéréo)	+/- 11 ms
	faiblement couplés (plusieurs orateurs)	+/- 120 ms
	très faiblement couplés (musique de fond)	+/- 500 ms
image	fortement couplés	+/- 5 ms
	faiblement couplés (audio avec diapositives)	+/- 500 ms
texte	annotations textuelles	+/- 240 ms
curseur graphique	commentaire associé à un élément désigné	+/- 500 ms

Fig 5. Seuils de décalages entre les différents types de médias

Afin de ne pas dépasser les seuils de décalage, le système de présentation doit appliquer des mécanismes pour diminuer la désynchronisation introduite par les médias indéterministes.

## 3. Besoins de l'auteur et du lecteur

Afin de mieux cerner les fonctions que doit offrir un SEPDMI, une étude des besoins de l'auteur, élargie à ceux du lecteur, a été réalisée.

### 3.1 Besoins de l'auteur

Le processus de construction d'une présentation multimédia devient de plus en plus complexe notamment avec l'émergence de nouvelles applications comme le téléenseignement, la télémédecine, le tourisme ou la valorisation du patrimoine. Le système doit aider son utilisateur à surmonter cette complexité sans pour autant réduire de sa puissance. Plus précisément, nous identifions les besoins suivants :

1. La spécification de la synchronisation temporelle et de l'emplacement spatial doit être faite sans aucun préalable. Imposer, par exemple, à un auteur la connaissance d'un langage de programmation limite considérablement l'accès à l'écriture de ce type de documents.
2. L'approche de spécification doit permettre un passage simple et intuitif de la représentation mentale que se fait l'auteur du document à un véritable scénario formalisé. L'auteur qui conçoit un document s'en fait sa propre représentation mentale qu'il peut exprimer sous forme d'une description informelle. On retrouve dans les descriptions informelles principalement trois types d'information : absolu, par exemple *l'image X sera présentée 20 secondes après le début de la vidéo Y*, relatif *le texte X sera affiché après la vidéo Z* et imprécis *l'image X doit durer environ 1 minute*. Le codage de cette représentation dans les termes du système doit être le plus direct possible et accepter ces trois types d'information sans en imposer une à l'auteur [57].
3. L'approche de spécification doit garantir une puissance d'expressivité des relations temporelles et spatiales : l'auteur doit pouvoir spécifier, sans restriction, les scénarios qu'il désire.
4. Le système doit offrir un processus d'édition incrémental permettant un enrichissement progressif de son contenu et de sa structure. Pour cela, l'auteur doit pouvoir :
  - Effectuer des modifications locales sans avoir à reconsidérer globalement le document. Par exemple, l'auteur doit pouvoir modifier la durée d'un média sans pour autant mettre en cause toutes les synchronisations temporelles qu'il a du auparavant spécifier.
  - Passer aisément du mode édition vers le mode présentation (et vice-versa) pour, d'une part, aller voir le résultat de ces modifications et d'autre part pour retrouver facilement les informations qu'il souhaite modifier.
5. Le système doit assurer le maintien de l'intégrité du document : pour chaque opération d'édition effectuée par l'auteur, le système doit vérifier la cohérence des informations qu'il a introduites par rapport à l'état courant du document.
6. Le système doit gérer le recouvrement spatial des médias et l'indéterminisme garantissant ainsi une qualité de service à la présentation.
7. Le média ne doit plus être perçu comme une entité indissociable. L'auteur doit pouvoir utiliser une partie d'un média sans pour autant la reproduire physiquement, par exemple, spécifier une relation ou un lien avec une partie (séquence) d'une vidéo.
8. Un document devrait être présenté différemment selon le contexte de lecture pour s'adapter par exemples aux profils des lecteurs et à leur disponibilité (présentation d'un résumé par exemple). Le système doit permettre la définition de plusieurs parcours de lecture à partir d'un document primaire. Afin de limiter les coûts de création, ces différents parcours de ne doivent pas être conçus comme autant de documents car ils partagent les mêmes données.

### 3.2 Besoins du lecteur

Dans les systèmes actuels, le lecteur est considéré comme un acteur passif, il *subit* la présentation du document conçu par l'auteur. Permettre au lecteur, dans un respect strict de l'information contenue dans le document de l'auteur, de définir des parcours de lecture qu'il juge plus adaptés à ses besoins contribuera à mieux exploiter les documents.

## 4. Fonctions des systèmes multimédia

À partir des caractéristiques et des besoins identifiés ci-dessus, il en ressort qu'un SEPDMI comporte principalement deux types de fonctions : des fonctions d'édition et des fonctions de présentation.

Les fonctions d'édition, réservées à l'auteur, et dans une moindre mesure au lecteur, peuvent être résumées dans ce qui suit :

- Création, modification, suppression et sauvegarde de documents/parcours de lecture.
- Création, insertion, modification, suppression et sauvegarde d'un média.
- Ajout, modification et suppression des informations temporelles et spatiales.
- Gestion de la synchronisation temporelle et de l'emplacement spatiale.
- Gestion de l'indéterminisme.
- Gestion du recouvrement spatial.
- Création, modification, sauvegarde et suppression de liens hypermédia.
- Importation et exportation de documents/parcours de lecture.

Les fonctions de présentation, quant à elles, permettent de restituer au lecteur le contenu du document/parcours de lecture une fois que son édition est achevée. Cette phase consiste à fournir au lecteur un ensemble de commandes permettant d'explorer le document/parcours de lecture pour découvrir l'information qu'il contient. Les commandes fournies au lecteur lui permettent l'arrêt d'une présentation, son redémarrage, son avance et retour rapides ainsi que la possibilité d'interagir avec les médias du document/parcours de lecture.

## 5. Étude des systèmes multimédia

Parmi les fonctions présentées dans la section précédente, un certain nombre ont fait l'objet de travaux, notamment celles se rapportant à la spécification et la validation de la dimension temporelle puisque c'était la caractéristique nouvelle à prendre en compte. Dans les sections qui suivent, nous présentons brièvement quelques standards et systèmes ainsi que leurs limites dues essentiellement à la difficulté de leur utilisation ou au manque de réalisations pour les valider.

Afin de simplifier l'étude, les systèmes ont été classés dans deux types d'approches [51] :

1. Les approches opérationnelles ou impératives : trois familles ont été identifiées, celles qui s'appuient sur un axe temporel absolu, sur les langages de programmation et sur les représentations par structures de graphes.
2. Les approches à base de contraintes.

### 5.1 Les approches opérationnelles

#### 5.1.1 Axe temporel absolu

Les systèmes fondés sur les axes de temps absolu, ou *timeline*, sont les plus répandus dans les systèmes commerciaux comme Flash et Director de Macromedia [61] ou MAestro [26]. De même, le standard HyTime [41] est un langage qui s'appuie également sur la métaphore de la ligne de temps.

### - Environnements auteur fondés sur les timelines

Ces systèmes s'appuient sur une métaphore graphique d'un axe pour représenter la dimension temporelle d'un document multimédia. Les interfaces à base de timelines représentent les activités de présentation concurrentes au moyen de plusieurs axes parallèles dont chacun est réservé à un média particulier. Un axe commun, partagé par tous les axes, est utilisé comme référence pour représenter le temps physique : l'unité est exprimée en nombre de trames (unité logique) à laquelle on peut attacher une durée. Nous allons, dans ce qui suit présenté Hytime, le standard le plus connu dans cette famille.

### - HyTime

HyTime est un langage standard proposé par l'ISO pour la structuration des documents multimédia et hypermédia, la représentation des liens hypertexte, et la spécification de l'ordonnancement des événements spatiaux et temporels ainsi que de leur synchronisation [31] [25]. Il constitue une application de SGML [34] qui est un standard de représentation de documents structurés statiques.

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

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

Il existe un nombre très faible de réalisations et d'outils pour le standard HyTime, probablement à cause de sa complexité et de la difficulté que les développeurs rencontrent pour le lire et le comprendre [41]. Néanmoins, quelques travaux ont été effectués à partir de HyTime comme par exemple les spécifications SMDL [92] (*Standard Music Description Language*) qui est un langage conforme au standard HyTime pour représenter les informations musicales accompagnées par d'autres informations textuelles et graphiques. L'un des principaux avantages du standard HyTime est le fait qu'il propose une modélisation générale d'un document hypermédia capable de supporter la représentation d'un grand nombre de configurations de documents. Du point de vue de l'expression des contraintes temporelles, il permet de modéliser tous les scénarios de présentation possibles si les médias ont des comportements temporels déterministes. En revanche, il présente un ensemble d'insuffisances que nous résumons dans ce qui suit :

- Complexité d'utilisation, ce qui limite son utilisation à des profils bien déterminés alors qu'il s'agit d'un éditeur de documents.
- Une absence de flexibilité de la synchronisation temporelle et de l'emplacement spatial : le début et l'emplacement d'un média sont données à une date et à des coordonnées précises.

- Il ne supporte pas les médias indéterministes à cause de l'utilisation des axes de temps absolus.
- Toute modification d'un FCS peut entraîner des modifications en cascade en raison de l'absence de dépendances relatives entre les médias.

D'une façon générale, les systèmes à base de timelines répondent bien aux deux premiers besoins pour les auteurs identifiés en section 3 : en effet, l'interface utilisateur est accessible sans préalable et rend compte de façon intuitive du placement temporel des différents événements de présentation. Par contre, ces inconvénients sont multiples et liés aux effets de bord que cette même interface engendre dans les différents traitements d'édition, à savoir :

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

### 5.1.2 Langages de scripts

Les outils à base de timelines, du fait de leur pouvoir d'expression limité, ont été le plus souvent complétés par des langages de programmation (scripts). C'est le cas de Director dont le langage de script associé s'appelle Lingo [62]. L'environnement IconAuthor [42], permet de spécifier également un flot d'exécution au travers d'une interface à base d'icônes. De même, le standard MHEG [74][79] s'appuie sur le paradigme des langages de programmation orientés objets.

Il est clair qu'avec un langage de programmation, il est possible d'exprimer des scénarios aussi complexes que l'on veut : le scénario est spécifié sous une forme opératoire par un ensemble d'instructions « événement & conditions → actions ». Dans ce qui suit nous allons présenter les standards les plus connus.

## - MHEG

Le but du groupe MHEG [74][79] (*Multimedia and Hypermedia Information Coding Experts Group*) est la définition et la standardisation d'un format portable de représentation multimédia pour faciliter l'échange de médias entre sites. Il faut noter que MHEG, par opposition à HyTime, gère la forme finale de la présentation, c'est-à-dire qu'il garde seulement les informations sur les relations temporelles et spatiales mais il supprime les informations sur la structuration logique de la présentation. Ceci a abouti à un modèle constitué d'une hiérarchie de classes *média MHEG* comprenant les classes *Content* (qui permettent de décrire le contenu des médias), *Script* (pour définir des comportements sous forme de programmes), *Composite*, *Link* et *Macro*, ces trois dernières étant décrites ci-dessous.

Les *médias composites* sont utilisés comme *container* pour un groupe de *médias MHEG* appartenant à une partie d'une application multimédia, formant ainsi un média complexe. Ils permettent donc d'emballer les composants d'une application multimédia en un ensemble de médias composites pour faciliter l'échange de l'application entre sites. Le début de la présentation correspond à un média composite nommé le média racine. Les médias composites sont liés entre eux à travers des pointeurs. De plus, les médias composites permettent la synchronisation entre parties d'une application multimédia à l'aide de primitives de composition spatiale et temporelle. Deux primitives temporelles binaires : *serial* (qui définit une exécution en séquence entre des médias) et *parallel* (qui indique une exécution simultanée des médias) sont utilisées pour spécifier les scénarios multimédia.

Les *médias MHEG links* ne sont pas les liens hypermédia (hyperliens) conventionnels mais permettent de spécifier les relations spatiales, temporelles et conditionnelles entre les médias ainsi que les actions effectuées par ou sur ces médias. Un *MHEG link* consiste en une condition (*LinkCondition*) et un effet (*LinkEffect*) qui est une liste d'actions à exécuter sur un ou plusieurs médias cibles lorsque la *LinkCondition* est vérifiée. En ce sens, le modèle d'exécution de MHEG est fondé sur la programmation par événements. Un *MHEG link* peut être un composant d'un média composite ou bien un média totalement à part.

Les *médias Macro* constituent une autre façon de construire les médias complexes en utilisant des macros paramétrées. Les valeurs de paramètres sont définies par l'application qui utilise la macro correspondante.

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

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

Le standard MHEG se compose de plusieurs parties notamment les parties MHEG-5 et MHEG-6. La partie MHEG-5 est destinée à supporter la distribution des applications multimédia dans une architecture client/serveur à travers différents types de plates-formes. MHEG-5 est spécifiée afin de permettre le développement d'un interpréteur du format d'échange MHEG, qui requiert très peu de ressources. Les applications MHEG résident sur un serveur et le client charge les portions d'application dont il a besoin. L'interpréteur réside sur le client qui s'occupe d'interpréter le format d'échange MHEG, de présenter l'application à l'utilisateur et de traiter les interactions de l'utilisateur.

La partie MHEG-6 est destinée à apporter la portabilité à MHEG-5 en ajoutant des fonctions de traitement de données, des fonctions de communication entre les serveurs et les clients et des fonctions d'interface avec les périphériques.

MHEG est à l'origine de plusieurs prototypes d'application multimédia. Le projet GLUE [9] de DeTeBerkom est une réalisation du système MHEG-5. Le projet MAJA [11], qui est un projet plus récent chez DeTeBerkom, consiste en une *Java Applet* dont le rôle est de présenter des applications MHEG en reliant, au moyen de la technologie de communication CORBA, un client de présentation MHEG à un moteur MHEG sur un site serveur.

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

### - PREMO

L'objectif de la norme PREMO (*PResentation Environments for Multimedia Objects*) développée par l'ISO [38] est de fournir un environnement de développement standard et portable pour les applications multimédia. PREMO s'intéresse essentiellement aux techniques de présentation des données multimédia selon une approche média issue du modèle développé par le consortium OMG (*Object Management Group*) et qui permet de spécifier les caractéristiques visibles des médias (*PREMO objects*) indépendamment de leur réalisation. Le modèle média PREMO est défini à partir de trois composants (hiérarchies de classes de médias) [43] : le composant *Foundation* pour la spécification abstraite des informations multimédia [44], le composant *Multimedia System Services* pour la spécification des traitements de bas niveau sur les données multimédia [45], et le composant *Modelling, Presentation, and Interaction* pour la présentation des médias en scènes [46]. Ainsi, le composant *Foundation* est une collection de médias réalisant des services utiles pour une large variété de composants. Ces médias sont :

- *PREMOObject*, qui définit le comportement de base des médias PREMO : initialisation, destruction et fourniture d'informations sur le type du média ;
- *SimplePREMOObject*, qui permet de spécifier les structures de données (sans méthode) qui seront utilisées par les autres médias PREMO ;
- *EnhancedPREMOObject*, qui sert de base pour des médias comme les gestionnaires d'événements, les horloges ainsi que les médias temporisés.
- *Callback*, utilisé pour la notification entre médias lors de leurs changements d'état ou lors de l'occurrence de certains événements.

Dans PREMO, la synchronisation est exprimée en termes de classes *parallel* et *sequential*. Celles-ci représentent des médias containers groupant un ensemble de médias avec les relations temporelles *par* et *seq*, respectivement. De plus, PREMO utilise un modèle d'événements qui réalise la synchronisation à base d'événements. Le modèle d'exécution spécifié par PREMO est donc fondé sur la programmation par événements.

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

Au niveau de la réalisation, il n'y a aucune application réelle basée sur PREMO.

D'une façon générale, le point faible de ces approches vient de leurs capacités limitées à répondre aux besoins des auteurs : les compétences en programmation requises pour les utiliser restreignent leur emploi aux seuls informaticiens ; il est de plus très difficile de percevoir l'enchaînement temporel des médias par la simple lecture d'un script car la composition temporelle est dispersée dans le script ; comme pour tout langage de programmation, la mise à jour d'un scénario est une tâche souvent délicate du fait des erreurs qu'il est facile d'y introduire ; la gestion du recouvrement spatial est laissée à la charge de l'auteur ; enfin le document ainsi construit délivre une présentation unique.

### 5.1.3 Structures de graphes

Dans les approches fondées sur les graphes, le document est représenté sous la forme du diagramme de son flot de contrôle. Ce diagramme décrit l'interaction entre les médias du document à l'exécution. L'édition d'un document peut ainsi s'appuyer sur une description plus riche et plus facile à utiliser de la synchronisation temporelle. Il est aussi possible de représenter beaucoup plus de combinaisons de scénarios que dans le cas des *timelines*. Les graphes fournissent également un support plus adapté pour les opérations d'édition comme la suppression ou l'insertion d'éléments car la disposition des éléments est maintenue sous forme explicite comme dans l'interface de Firefly [17]. Il existe deux variantes de ces diagrammes : les graphes plats et les graphes hiérarchisés.

#### - Graphes plats

Les graphes plats décrivent la synchronisation entre les instants de début et de fin de l'ensemble des éléments d'un document. Souvent, la synchronisation est décrite au moyen des trois opérateurs d'instant « *avant* », « *après* » et « *en même temps* ». Les graphes sont utilisés de deux façons : dans Firefly, leur utilisation se limite à un cadre informel de synchronisation motivé par des besoins d'interface graphique, la vérification étant réalisée par un algorithme de gestion de contraintes. Bien que le principe en soit simple, l'approche des graphes plats présente un ensemble d'insuffisances :

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

## - Graphes hiérarchisés

L'approche basée sur une structure de graphes hiérarchisés exploite l'organisation logique du document pour décrire et mettre en œuvre sa synchronisation temporelle. Le grand avantage de cette approche est la possibilité d'organiser le document en modules indépendants sur lesquels on peut appliquer des primitives globales de synchronisation. Ces primitives s'appliquent sur des intervalles et sont la mise en parallèle ou en séquence des médias appartenant à une même entité logique. Les inconvénients cités pour les graphes plats sont compensés dans les graphes hiérarchisés grâce à des mécanismes d'édition plus élaborés permettant d'encapsuler des parties de document et de les utiliser comme des médias de base. Nous allons, dans ce qui suit, présenter le langage SMIL, l'un des outils les plus représentatifs de ce type d'approche.

## - SMIL

SMIL (*Synchronized Multimedia Integration Language*) [93][106] est un langage déclaratif développé par le consortium World Wide Web dont le but est de spécifier des documents multimédia sur le Web et de les jouer par des navigateurs SMIL. Un tel navigateur peut être soit un système de présentation isolé soit intégré dans un navigateur Web.

SMIL est un format d'intégration, c'est-à-dire qu'il ne décrit pas le contenu des médias faisant partie d'une présentation multimédia, mais plutôt leur composition temporelle et spatiale ainsi que les hyperliens qui les lient.

SMIL 2.0 définit 10 groupements fonctionnels majeurs des éléments et des attributs (figure 6). Le module *Timing and synchronisation* est considéré comme le cœur des spécifications SMIL. L'objectif de cette organisation de SMIL 2.0 en groupes est de faciliter la réutilisabilité des fonctions SMIL en d'autre langage à base de XML.

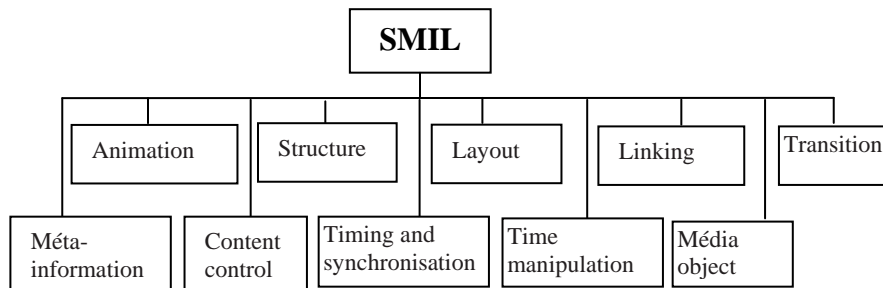


Fig 6. Groupage fonctionnel des modules de SMIL 2.0

L'organisation temporelle d'un document SMIL est basée sur une structure hiérarchique d'opérateurs de synchronisation (parallèles et séquentiels) et un ensemble d'attributs temporels associés aux médias et aux opérateurs. La sémantique associée à une telle structure est définie en appliquant deux ensembles de règles : le premier pour définir la durée du média et le second pour définir les comportements des opérateurs. L'attribut *Dur* indique la durée du média en valeur absolue. S'il n'est pas employé, la durée du média dépend de sa nature : un média continu (vidéo et audio) est joué jusqu'à la fin, un média discret (image, texte,...) est affiché indéfiniment. Cette valeur peut être modifiée par les règles liées à l'opérateur temporel (*Seq* ou *Par*) auquel est lié le média. Une présentation SMIL est une composition structurée de médias. Comme le montre la figure 7, trois opérateurs de base sont utilisés.

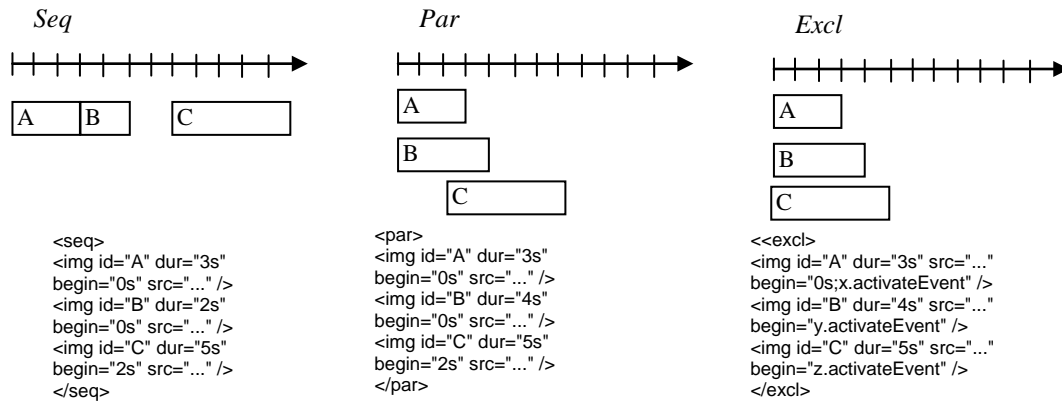


Fig 7. Les opérateurs temporels

L'opérateur séquentiel *Seq* exprime la présentation séquentielle des deux ou plusieurs médias. Les médias contenus à l'intérieur de l'opérateur *Seq* sont organisés d'une façon à ce qu'un média ne puisse commencer avant la fin de son prédécesseur.

L'opérateur parallèle *Par* exprime la présentation simultanée de deux ou plusieurs médias. Par défaut, la fin d'une construction parallèle est égale à la fin du média le plus long. Cependant, cette sémantique peut être changée en employant l'attribut *end\_sync* de telle manière que la fin de la construction soit égale à la fin du média le plus court (*end\_sync = first*) ou égale à la fin d'un média bien précis (*end\_sync = id (média)*). L'opérateur *Excl* exprime l'exclusion : seulement un média sera présenté parmi ceux contenus dans cet opérateur.

Cette structure hiérarchique permet de définir le comportement temporel de base. Pour plus de détail, l'auteur peut modifier le comportement par défaut de la structure hiérarchique en utilisant les attributs *begin* et *end* qui expriment des valeurs absolues ou relatives à la composition parallèle. Une valeur absolue pour l'attribut *begin* associée à un média dans un nœud parallèle (respectivement un nœud séquentiel) exprime le décalage du commencement du média par rapport au nœud parallèle (respectivement au nœud séquentiel). Si l'auteur emploie une valeur relative, il impose que le moment de début ou de fin d'un média soit identique au moment de début ou de fin d'un autre média du même nœud.

Pour ce qui est du positionnement spatial, celui-ci est réalisé de manière absolue. Dans l'entête du document, l'espace de présentation est divisé en régions rectangulaires appelées *channels* et chaque région est définie par sa position absolue, sa taille et une valeur indiquant l'ordre de recouvrement de ces régions. Dans le corps du document, à chaque média est affectée une région indiquant son espace de présentation. En plus des attributs de la région (*left*, *top*, *right*, *bottom*...), un attribut *fit* est introduit, si les attributs intrinsèques du média diffèrent de ceux de la région associée, indiquant la manière dont les médias sont présentés, il prend l'une des valeurs suivantes :

- *Hidden* : si le média est plus petit que la région alors le reste de la région est rempli par la couleur de fond, et s'il est plus grand alors la partie qui déborde ne sera pas incluse dans la présentation.
- *Fill* : il modifie la hauteur et la largeur du média indépendamment de sorte qu'il soit exactement de même dimensions que la région.
- *Meet* : cette valeur assure que les médias qui n'ont pas les mêmes dimensions que la région soient rétrécis ou agrandis afin qu'au moins une de leurs dimensions corresponde à au moins une dimension de la région.

- *Scroll* : un mécanisme de défilement sera mis en place si le média dépasse les limites de la région.

L'apport de SMIL 2.0 par rapport à SMIL 1.0 dans l'aspect spatial est la possibilité d'utiliser des sous régions, ce qui permet de placer un média dans une position (x, y) par rapport au coin supérieur gauche de la région.

SMIL 2.0 bénéficie également d'une dimension navigationnelle riche. À la différence des liens HTML, le concept fondamental du lien SMIL correspond à une recherche dans le scénario temporel de la présentation. Plutôt que d'activer simplement le média cible, son état sera identique à celui qu'il aurait été si la cible avait été atteint au fil de la présentation. Ceci signifie que tous les nœuds situés temporellement entre la source du lien et la destination doivent être évalués pour voir s'ils contribuent à l'état du média cible.

SMIL 1.0 ne gère pas l'indéterminisme et ceci est dû à son caractère absolu (utilisation de timelines). Pour pallier à ce problème, des solutions ont été proposées pour SMIL 2.0 parmi lesquelles nous citons les travaux de Yang et al [114].

La différence majeure dans la gestion temporelle entre SMIL 1.0 et SMIL 2.0 [93] est que les valeurs des attributs *begin* et *end* des opérateurs *Par* et *Seq* peuvent être associées à des événements indéterministes. Par exemple, la portion du script suivant indique que les images A et B seront présentées une fois que l'utilisateur aurait cliqué sur le bouton btn.

```
<seq begin="btn.click() >
<img id="A" dur="3s"/>

</seq>
```

L'idée des auteurs est de convertir le script SMIL vers la forme d'un DDTL (Dividable Dynamic Timeline) puis de reconverti ce dernier en un scripte SMIL 2.0 tout en gardant les spécifications de l'indéterminisme.

#### **- Conversion du script SMIL en forme DDTL [112][113]**

La conversion d'un script SMIL à la forme DDTL passe par deux étapes :

Conversion du script SMIL en modèle E-RTSM.

Extraction de la forme DDTL à partir du modèle E-RTSM : cette phase passe par deux étapes : Identifier les "splitting events" et Identifier les sections dynamiques.

#### **- Conversion de la forme DDTL vers SMIL 2.0**

Le résultat final du DDTL est converti en script SMIL 2.0. La conversion est effectuée sur le "dividable Timeline" et la "section dynamique DS".

#### **- Dividable Timeline**

Les deux parties du Timeline résultantes de l'application de l'opération TL-Divide seront présentées en séquence ; un opérateur *Seq* est défini entre ces deux parties et un opérateur *Par* avec sont attribut *<end>* égal à l'événement indéterministe est ajouté entre les deux parties, comme le montre le script ci-dessous.

```

<seq>
<TL1>
<par end="Btn.Click"> </par>
<TL2>
</seq>

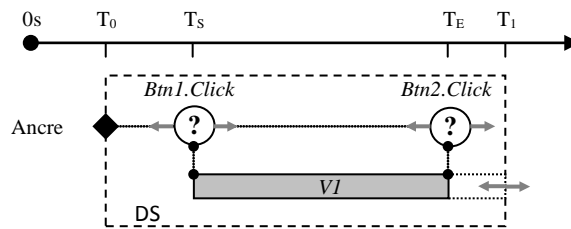
```

### - Section dynamique

Il existe deux cas pour la fin d'une Section Dynamique (DS), comme illustrer dans la figure 8:  
 Soit l'événement de fin du média correspondant à la section dynamique est déclenché.

Soit la date limite de la DS est atteinte.

Par conséquent, pour chaque DS, un opérateur *Par* est créé avec son l'attribut *<fin>* initialisé de manière à refléter les deux cas précédents.



```

<par begin="T0"end="Btn2.Click; T1">
  <video id=VI begin="Btn1.Click">
</par>

```

Fig 8. Cas de fin d'une DS

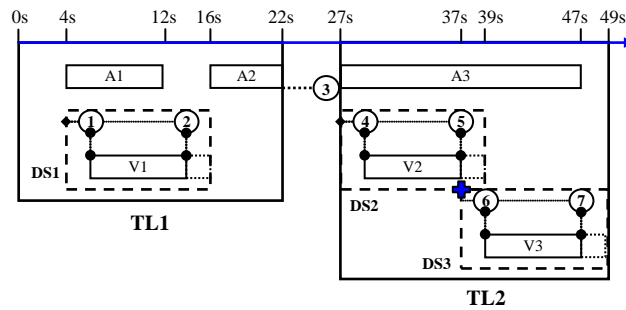
Par analogie à l'exemple précédent, une DS contenant plusieurs médias est convertie comme suit :

```

<par begin="T0"end="Btn2.Click; T1">
  <par begin="Btn1.Click">
    <video id=VI>
    <audio id=A1>
    <text id=Tx1>
  </par>
</par>

```

Pour une séquence de Sections Dynamiques, un opérateur *Seq* est créé pour toutes les DS de la séquence comme illustré dans la figure 9.



```

<seq>
  <par id=TL1>
    <seq>
      <audio id=A1 begin="4s" dur="12s-4s">
        <audio id=A2 begin="16s-12s" dur="22s-16s">
      </seq>
      <par id=DS1 begin="4s", end="Btm2.Click;16s">
        <video id=V1 begin="Btm1.Click">
      </par>
    </par>
  <par end="Btm3.Click"> </par>
  <par id=TL2>
    <audio id=A3 begin="0s" dur="47s-27s">
      <seq>
        <par id=DS2 begin="0s" end="Btm5.Click;39s-27s">
          <video id=V2 begin="Btm4.Click">
        </par>
        <par id=DS3 end="Btm7.Click;49s-37s">
          <video id=V3 begin="Btm6.Click">
        </par>
      </seq>
    </par>
  </seq>
</seq>

```

Fig 9. Exemple de conversion d'une séquence dynamique

Hormis la complexité de spécification d'un document et le fait qu'il s'agisse d'un langage de spécification (la présentation du document se fait sous un autre environnement), nous avons relevé les remarques suivantes :

- La puissance d'expressivité de la synchronisation et de l'emplacement spatial peut être insuffisante et peu précise pour des scénarios complexes.
- Le concept de régions retenu pour le positionnement spatial, en plus de la difficulté des mises à jour due à des spécifications absolues, peut conduire à une mauvaise qualité de service de la présentation lorsque des médias seront soit réduits ou agrandis en fonction de la taille de la région/sous région à laquelle ils appartiennent.
- La gestion du recouvrement spatial est laissée à la charge de l'auteur. Aucun mécanisme automatisé n'a été prévu.
- L'approche proposée pour la gestion de l'indéterminisme est difficilement applicable car l'auteur doit spécifier, au préalable, l'ancre et la longueur de la DS correspondants aux médias incontrôlables alors que ces informations dépendent du comportement imprévisible du lecteur.
- Enfin, le document ainsi construit délivre une présentation unique.

## 5.2 Les approches à base de contraintes

L'un des mécanismes d'abstraction de haut niveau utilisé dans le contexte des scénarios multimédia est sans doute la traduction en forme de contraintes linéaires les informations servant à construire un scénario et l'utilisation de la programmation logique par contraintes pour manipuler ces dernières [49].

Dans cette famille d'approches, un scénario est quasiment défini par un ensemble de relations temporelles et spatiales qui seront converties en une liste de contraintes. À partir de cette liste, les systèmes basés sur ces approches ont pour tâche d'une part de s'assurer de la cohérence temporelle et spatiale du document : l'existence d'au moins une solution qui respecte toutes les contraintes; d'autre part de produire statiquement et/ou dynamiquement une (des) solution(s) d'*emplacement* temporel et spatial.

L'avantage est que l'auteur spécifie ce qu'il souhaite obtenir comme *emplacement* temporel et spatial sans avoir à spécifier toutes les informations attachées à ces médias (instants de début et de fin, durée, coordonnées, etc.). La modification d'un document est rendue plus simple. En effet, une modification locale comme le changement de la durée d'un média, l'ajout ou le retrait d'une relation ou d'un média sera automatiquement répercutée sur le reste du document, et ce en conservant la synchronisation temporelle et les emplacements spatiaux auparavant spécifiés. Étant donnée la nature très itérative du processus de création d'un document, cette facilité avec laquelle l'auteur peut modifier son document constitue un avantage certain. Néanmoins, l'utilisation de ces techniques se heurte essentiellement à la difficulté de mettre au point un résolveur à ces contraintes assez puissant et rapide.

Il existe actuellement plusieurs systèmes basés sur les contraintes, on peut citer Isis [52], Tiempo [110] et Madeus [57]. Nous allons développer dans la suite de cette section l'environnement Madeus qui reste un des plus représentatifs de cette approche.

### - Madeus

Un document Madeus [49][50] possède une organisation logique décrite sous forme d'une hiérarchie de composants, les feuilles étant des médias de base (texte, image, vidéo, son). À chaque niveau de la hiérarchie, l'auteur peut spécifier des relations temporelles entre tous les médias (de base ou composites) qui partagent le même père. Ceci constitue le scénario temporel du document. Les relations dont dispose l'auteur sont données par les opérateurs de l'algèbre d'Allen, augmentées de l'opérateur *parmin* qui permet d'exprimer des comportements d'interruption. Pour aider l'auteur dans sa tâche de spécification du scénario temporel, Madeus permet de visualiser le scénario à travers une *vue scénario*. Cette vue est inspirée du timeline classique utilisé dans les approches temps absolu, mais elle est adaptée ici à l'utilisation des contraintes :

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

L'environnement d'édition d'un document dans Madeus est fondé sur une extension du principe "wysiwyg" utilisé par la majorité des éditeurs de texte. Celui-ci ne peut s'appliquer tel quel

puisque un document multimédia est par nature dynamique et qu'il n'est du coup plus possible de fusionner la phase d'édition et la phase de présentation. Cependant, comme il est important pour l'auteur de limiter autant que possible la distance entre ces deux phases, Madeus propose à l'auteur d'utiliser la fenêtre de présentation de son document comme une vue sur laquelle il est possible, après avoir stoppé la présentation du document, de sélectionner des médias pour leur associer une relation temporelle (ou spatiale). Cependant, cette approche n'est pas toujours satisfaisante et d'autres supports d'édition doivent être proposés à l'auteur.

La représentation graphique des éléments dans Madeus s'inspire de la notion de boîtes. Le formatage spatial se limite à un emplacement absolu des éléments de base sur l'écran. Pour cela, l'auteur renseigne, pour chaque élément de base, l'attribut de positionnement qui spécifie l'emplacement du coin supérieur gauche de l'élément.

La version actuelle de l'emplacement spatial dans Madeus souffre de plusieurs limitations dont la plus importante est qu'elle ne permet pas d'exprimer un emplacement relatif entre les éléments. Par exemple, l'auteur peut vouloir spécifier qu'une image est centrée par rapport à une légende ou que toutes les vidéos qui défilent dans une scène sont placées au même endroit. Pour cela, il doit positionner les éléments dans la configuration souhaitée et sauvegarder son document pour stocker les nouvelles valeurs des attributs de positionnement. Mais cette méthode ne spécifie pas les relations que l'auteur veut établir entre les éléments et de plus, ces derniers ne sont pas solidaires dans leurs déplacements. Toutefois, il est à signaler que d'autres travaux, notamment ceux de Carcone [20] ont tenté de combler ces insuffisances d'emplacement spatial en proposant une approche à base d'un ensemble de relations spatiales.

L'autre limitation dans la spécification des relations temporelles est qu'aucune relation temporelle ne peut être spécifiée entre un élément de base d'un composite et un autre élément de base d'un autre composite.

Les spécifications de relations temporelles cycliques ne sont pas permises et sont, dans le meilleur des cas, décomposées en spécifications non cycliques par des modules externes au résolveur choisi.

Enfin, le recouvrement spatial des médias n'est pas géré et le document ainsi construit ne délivre qu'une seule présentation.

## **6. Conclusion**

Dans ce chapitre, nous avons passé en revue les principaux aspects liés aux SEPDMI. Le premier constat que nous pouvons établir est que la quasi-totalité de ces systèmes se sont focalisés sur le problème de la dimension temporelle et ce au détriment de la dimension spatiale alors qu'on devrait lui accorder autant d'importance si on veut aboutir à la construction de systèmes performants dans le domaine.

De plus, la complexité de conception et de mise en œuvre qui caractérise ce type de systèmes recommandent de rechercher d'emblée un haut niveau d'intégration des quatre dimensions et ce afin de mutualiser au maximum leur conception et leur implémentation. Contrairement à ce qui a été proposé dans Madeus, par exemple, le traitement de la dimension spatiale, très limité dans sa première version, a connu plusieurs autres propositions basées sur des approches différentes de celle traitant la dimension temporelle.

Quant aux standards et systèmes, comme Hytime, MHEG, PRIMO et dans une moindre mesure SMIL, leur complexité de mise en œuvre et d'utilisation ont largement contribué à l'échec de leur utilisation, voire de leur validation, et ce malgré les performances qu'ils présentent dans la prise

en charge de la dimension temporelle. En effet, requérir, par exemple des connaissances en programmation pour l'utilisation d'un système d'édition et de présentation, censé être destiné à un large public, ne peut mener qu'à son rejet.

La gestion automatisée du recouvrement spatial n'a été considérée par aucun système existant, alors que ses conséquences peuvent affecter considérablement la qualité de service de la présentation.

Enfin, l'adaptation du contenu par rapport aux profils des lecteurs, qui n'est proposée par aucun des systèmes présentés, apparaît comme un nouveau besoins à prendre en charge.

En résumé, pour le développement de notre SEPDMI, nous adoptons les orientations principales suivantes :

- Définir un modèle de document adapté à la nature incrémentale de la phase d'édition, mais surtout assurant un haut niveau d'intégration des différentes dimensions du document et permettant de produire différentes lectures d'un même document.
- Assurer une simplicité d'utilisation alliée à une puissance d'expressivité du système. A ce niveau, nous allons adopter une approche à base de relations pour la spécification des informations temporelles et spatiales.
- Adapter une approche unique à base de contraintes, pour l'analyse et la synthèse des relations temporelle et spatial.
- Proposition d'une approche automatisée pour la gestion du recouvrement spatial à base de contraintes.

Dans le chapitre suivant nous allons présenter le modèle de document adopté.

# Chapitre 3

## Modèle de documents multimédia

### 1. Introduction

La conception d'un SEPDMI doit passer par la définition d'un modèle de document permettant de spécifier et de gérer les quatre dimensions qui le composent : la dimension logique qui décrit le regroupement des médias sous forme d'entités, la dimension temporelle qui décrit le scénario temporel du document, c'est-à-dire l'ordonnancement des médias dans le temps, la dimension spatiale qui décrit l'emplacement spatial des médias visibles sur le support de présentation et la dimension hypermédia qui décrit les liens de navigation des médias.

C'est en tenant compte des insuffisances constatées dans les modèles actuels, notamment l'impossibilité de définir des parcours de lecture adaptés aux profils, l'impossibilité de pouvoir utiliser une partie d'un média sans recourir à des traitements supplémentaires et l'impératif de considérer à la fois les quatre dimensions, que nous avons abordé cette partie.

Avant de présenter le modèle de document, nous allons commencer par présenter les concepts définis dans le modèle.

### 2. Concepts du modèle

#### 2.1 Le média

##### 2.1.1 Modélisation logique d'un média

Les systèmes actuels considèrent, tant dans leur phase d'édition que dans leur phase de présentation, un média dans sa globalité, c'est-à-dire comme une entité indissociable. L'auteur ne possède aucun moyen d'interagir avec une partie d'un média. Par exemple, de ne visualiser que la séquence se trouvant dans l'intervalle [30s, 60s] d'une vidéo ou de créer un lien vers une partie d'un média ou de composer un nouveau média à partir des médias existants.

Dans notre modèle de document, un média est perçu avec un niveau de granularité plus fin. À cet effet, nous introduisons le concept de média dérivé. Un média dérivé est défini comme une partie ou une concaténation de parties de médias physiques de même type. La figure 1 donne un exemple de définition d'un média dérivé et la figure 2 donne des exemples de leur utilisation dans des liens de navigation [70].

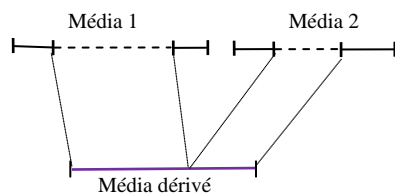


Fig 1. Définition d'un média dérivé

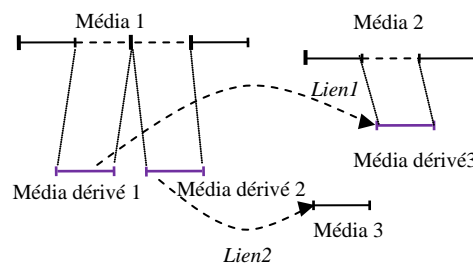


Fig 2. Liens avec un média dérivé

Formellement le média dérivé X est défini à partir d'un ou de plusieurs médias physiques  $X_1, \dots, X_j$  de même type :

$$X = \text{UNION}_{k \in [i,j]} X_k (\alpha, \beta, \mu) \text{ avec } \alpha \geq 0 \text{ et } \beta > \alpha.$$

- Pour la vidéo et l'audio,  $[\alpha, \beta]$  est un intervalle temporel,  $\beta \leq \text{Durée}(X_k)$  et  $\mu = 0$ .
- Pour le texte,  $\alpha$  et  $\beta$  correspondent, respectivement, aux numéros de lignes et de colonnes du premier et du dernier caractère du texte sélectionné et  $\mu = 0$ .
- Pour l'image :  $\alpha$  et  $\beta$  correspondent, respectivement, aux coordonnées de l'angle supérieur gauche et de l'angle inférieur droit de la partie sélectionnée et  $\mu = 0$  indique la direction de la concaténation :

$$\mu = \begin{cases} v, & \text{la partie est concaténée suivant l'axe vertical} \\ h, & \text{la partie est concaténée suivant l'axe horizontal} \end{cases}$$

Pour assurer un haut niveau d'intégration entre un média physique et un média dérivé, aucune distinction n'est faite dans leur traitement et leur présentation. Dans ce qui suit, ils sont tous deux désignés par le mot média.

### 2.1.2 Les attributs d'un média

Afin de mettre en œuvre les objectifs arrêtés, un ensemble d'informations doit être défini pour chaque type de média. Les tableaux suivants donnent les attributs retenus par type de média.

#### - Les attributs d'une image

Les attributs	Type	Valeur par défaut	
<b>Généraux</b>			
Nom	String		
Format	Image		
Localisation	Liste	Chemin	
		Coordonnées gauches supérieures	(0,0)
		Coordonnées droites inférieures	(n,m)
		Direction = {h, v}	h
Répéter	Entier positif	1	
État	{ activé, désactivé }	désactivé	
<b>Spatiaux</b>			
Abscisse Supérieure Gauche	Entier (pixel)		
Ordonnée Supérieure Gauche	Entier (pixel)		
Largeur	Entier (pixel)		
Hauteur	Entier (pixel)		
<b>Temporels</b>			
Début	Entier (seconde)		
Fin	Entier (seconde)		
Durée calculée	Entier (seconde)		
Durée Pref	Entier (seconde)		
Durée Max	Entier (seconde)		
Durée Min de présentation	Entier (seconde)	0	
<b>De navigation</b>			

Destination	Entier (instant)	
	String	
<b>De recouvrement</b>		
Recouvrable	Booléen	false
Tolérance gauche	Entier (pixel)	0
Tolérance droite	Entier (pixel)	0
Tolérance sup	Entier (pixel)	0
Tolérance inf	Entier (pixel)	0
zoom	Entier (%)	100%

- Les attributs d'un texte

Les attributs	Type	Valeur par défaut
<b>Généraux</b>		
Nom	String	
Format	Texte	
Localisation	Chemin	
	Numéro de ligne et de colonne du premier caractère	
	Numéro de ligne et de colonne du dernier caractère	
	Saisie	
Répéter	Entier positif	1
État	{ activé, désactivé }	désactivé
<b>Spatiaux</b>		
Abscisse Supérieur Gauche	Entier (pixel)	
Ordonnée Supérieur Gauche	Entier (pixel)	
Largeur	Entier (pixel)	
Hauteur	Entier (pixel)	
<b>Temporels</b>		
Début	Entier (seconde)	
Fin	Entier (seconde)	
Durée calculée	Entier (seconde)	
Durée Pref	Entier (seconde)	
Durée Max	Entier (seconde)	
Durée Min de présentation	Entier (seconde)	0
<b>De navigation</b>		
Destination	Entier (instant)	
	String	
<b>De recouvrement</b>		
Recouvrable	Booléen	false
Tolérance gauche	Entier (pixel)	0
Tolérance droite	Entier (pixel)	0
Tolérance sup	Entier (pixel)	0
Tolérance inf	Entier (pixel)	0

- Les attributs d'une audio

Les attributs	Type	Valeur par défaut	
<b>Généraux</b>			
Nom	String		
Format	Audio		
Localisation	Liste	Chemin	
		Début partie	0
		Fin partie	fin intrinsèque
Répéter	Entier positif	1	
État	{ activé, désactivé }	désactivé	
<b>audio</b>			
volume	Entier (%)	50%	
<b>Temporels</b>			
Début	Entier (seconde)		
Fin	Entier (seconde)		
Durée calculée	Entier (seconde)		
Durée Pref	Entier (seconde)	durée intrinsèque	
Durée Max	Entier (seconde)		
Durée Min de présentation	Entier (seconde)	0	
<b>De navigation</b>			
Avance	Booléen	true	
Recul	Booléen	true	
Pause	Booléen	true	
Fermer	Booléen	true	

Les Attribut d'une vidéo

Les attributs	Type	Valeur par défaut	
<b>Généraux</b>			
Nom (unique)	String		
Format	Vidéo		
Localisation	Liste	Chemin	
		Début partie	0
		Fin partie	fin intrinsèque
Répéter	Entier positif	1	
État	{ activé, désactivé }	désactivé	
<b>audio</b>			
volume	Entier ( % )	50%	
<b>Spatiaux</b>			
Abscisse Supérieur Gauche	Entier (pixel)		
Ordonnée Supérieur Gauche	Entier (pixel)		
Largeur	Entier (pixel)		
Hauteur	Entier (pixel)		
<b>Temporels</b>			
Début	Entier (seconde)		
Fin	Entier (seconde)		
Durée calculée	Entier (seconde)		
Durée Pref	Entier (seconde)	durée intrinsèque	
Durée Max	Entier (seconde)		

Durée Min de présentation	Entier (seconde)	0
<b>De navigation</b>		
Destination	Entier (instant)	
	String	
Avance	Booléen	true
Recul	Booléen	true
Pause	Booléen	true
Fermer	Booléen	true
<b>De recouvrement</b>		
Recouvrable	Booléen	false
Tolérance gauche	Entier (pixel)	0
Tolérance droite	Entier (pixel)	0
Tolérance sup	Entier (pixel)	0
Tolérance inf	Entier (pixel)	0

## 2.2 Le document Multimédia

Dans notre modèle, nous distinguons deux types de documents multimédia : le document primaire et le parcours de lecture.

### 2.2.1 Le document Primaire

C'est un document valide créé par l'auteur. Un document valide est un document composé d'au moins un média, il est cohérent temporellement et spatialement et il ne possède aucun recouvrement non désiré et aucun média marqué à désactivé (cette notion est expliquée dans le chapitre 7).

### 2.2.2 Le parcours de lecture

Nous allons commencer par définir les concepts de document dérivé et de scène qui rentrent dans la définition d'un parcours de lecture.

#### 2.2.2.1 Le document dérivé

Un document dérivé est constitué d'une partie ou de l'ensemble des médias d'un document primaire organisés en scènes. Différents documents dérivés peuvent être définis à partir du même document primaire. Ils forment des réservoirs pour la construction de parcours de lecture. Un document dérivé peut être défini aussi bien par l'auteur que par le lecteur.

#### 2.2.2.2 La scène

Une scène est un regroupement de médias véhiculant une information supposée cohérente par rapport à l'information contenue dans le document primaire (Fig 3.). Afin de s'assurer de la cohérence d'une scène, nous posons la condition suivante : Quels que soient deux médias d'une même scène, sauf si elle est composée d'un seul média, il faut qu'ils soient liés par une relation temporelle directe ou indirecte, si elle est indirecte elle doit passer uniquement par les médias de cette scène. Formellement, la scène est représentée par un graphe non orienté  $S=(M,R)$  où  $M$  est l'ensemble des sommets représentant les médias de la scène et  $R$  l'ensemble des arêtes représentant les relations temporelles. Afin que la scène soit cohérente, nous posons la condition suivante : le graphe  $S$  doit être connexe.

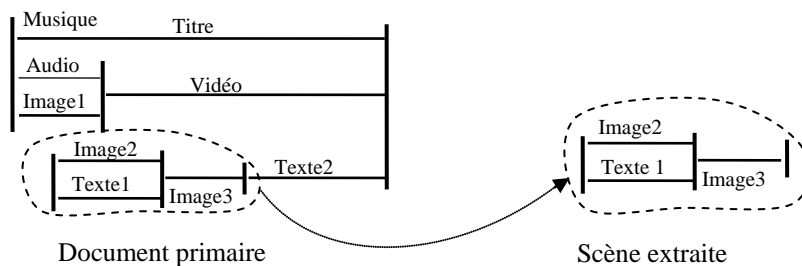


Fig 3. Définition d'une scène

Une scène peut être définie par l'auteur et par le lecteur.

### 2.2.2.3 Le parcours de lecture

Un parcours de lecture est défini par un ensemble de scènes d'un même document dérivé. Il est valide par construction. Plusieurs parcours peuvent être définis d'un même document dérivé. La figure 4 illustre un exemple de définition de trois parcours de lecture : *débutant*, *intermédiaire* et *avancé*, en fonctions du niveau des apprenants dans le cadre d'un cours interactif.

Afin de préserver l'information contenue dans le document primaire, dans un parcours de lecture un média ne doit pas appartenir à plus d'une scène. L'ordonnancement temporel et le placement spatial du parcours de lecture sont ceux définis dans le document primaire, toutefois pour plus de commodités de lecture, nous permettons la définition d'un délai entre les scènes séquentielles d'un parcours (fig 5.).

Un parcours de lecture peut être défini par l'auteur et par le lecteur.

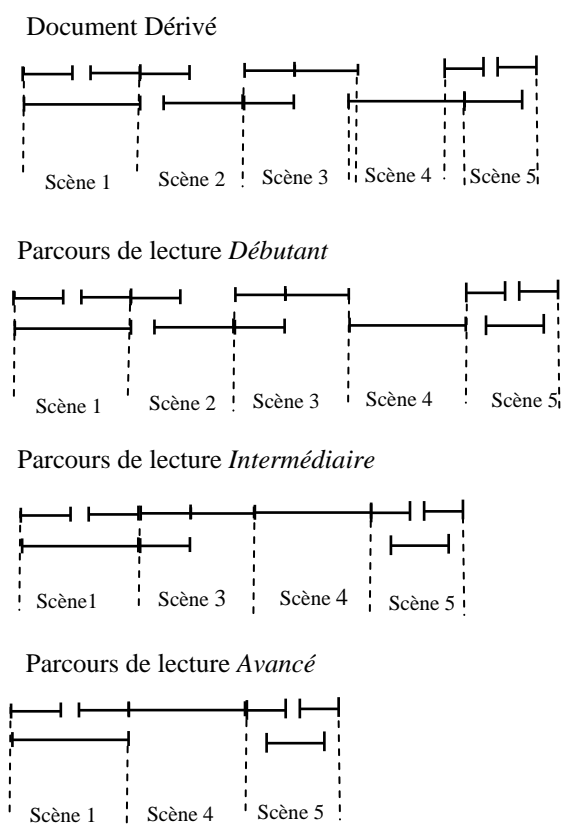


Fig 4. Définitions de parcours

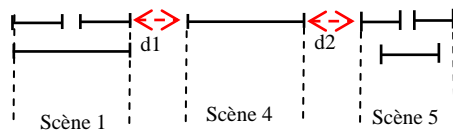


Fig 5. Définition de délais dans un parcours de lecture

### 3. Modèle de document multimédia

Nous allons dans ce qui suit présenter les quatre dimensions du modèle.

#### 3.1 La dimension logique

C'est sur le modèle logique que repose toute la conception d'un document multimédia. A ce niveau, toutes les options retenues doivent apparaître. Le modèle proposé (Fig 6) est hiérarchique et basé sur les concepts de document primaire, de document dérivé, de scène et de parcours de lecture [70].

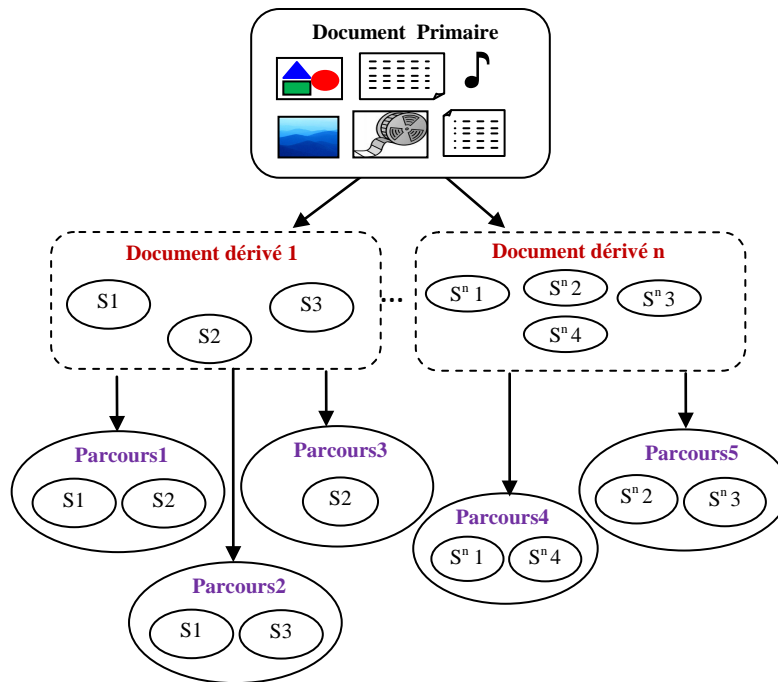


Fig 6. Modèle logique de document Multimédia

Afin de limiter les coûts de création, en assurant une réutilisabilité des composants, les médias dérivés, les documents dérivés, les scènes et les parcours de lecture sont des entités logiques, ils n'ont pas d'existence physique : ils partagent les mêmes données physiques avec le document primaire.

## 3.2 La dimension spatiale

La dimension spatiale d'un média est composée des informations de la taille de l'espace qu'il occupe lors de son affichage et des informations de son emplacement (position sur l'écran). Les informations typographiques sont supposées déjà définies pour chaque média.

### 3.2.1 La taille de l'emplacement

La taille de l'espace d'affichage d'un média est donnée par les attributs *largeur* et *hauteur* qui sont automatiquement calculées ou introduits par l'auteur. Les médias considérés dans notre modèle de documents, à l'exception de l'audio, possèdent une forme rectangulaire [67]. Un média est représenté par une boîte correspondant au rectangle minimal qui l'englobe. Ce dernier est déterminé par les coordonnées de l'angle supérieur gauche et de l'angle inférieur droit du média (voir figure 7).

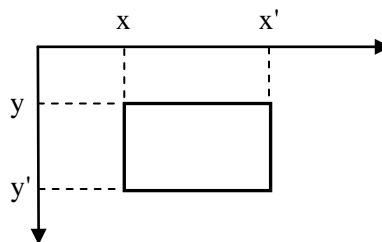


Fig 7. Représentation spatiale d'un média

### 3.2.2 Les informations de l'emplacement

La position d'un média est déterminée à l'aide des relations spatiales qui le lient aux autres médias et des informations de placement spécifiés par l'auteur. Elle est donnée par les coordonnées de son angle *supérieur gauche*. Les relations sont spécifiées par un langage à base de relations. Elles sont soumises à des mécanismes d'analyse et de synthèse utilisant une approche à base de contraintes. Ces trois aspects (langage, analyse et synthèse) sont développés dans les chapitres 4 et 5. Le placement peut être introduit directement par l'auteur.

Dans le modèle, seules les informations *Abscisse gauche supérieur*, *Ordonnée gauche supérieur*, *largeur* et *hauteur* d'un média sont retenues.

## 3.3 La dimension temporelle

Les informations temporelles d'un média sont composées des informations : *début*, *fin*, *durée calculée*, *durée pref*, *durée max* et *durée min-de-présentation*.

L'information *durée calculée* est la durée réelle du média lors de son affichage, elle est décrite comme un sous ensemble de  $\mathcal{R}^+_0$  indiquant ses valeurs possibles. Dans notre modèle, elle est prise dans l'intervalle  $[pref, max]$  où *pref* et *max* sont des valeurs spécifiées par l'auteur. La valeur *pref* représente la durée préférentielle pour un média. Pour les médias continus (vidéo et audio), elle correspond par défaut à leur durée intrinsèque. La valeur *max* représente la durée maximale de présentation tolérée par l'auteur pour un média.

La *durée calculée* est déterminée en fonctions des relations temporelles du média définissant son ordonnancement et les informations (début, fin, durée *pref* et durée *max*) introduites par l'auteur ou déduites. L'ordonnancement est spécifié par un langage relationnel. L'analyse et la synthèse utilisent une approche à base de contraintes. Ces trois aspects sont développés dans les chapitres 4 et 5

Pour des raisons propres à l'auteur, ce dernier pourrait vouloir qu'un média soit présenté durant un certain temps avant qu'il ne puisse être arrêté ou servir de point de départ pour un lien, pour cela nous introduisons le paramètre *durée-min-présentation* indiquant le temps minimum de présentation d'un média.

### **3.4 La dimension hypermédia**

Contrairement au document classique, lors de la présentation d'un document multimédia l'aspect temporel des médias (apparition et disparition d'un média, changement progressif de l'état d'un média, etc.) lui confère une certaine complexité de perception. De nouveaux besoins, comme un retour sur un média particulier, un détail passé ou un saut vers un instant donné, doivent être considérés pour maîtriser cette complexité et permettre par la même une meilleure lecture du document. La réponse réside dans les capacités d'interaction offertes par le système. Dans notre modèle, deux types d'interactions sont considérés : la navigation et les fonctions de manipulation.

#### **3.4.1 Navigation**

La navigation dans un document multimédia est l'ensemble des opérations permettant d'exécuter des sauts temporels internes au document et des inclusions de documents externes dans le document. Deux types de navigation sont considérés: la navigation par liens et la navigation libre.

##### **3.4.1.1 Navigation par liens**

Un lien est défini par une ancre de départ et une ancre d'arrivée. L'activation de l'ancre de départ est restreinte à son laps de temps de présentation. L'ancre d'arrivée peut correspondre à un média, à un instant précis de la présentation ou à un autre document.

Dans notre modèle, nous définissons deux types de liens : le saut temporel et le lien référence. Un saut temporel est un lien interne au document ou au parcours de lecture, son activation entraîne un saut vers le début ou vers un instant d'un média ou d'une scène.

Une référence est un lien vers un document externe, à son activation le document courant est interrompu et la destination est exécutée. Le document courant reprend sa présentation à partir du point d'interruption [70].

Dans le modèle proposé, nous définissons les liens suivants (Fig 8.) :

- un saut temporel d'un média vers un média ou vers une scène,
- un saut temporel d'un média vers un instant de la présentation,
- et une référence d'un média vers un document externe.

Afin de préserver la cohérence et la sémantique de l'information contenue dans le document primaire, seul l'auteur peut définir des sauts et des références au niveau du document primaire et des parcours de lecture qu'il définit. Le lecteur ne peut définir que des liens de type sauts temporels au niveau de ses parcours de lecture. De plus, lorsque la destination d'un saut temporel défini dans le document primaire ne fait pas partie d'un parcours de lecture, deux possibilités se présentent, soit que le lien est désactivé soit que la définition du parcours selon est reprise.

Enfin, une attention particulière doit être accordée à la matérialisation des ancres des liens pour leurs signalisations.

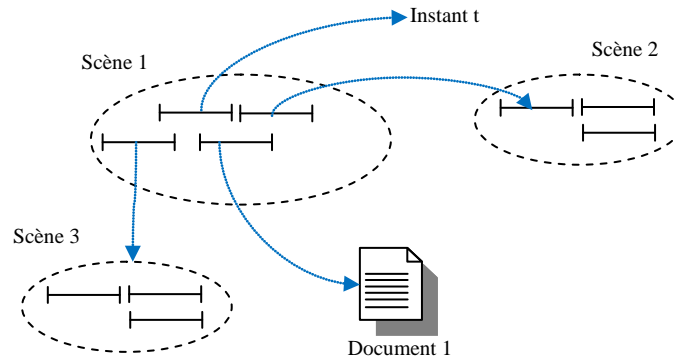


Fig 8. Liens possibles

### 3.4.1.2 Navigation libre

Dans le cas où le lecteur s'intéresse à un média ou à une scène particulière, une vue générale de son document ou de son parcours est mise à sa disposition, d'un simple click, il peut visualiser le média ou la scène qu'il désire.

### 3.4.2 Fonctions de manipulation

#### 3.4.2.1 Fonctions de manipulation d'un document primaire ou d'un parcours de lecture

Pour répondre, par exemples aux besoins d'avoir une idée rapide sur le contenu d'un document ou de s'arrêter sur un détail, le lecteur dispose d'un ensemble d'opérations de manipulation fournies à travers des boutons (avance rapide, retour rapide, pause, reprise etc.), qui n'apparaissent pas dans la spécification du scénario du document.

#### 3.4.2.2 Fonctions de manipulation d'un média

Le lecteur peut avoir besoin par exemple de revoir une partie d'une vidéo ou de réécouter une séquence audio, pour cela, la présentation du document est alors arrêtée et seul le média concerné est présenté avec les fonctions associées (avance, retour, zoom, etc.).

## 4. Conclusion

Dans ce chapitre, nous avons proposé un modèle de document multimédia où ses quatre dimensions sont à la fois considérées et permettant l'adaptation du contenu d'un document grâce à la définition des parcours de lecture. De plus, l'utilisateur, n'est plus considéré comme un acteur passif, il peut interagir avec le système en définissant ses propres parcours de lecture. A cet effet, un ensemble de concepts a été proposé, allant du média dérivé jusqu'au parcours de lecture, pour lesquels des règles de construction ont été définies afin de préserver la cohérence du document primaire.

Pour une meilleure perception du contenu d'un document, des fonctions de navigation et de manipulation ont été proposées.

Dans le chapitre suivant, nous allons présenter les modèles proposés pour la spécification des relations temporelles et spatiales.

# Chapitre 4

## Modèles de spécification des relations temporelles et spatiales

### 1. Introduction

La gestion de la synchronisation temporelle, de l'emplacement spatial et du recouvrement des médias constitue le point essentiel et le plus délicat dans la conception de documents multimédia interactifs. Pour les deux premiers aspects, il s'agit de pouvoir exprimer les relations temporelles et spatiales qui existent entre les médias et de garantir leur cohérence. Cette gestion est représentée par un modèle composé de trois parties [83]:

- *Un langage temporel/spatial* qui permet la spécification des relations entre les médias.
- *Des mécanismes d'analyse* qui permettent de prévenir les incohérences ou de les détecter à l'édition.
- *Des mécanismes de synthèse* qui permettent statiquement ou dynamiquement de produire une présentation conforme aux spécifications de l'auteur.

Enfin, il est utile de rappeler à ce niveau que la puissance des fonctions d'édition, la facilité d'utilisation du système et la recherche d'un haut niveau d'intégration des quatre dimensions d'un document sont des qualités très recherchées et par conséquent elles constituent pour nous des objectifs à atteindre.

Dans ce chapitre, nous allons étudier et proposer des modèles pour les langages de spécification des informations temporelles et spatiales, l'analyse, la synthèse et le recouvrement spatial des médias seront traitées dans le chapitre suivant.

Nous allons commencer par présenter quelques concepts utiles pour la définition des modèles temporel et spatial.

### 2. Scénario d'un document multimédia

Dans cette section, nous présentons la notion de scénarios multimédia et les différentes façons de les décrire.

#### 2.1 Définitions

Un scénario multimédia définit comment les médias d'un document s'enchaînent dans le temps et leurs emplacements spatiaux.

À tout scénario, peuvent correspondre une ou plusieurs traces d'exécution qui le respectent. Une trace d'exécution est définie de la façon suivante : c'est une suite de triplets (instant, ensemble d'observations, séquence d'actions). Chaque action correspond au démarrage d'un média et une observation à une terminaison [57].

On peut distinguer deux types de scénarios dans une présentation multimédia [55][56] :

- Scénarios déterministes : ils correspondent à des enchaînements entièrement définis avant la présentation d'un document. On peut calculer statiquement une date de début et une date de fin pour tous les médias. Cela suppose en particulier qu'on connaisse la durée de tous les éléments. La trace d'un scénario déterministe est unique.
- Scénarios indéterministes : ils correspondent à des enchaînements qui sont définis en fonction de données connues uniquement au moment de la présentation (occurrence d'une interaction utilisateur). Un scénario indéterministe est caractérisé par l'existence d'un ensemble de traces qui lui correspondent.

## 2.2 Compositions d'un scénario

On peut composer un scénario de deux façons [51]. La première consiste, pour l'aspect temporel, à décrire un scénario à travers une trace particulière, c'est le principe des *timelines*. Cela revient à le considérer comme un ensemble d'instantanés indépendants qui se rapportent à un repère temporel unique : le début du document. Et pour l'aspect spatial, elle consiste à spécifier les coordonnées absolues des médias. Dans ce cas les relations entre les médias sont implicites et l'ordre induit est total. L'analyse et la synthèse sont dans ce cas inutiles, et la composition du document peut être exploitée telle quelle pour ordonnancer la présentation. Ce type de composition ne permet pas de spécifier des scénarios indéterministes. De plus, il n'est pas adapté à la nature incrémentale du processus d'édition. En effet, toute modification nécessite une reconsidération globale de la spécification, du fait de l'incapacité de ce type de spécification à expliciter les relations temporelles et spatiales entre les médias.

À l'opposé, la seconde façon consiste à décrire le scénario à partir d'une mise en relation des différents médias qui le composent. Cette forme relative permet de rendre explicite toutes les dépendances temporelles et spatiales du scénario et induit un ordre partiel entre ses différents instantanés et emplacements. Cette spécification permet d'obtenir des structures abstraites plus facile à organiser et à mettre à jour. Cependant, ces qualités ne sont pas acquises sans contrepartie. La nature de la représentation des relations introduit des risques d'incohérences d'où la nécessité de l'accompagner par des mécanismes d'analyse et de synthèse [48][49].

C'est cette dernière approche que nous avons retenue.

## 3. Critères d'évaluation d'un modèle de spécification

Avant d'aborder les modèles de spécification, il convient de préciser d'abord les critères qui permettent d'évaluer un modèle donné. Un modèle doit répondre aux objectifs ci-dessous :

1. Puissance d'expressivité : le langage du modèle doit permettre des spécifications arbitrairement complexes. Une évaluation de l'expressivité peut se ramener à une mesure du nombre de schémas de synchronisation exprimables [12].
2. L'intuitivité et la facilité d'utilisation : l'auteur doit pouvoir retrouver dans les relations du modèle, la représentation mentale qu'il se fait de son scénario. Aucune contrainte d'utilisation, autres que celles inhérentes à la conception de son document, ne doit entraver la création de l'auteur [57].
3. Vérification de la cohérence temporelle et spatiale : le modèle doit être doté de mécanismes permettant de détecter et d'indiquer à l'auteur les incohérences d'une spécification donnée.
4. Adaptation à la nature incrémentale du processus d'édition : la création d'un document se fait généralement par une modification progressive de son contenu et de sa structure temporelle et spatiale. L'incrémentalité, qui consiste à prendre en compte les opérations d'édition une par une, doit être supportée [50].

5. Spécification et gestion de l'indéterminisme pour le modèle temporel : le modèle doit permettre la spécification et la gestion de l'indéterminisme.
6. Spécification du recouvrement désiré pour le modèle spatial : l'auteur doit pouvoir exprimer des recouvrements désirés sans que cela ne soit considéré comme une incohérence spatiale.
7. Performance temporelles des mécanismes d'analyse et de synthèse : un environnement d'édition et de présentation de documents multimédia est un environnement interactif, d'où la nécessité d'apporter une attention particulière aux performances temporelles des techniques employées [24].

Les points 3, 5 et 7 sont traités dans le chapitre suivant consacré à l'analyse et la synthèse.

## 4. Modèle temporel

Avant de présenter le modèle de spécification des relations temporelles adopté, il est utile d'aborder quelques notions essentielles rentrant dans sa définition.

### 4.1 Modélisation de l'information temporelle de base

Nous allons commencer par la définition des entités de base qui composent un scénario. En particulier, nous caractérisons le comportement temporel de chaque média indépendamment d'un scénario donné. Cette étape nous permet de modéliser les différents médias de façon homogène et indépendante de leur contenu et de leur type (audio, vidéo, texte, image) et d'assurer ainsi le niveau d'intégration recherché pour les médias.

Tout élément multimédia peut être mis en relation à travers trois informations temporelles [17] :

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

L'une de ces trois informations est redondante. En effet, il est possible de calculer l'une en fonction des deux autres. Le choix des informations retenues fait partie du langage et donc du modèle, car la mise en relation des éléments se fait à partir de ces informations.

### 4.2 Domaine de validité temporelle

La durée d'un média est souvent décrite comme un sous ensemble de  $\mathfrak{R}^+$  indiquant ses valeurs possibles. Dans [57] on parle de *domaine de validité*, il est défini par un triplet de valeurs  $[min, nom, max]$  où *min* représente la borne inférieure acceptable, *max* la borne supérieure acceptable et *nom* la durée calculée par le système [60].

Dans notre modèle, nous définissons le domaine de validité par l'intervalle  $[pref, max]$  où *pref* et *max* sont des valeurs positives spécifiées par l'auteur (fig 1). La valeur *pref* représente la durée de présentation préférentielle pour un média. Pour les médias continus (vidéo et audio), elle correspond, par défaut, à leurs durées intrinsèques. La durée *max* représente la valeur maximale de présentation tolérée.

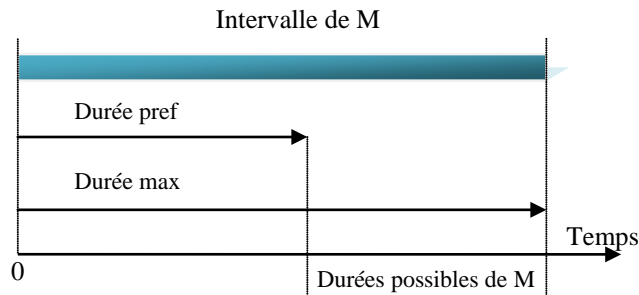


Fig 1. Domaine de validité temporelle

Si on se restreint à la manipulation des médias à travers ces domaines, il devient possible d'abstraire leur contenu au sein du modèle et de rendre homogène leur représentation et leur traitement.

### 4.3 Représentation du déroulement temporel d'un scénario

Il existe deux façons de représenter le déroulement temporel d'un scénario : à travers les changements qui surviennent (le début de la vidéo1 correspond au début de l'audio2) ou au contraire en reliant globalement les activités entre elles (le texte1 est présenté pendant l'image3). Ceci débouche sur deux modes de représentations :

1. Une représentation fondée sur les *instants*. Dans ce cas, un média est décrit dans un scénario par un instant de début et un instant de fin, comme dans Firefly [18] et Maestro [26].
2. Une représentation fondée sur les *intervalles*. Le média est considéré comme une entité temporelle décrite par sa durée comme dans OCPN [30], Cmifed [85], Madeus [57].

### 4.4 Expression des relations temporelles

Les relations temporelles décrivent la façon dont les médias doivent être combinés temporellement afin de produire le scénario temporel d'un document. Étant donnée qu'il existe deux modes de représentation temporelles d'un scénario : les instants et les intervalles, il en résulte deux classes de relations temporelles. Les relations fondées sur les instants et les relations fondées sur les intervalles.

#### 4.4.1 Modèles à base d'instant

Dans les modèles temporels à base d'instant (PA : *Point Algebra*), les unités temporelles considérées sont les instants de début et de fin des médias. Étant donné deux instants dans un scénario, trois relations peuvent exister entre eux. Un instant peut en précéder ( $<$ ), lui succéder ( $>$ ) ou lui être égal ( $=$ ). L'ensemble des relations primitives de l'algèbre d'instant est noté  $B = \{<, >, =\}$  [107].

Dans certains cas, les relations entre certains instants d'un scénario peuvent être connues de façon moins précise. Afin d'illustrer ce type de situations, supposons que la seule relation que nous connaissons entre deux instants  $e1$  et  $e2$  est que  $e1$  ne peut avoir lieu avant  $e2$ . Cela signifie que  $e1$  a lieu après ou en même temps que  $e2$ . Cette relation est une relation composée, elle est notée  $e1 > e2 \vee e1 = e2$  ou encore  $e1 \{>, =\} e2$ .

L'ensemble des relations composées ou disjonctives, noté  $2^B$ , représente des disjonctions des relations primitives. Avec 3 relations à base d'instant, on peut construire  $2^3 = 8$  disjonctions. A chaque relation, on associe une notation particulière, les 8 relations indéfinies sont donc :  $\emptyset, \leq, <, =, >, \geq, \neq, ?$ , où ? représente une disjonction de l'ensemble des relations primitives et  $\emptyset$  est l'ensemble vide.

De façon intuitive les relations primitives  $<$ ,  $>$ ,  $=$  sont utiles pour la composition multimédia. En est-il de même pour les autres relations  $\{<=, >=, \neq, ?\}$  ? Afin de répondre à cette question, nous devons d'abord tenir compte des caractéristiques temporelles d'une présentation multimédia. Dans le domaine du multimédia, il est admis que de légères fluctuations sur l'instant de début et sur la durée des médias sont tolérées [47]. Par exemple, la présentation d'un média audio une milli seconde plus tard ou plus tôt que sa date prévue n'affecte en rien la qualité de la présentation. Ceci signifie qu'il n'y a pas de différence perceptible si on spécifie entre deux instants  $e1$  et  $e2$  la relation  $e1 < e2$  ou  $e1 <= e2$ . De façon similaire la relation  $\neq$  se démarque de la relation  $?$  sur un seul instant ( $=$ ). Les quatre relations d'instant :  $\{<, =, >, ?\}$  sont donc les seules qui sont utiles pour la spécification de scénarios multimédia [108][77]. Un modèle temporel à base d'instant devrait donc être capable d'exprimer au moins cet ensemble de relations.

#### 4.4.2 Modèle à base d'intervalles

Dans les relations à base d'intervalles (IA : *Interval Algebra*), les relations possibles entre deux médias se réduisent à toutes les combinaisons de positionnement possibles de deux intervalles sur une droite orientée. Le modèle le plus général, proposé par Allen J.F. [1], dresse la liste exhaustive de toutes ces relations. La liste des combinaisons possibles entre deux médias comporte 13 relations (fig 2), 7 relations de base et leurs relations inverses, l'égalité étant l'inverse d'elle-même. La dernière colonne de la figure 2, représente 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  $x$  [57].

Relation	Symbole	Inverse	Relation à base d'instant équivalente
x before y	b	bi	$x^- < x^+ < y^- < y^+$
x meets y	m	mi	$x^- < x^+ = y^- < y^+$
x overlaps y	o	oi	$x^- < y^- < x^+ < y^+$
y finishes x	f	fi	$x^- < y^- < x^+ = y^+$
y during x	d	di	$x^- < y^- < y^+ < x^+$
x starts y	s	si	$x^- = y^- < x^+ < y^+$
x equals y	e	ei	$x^- = y^- < x^+ = y^+$

Fig 2. Relations à base d'intervalle

De façon analogue aux relations à base d'instant, il existe  $2^{13} = 8192$  relations composées.

Afin de mesurer l'apport de l'algèbre d'intervalles au multimédia, il faut commencer d'abord par la comparer avec l'algèbre d'instant. La dernière colonne de la figure 2 ci-dessus indique que chaque relation primitive à base d'intervalles est équivalente à plusieurs relations d'instant. S'il en est de même pour les relations disjonctives à base d'intervalles, alors, ces deux représentations seraient équivalentes [101]. La figure 3 donne un résumé du nombre de relations disjonctives de IA qui peuvent être générées à partir d'un sous-ensemble des relations composées de PA.

Relations à base d'instant de PA	Nombre de disjonctions de IA générées
$<, =, >$	13
$<, =, >, ?$	29
$<=, <, =, >, >=, ?$	82
$<=, <, =, >, >=, ?, \neq$	187

Fig 3. Relations disjonctives de IA engendrées par les relations de PA

La dernière ligne montre que seules 187 sur les 8192 relations à base d'intervalles sont exprimables à partir des relations à base d'instant. Ce fait témoigne de la plus grande richesse

des relations de IA par rapport à celle de PA. Sur les  $8192-187 = 8004$  relations non exprimables moyennant PA, il existe plusieurs relations qui peuvent être utiles dans un scénario multimédia. En particulier, la relation  $A \{b, b_i, m, m_i\} B$  qui dénote l'exclusion mutuelle entre deux intervalles. Cette relation peut être utilisée pour traduire l'impossibilité de présenter A et B en même temps [47]. Notons également que les relations d'instants jugées utiles pour le multimédia dans PA engendrent uniquement 29 relations de IA. Plusieurs auteurs [77][107][89] les retiennent d'ailleurs comme une référence pour mesurer l'expressivité d'un langage de synchronisation temporelle multimédia.

#### 4.4.3 Choix entre algèbres d'intervalles et d'instants

Dans le cadre des systèmes d'édition de documents multimédia, plusieurs arguments plaident en faveur d'une spécification des relations temporelles à base d'intervalles [57] :

- Les opérations d'édition portent sur les médias dont la projection dans le temps correspond à des intervalles.
- La description d'un scénario est syntaxiquement plus compacte.
- Elle répond mieux au critère d'encapsulation et de modularité puisqu'il est possible d'une part de délimiter et de regrouper des portions d'un scénario sous forme d'un seul intervalle fictif (les scènes) ou de dériver des média à partir d'un média, et d'autre part d'isoler les relations qu'il contient afin de le relier aux autres portions du scénario.

En revanche, la représentation à base d'instants fournit un très bon support d'exécution pour la présentation. En effet, on peut considérer que l'état du système de présentation ne change pas de façon continue dans le temps mais évolue lors de l'apparition d'événements discrets, on parle de Système Dynamique à Événements Discrets) [91].

Ce qui serait intéressant c'est bien évidemment de pouvoir exploiter les deux représentations. Cela est possible dans notre approche car les relations temporelles et spatiales seront traduites, au niveau du résolveur, en relations à base d'instants.

### 4.5 Modèle temporel proposé

Les caractéristiques recherchées dans un modèle temporel sont sa puissance d'expressivité, sa facilité d'utilisation, son degré d'intuitivité et la possibilité de spécification de l'indéterminisme. Divers modèles temporels ont été proposés dans la littérature [1][59][39][75][53] [18][26] [108] [48][8][54][94], où les plus connus sont généralement des modèles impératifs qui ont un degré d'expressivité assez puissant et qui gèrent l'indéterminisme, pour quelques uns, mais ils ne sont ni intuitifs ni faciles à manipuler pour des utilisateurs non informaticiens.

En partant, des travaux de [107][77][90] d'où il ressort que tout modèle temporel doit permettre l'expression des 29 relations à base d'intervalle jugées utiles pour le multimédia, du modèle de Wahl et Rothermel [108] et de l'introduction du concept du délai flexible, nous proposons un modèle temporel qui contribue à l'amélioration des critères mentionnés.

Avant de présenter le modèle, nous commençons par donner quelques définitions utiles pour la définition de notre modèle.

#### - La relation à base d'intervalle

Une relation entre deux médias A et B décrit la combinaison de positionnement des intervalles de A et B sur une droite temporelle orientée. Elle est notée  $B R(d) A$ , avec :

- R la relation,
- d le délai temporel pour les relations qui en admettent,
- A le média de référence,
- et B le média secondaire.

**- Lecture d'une relation**

Pour pouvoir interpréter une relation  $R$  reliant deux médias  $A$  et  $B$ , il est nécessaire de prendre une convention de lecture. Dans notre modèle, la relation  $B R(d) A$ , signifie que  $B$  est en relation directe avec  $A$ , c'est-à-dire que c'est  $A$  qui est pris comme média de référence et  $B$  comme média secondaire. Un média référence est supposé déjà défini dans le document.

**- Portée d'une relation**

Dans notre modèle, une relation est binaire, elle met en relation deux médias.

**4.5.1 Le modèle proposé**

Nous allons présenter le modèle proposé en se référant aux critères d'évaluation cités.

**4.5.1.1 Degré d'expressivité du modèle**

Le degré d'expressivité d'un modèle temporel est déterminé par les relations qui le composent et par le nombre et les valeurs des délais associés aux relations. Par conséquent, le degré d'expressivité se mesure par le choix de ses relations et la définition donnée au délai.

**- Définition du délai**

Un délai est défini par le laps de temps qui sépare les instants de début/fin de deux médias  $A$  et  $B$  en relation. Une relation peut admettre plus d'un délai.

Les valeurs associées à un délai et le nombre de délai associé à une relation, qui en admet, influent sur le degré d'expressivité d'un modèle. En effet, une relation avec un délai est plus expressive que la même relation sans délai et qu'une relation avec plusieurs délais est plus expressive que la même relation avec moins de délais. En partant de là, nous allons déterminer le nombre de délais possible pour une relation et les valeurs qu'ils peuvent prendre [68].

Il a été démontré dans [108] qu'à la spécification d'une relation, fournir la durée des deux médias en relation et plus de 3 paramètres de délai, détermine un ordre total de présentation, c'est-à-dire que les relations n'ont plus lieu d'exister. De là, et afin d'éviter des spécifications au dessus des relations, au plus 3 délais peuvent être définis pour une relation.

Pour déterminer les valeurs possibles d'un délai, nous allons examiner les interprétations des expressions que l'on peut formuler avec une relation à laquelle on associe un délai. Soit la relation  $B R(d) A$  où  $R$  est la relation *commence*, par exemple, et  $d$  le délai associé. La figure 4 donne les différentes expressions que l'on peut formuler avec la relation  $R$  et les valeurs du délai associé [68].

Expressions de la relation	Valeur du délai
B (commence) en même temps que A	délai=0
B (commence) x unités (avant, après) A	délai = x, x>0
B (commence) (avant, après) A	délai indéterminée
B (commence) au plus tôt x1 unités et au plus tard x2 unités (avant, après) A	$x1 \leq \text{délai} \leq x2$ , $x1, x2 > 0$ et $x2 > x1$

Fig 4. Valeurs possibles du délai

D'après les différentes expressions ci-dessous, le délai peut prendre une valeur nulle, fixe positive, indéterminée ou prise dans un intervalle. De là, un modèle temporel complet doit permettre l'expression de ces quatre valeurs de délai. Actuellement, dans les modèles existants à base d'intervalles, au maximum les trois premières valeurs sont considérées pour un délai.

Afin que le délai puisse prendre les 4 valeurs, nous définissons la notion de délai flexible. Contrairement aux modèles existants où le délai prend une valeur fixe, la valeur du délai flexible  $d$  est prise dans un intervalle,  $d \in [\delta^1, \delta^2] \subset \mathcal{R}^+$ , les valeurs des bords,  $\delta^1$  et  $\delta^2$ ,

peuvent être spécifiées ou indéterminées. Lorsqu'elles sont spécifiées, elles sont positives ou nulles ( $\delta^1, \delta^2 \geq 0$ ), sinon elles sont représentées par le caractère '-'.

La figure 5 donne l'interprétation du délai en fonction des valeurs des bords de l'intervalle.

Intervalle du délai	Interprétation
$[\delta^1, -]$	au plus tôt $\delta^1$ unités; $\delta^1 > 0$ .
$[-, \delta^2]$	au plus tard $\delta^2$ unités; $\delta^2 > 0$ .
$[\delta^1, \delta^2]$	au plus tôt $\delta^1$ unités et au plus tard $\delta^2$ unités; $\delta^1, \delta^2 > 0$ et $\delta^2 > \delta^1$
$[\delta, \delta]$	exactement $\delta$ unités; $\delta \geq 0$ .
$[-, -]$	délai indéterminé

Fig 5. Interprétations du délai

En plus de retrouver les 3 valeurs généralement utilisées pour un délai, la notion de délai flexible nous permet d'introduire les expressions *au plus tôt* et *au plus tard*, expressions qui ne peuvent être exprimées par aucun modèle existant à base d'intervalles. Pour l'exemple, prenant la relation *before*, dans le modèle de Wahl-Rothermel seules les 3 spécifications suivantes sont possibles [108] :

*B before (0) A*            A commence exactement après la fin de B.  
*B before (+) A*            A commence x unités après la fin de B,  $x > 0$ .  
*B before (\*) A*            A commence après la fin de B.

Avec le délai flexible proposé, nous avons 5 spécifications possibles [68]:

*B before [ $\delta^1, \delta^1$ ] A*    A commence exactement  $\delta^1$  unités après la fin de B;  $\delta^1 \geq 0$ .  
*B before [ $\delta^1, \delta^2$ ] A*    A commence au plus tôt  $\delta^1$  unités après la fin de B et au plus tard  $\delta^2$  unités après la fin de B;  $\delta^1 \geq 0$  et  $\delta^2 > \delta^1$   
*B before [-, -] A*            A commence après la fin de B.  
*B before [ $\delta^1, -$ ] A*        A commence au plus tôt  $\delta^1$  unités après la fin de B;  $\delta^1 \geq 0$ .  
*B before [-,  $\delta^2$ ] A*        A commence au plus tard  $\delta^2$  unités après la fin de B;  $\delta^2 \geq 0$ .

### - Choix des relations

Dans leur article intitulé, *Representing time in multimedia systems*, Wahl et Rothermel [107][108], en définissant 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. Ils ont ainsi augmenté l'expressivité des 29 relations tout en simplifiant l'édition d'un scénario.

Dans notre modèle, nous avons retenu les 10 relations de Wahl et Rothermel auxquelles nous avons ajouté les relations d'Allen *starts*, *finishes* et *equals*. La figure 6 donne les relations du modèle, leurs représentations graphiques et leur interprétation [68].


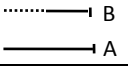
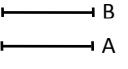
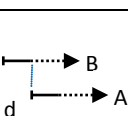
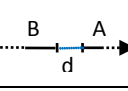
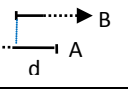
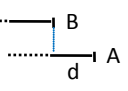
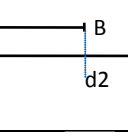
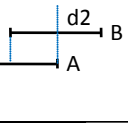
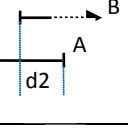
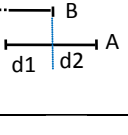
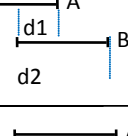
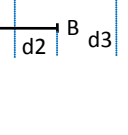
Relations temporelles	Représentation graphique	Interprétation
<i>B starts A</i>		<b>B</b> et <b>A</b> commencent en même temps.
<i>B finishes A</i>		<b>B</b> et <b>A</b> se terminent en même temps.
<i>B equals A</i>		<b>B</b> et <b>A</b> commencent et se terminent en même temps.
<i>B cobegin (#) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_2^2$ secondes avant début <b>A</b> .
<i>B before (#) A</i>		<b>B</b> se termine au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_2^2$ secondes avant début <b>A</b> .
<i>B beforeendof (#) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_2^2$ secondes avant fin <b>A</b> .
<i>B coend (#) A</i>		<b>B</b> se termine au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_2^2$ secondes avant fin <b>A</b> .
<i>B while (#, #) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes après début <b>A</b> et <b>B</b> se termine au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes avant fin <b>A</b> .
<i>B delayed (#, #) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes après début <b>A</b> et <b>B</b> se termine au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes après fin <b>A</b> .
<i>B startin (#, #) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes après début <b>A</b> et <b>B</b> commence au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes avant fin <b>A</b> .
<i>B endin (#, #) A</i>		<b>B</b> se termine au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes après début <b>A</b> et <b>B</b> se termine au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes avant fin <b>A</b> .
<i>B cross (#, #) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes avant fin <b>A</b> et <b>B</b> se termine au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes après début <b>A</b> .
<i>B overlaps (#, #, #) A</i>		<b>B</b> commence au plus tôt $\delta_1^1$ secondes et au plus tard $\delta_1^2$ secondes avant début <b>A</b> et <b>B</b> se termine au plus tôt $\delta_2^1$ secondes et au plus tard $\delta_2^2$ secondes après début <b>A</b> et au plus tôt $\delta_3^1$ secondes et au plus tard $\delta_3^2$ secondes après fin <b>A</b> .

Fig 6. Relations du modèle proposé

Avec :

- Le caractère # représente l'intervalle du délai.
- Les pointillés ( - - - ) associés aux instants (début ou fin) d'un média indiquent que ces instants ne sont pas connus.
- Le caractère | indique l'instant (début ou fin) sur lequel s'applique la relation.

Afin de mesurer l'apport du délai flexible pour l'expressivité du modèle, comparons le nombre de spécifications dans notre modèle et dans le modèle de Wahl-Rothermel qui est jugé très expressif. Le résultat est donné par la figure 7.

Relation	Nombre de spécifications Wahl-Rothermel	Nombre de spécifications du modèle proposé
<i>cobegin</i>	3	5
<i>before</i>	3	5
<i>beforeendof</i>	2	5
<i>coend</i>	3	5
<i>while</i>	9	25
<i>delayed</i>	4	25
<i>startin</i>	4	25
<i>endin</i>	4	25
<i>cross</i>	4	25
<i>overlaps</i>	8	125
<b>Total</b>	<b>44</b>	<b>270</b>

Fig 7. Comparaison de l'expressivité dans les deux modèles

Nous remarquons que le nombre total de spécifications dans le modèle proposé est plus élevé d'un facteur de 6 par rapport au modèle de Wahl-Rothermel.

Le nombre de spécification est déterminé par le nombre de délais associés à une relation et les combinaisons de leurs valeurs. Ainsi pour une relation à un seul délai nous avons 5 spécifications, pour une relation à deux délais nous avons  $5^2$  (25) spécifications et pour une relation à trois délais nous avons  $5^3$  (125) spécifications.

#### 4.5.1.2 Simplicité d'utilisation et intuitivité

En plus du nombre réduit des relations retenues et du choix de relations binaires, pour simplifier d'avantage la phase d'édition nous avons adopté la règle suivante : on ne peut définir qu'une seule relation entre deux médias. En effet, nous avons remarqué, empiriquement, que toute spécification au-delà d'une relation pouvait être ramener à une seule relation ou à une incohérence. Par exemples, spécifier les relations *B cobegin (#) A* et *B coend (#) A*, reviendrait à spécifier la relation *B while (#, #) A*. Par contre la spécification des relations *B cobegin (#) A et B while (#, #) A*, va provoquer une incohérence car *B* ne peut pas commencer en même temps avant et après *A*.

Enfin, l'ajout des relations d'Allen, *starts*, *finishes* et *equals*, au modèle se justifie par le fait qu'elles présentent un degré d'intuitives très élevé et qu'elles sont très utilisées dans les scénarios multimédia, et ce malgré le fait qu'elles peuvent être exprimées à travers, respectivement, les relations *cobegin*, *coend* et *while*.

#### 4.5.1.3 L'indéterminisme

L'expression de l'indéterminisme dans le modèle proposé, est non seulement possible, par exemple la relation *B before [-,-]A* est une forme d'indéterminisme qui indique seulement que *A* commence après *B*. De plus, le délai flexible offre dans certains cas des informations utiles pour la gestion de l'indéterminisme. Par exemple, de la relation *B before [ $\delta^1$ ,  $\delta^2$ ] A*, on déduit que le début de la présentation du média *A* est localisé entre les instants  $\delta^1$  et  $\delta^2$ . Il est à noter que dans notre étude, nous ne considérons que l'indéterminisme du aux interactions utilisateur et à la non connaissance des débuts/fins des médias, nous nous plaçons dans une logique de traitement local.

## 5. Modèle spatial

Il s'agit de pouvoir spécifier et de placer les médias, et notamment ceux qui se présentent en même temps, sur un espace limité (l'écran) et ce tout en garantissant la cohérence de leurs relations spatiales. Comme pour le modèle temporel, les caractéristiques recherchées dans un modèle spatial sont sa puissance d'expressivité, sa précision de placement, sa facilité d'utilisation, son degré d'intuitivité et la spécification du recouvrement désiré.

Avant de présenter notre modèle et de le situer par rapport aux modèles existants, nous allons présenter une étude critique des principaux modèles existant dans la littérature.

### 5.1 Modèles de présentation spatiale

Dans la littérature, il existe deux familles de modèles spatiaux : les modèles absolus et les modèles relatifs [13]. Dans un modèle absolu, un média est placé en spécifiant ses coordonnées. Dans un modèle relatif, le placement se fait en spécifiant des relations spatiales entre les médias. Dans cette dernière famille, qui nous intéresse, nous rencontrons plusieurs modèles de présentation dans un espace à deux dimensions, ils sont classés en deux familles : les modèles orientés intervalles et les modèles orientés régions [115].

#### 5.1.1 Les modèles orientés intervalles

Un média est représenté comme une composition d'intervalles représentant ses projections sur les axes  $x$  et  $y$ . Ainsi, le média est modélisé par la plus petite boîte rectangulaire le contenant (MBR : *Minimum Bounding Rectangle*). La composition d'une présentation revient à déterminer les positions absolues et/ou relatives des médias sur les axes  $x$  et  $y$ . Les positions relatives sont exprimées à l'aide des relations d'intervalles d'Allen [1]. Elles sont définies par les combinaisons des positionnements de deux intervalles sur une ligne droite orientée. Cet ensemble est composé des sept relations de base (*before, meet, overlap, finish, during, start et equal*) et de leurs relations inverses.

#### 5.1.2 Les modèles orientés régions

Les médias sont représentés par des régions telles que les rectangles [80][81], les cercles [86] ou des formes irrégulières [30]. Chaque région possède un repère comme par exemple son centre de gravité. La position absolue d'un média est donnée par la position de son repère et sa position relative est donnée par des relations spatiales. Le modèle le plus connu dans cette approche est celui de M. J. Egenhofer [30] où un média est représenté dans un espace à une, deux ou trois dimensions, par une région irrégulière décrite par ses facettes internes et externes et son emplacement est spécifié par des relations topologiques et directionnelles. Et ceux proposés dans [81][82][86], qui sont des cas particuliers du modèle de M. J. Egenhofer où un média est représenté par une région régulière comme des MBR (*Minimum Bounding Rectangle*) ou des MBC (*Minimum Bounding Circle*). Dans le domaine du raisonnement spatial, deux types de relations entre régions sont distingués : les relations topologiques et les relations directionnelles.

##### 5.1.2.1 Relations topologiques

La composition spatiale d'une présentation se fait par la spécification de relations entre les médias. Une manière de spécifier cette composition est l'utilisation des relations spatiales topologiques qui indiquent si deux médias se superposent, se touchent ou sont disjoints. Un ensemble complet de ces relations, représenté dans la figure 8, a été proposé par M. J. Egenhofer [29].

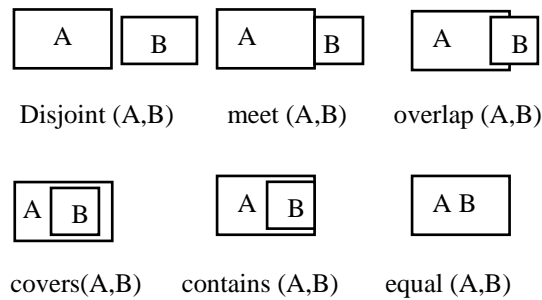


Fig 8. Relations Topologiques

### 5.1.2.2 Relations directionnelles

Les relations directionnelles caractérisent l'ordre des médias dans un espace. Une région est présentée par deux intervalles correspondant respectivement aux axes x et y. L'ensemble des relations directionnelles est obtenu en combinant les intervalles de deux médias sur les axes x et y. Sur chaque axe, il existe 13 relations possibles entre deux intervalles [1], nous avons donc  $13^2$ , soit 169 relations entre deux régions. Cet ensemble des 169 relations couvre également les relations topologiques du fait que chaque relation topologique peut être exprimée sous forme d'un sous ensemble de ces 169 relations [80].

Dans le modèle proposé par Papadias et al. [82], appliqué aux systèmes d'informations géographiques, les auteurs proposent le regroupement des 169 relations en un ensemble de 9 relations : *north*, *south*, *east*, *west*, *northeast*, *northwest*, *southeast*, *southwest* et *sameposition*, jugé suffisant pour exprimer n'importe quelle relation directionnelle entre deux régions.

### 5.1.2.3 Discussion

L'approche par intervalles permet de spécifier n'importe quel emplacement relatif de deux médias et permet également la spécification d'un recouvrement désiré. Néanmoins, l'emplacement est imprécis, aucune distance entre les emplacements des médias n'est définie. L'auteur ne dispose d'aucun moyen pour spécifier, par exemple, que le média B doit être placé exactement 5 unités après le média A. Certaines spécifications peuvent devenir très complexes, par exemple, pour exprimer que B peut être placé n'importe où à gauche de A, l'auteur doit spécifier l'ensemble des relations suivant :

$$\left\{ \begin{array}{l} B_x \text{ before } A_x \\ B_y \text{ (before ou meet ou overlap ou finish ou during ou start ou equal) } A_y \end{array} \right.$$

De plus, les relations ne sont pas intuitives du fait qu'elles ont une connotation temporelle.

Quant à l'approche par régions, elle a l'avantage d'offrir un degré d'expressivité élevé, d'être très intuitive et de permettre la spécification du recouvrement désiré. Toutefois, considérer un média comme étant un point (son centre de gravité par exemple) peut conduire à des emplacements qui ne respectent plus le sens des relations spécifiées. Par exemples, si l'auteur spécifie la relation *B east A*, qui signifie que B doit être placé à droite de A, peut conduire à un recouvrement non désiré (figure 9.a), voire même à une inclusion de A dans B (figure 9.b).

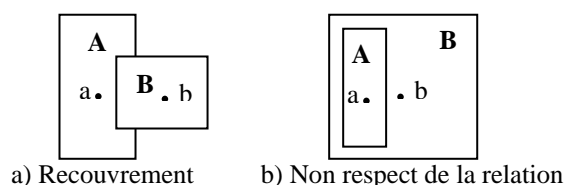


Fig 9. Génération d'emplacements non spécifiés

Une solution à ce problème serait de considérer et de calculer la distance séparant les deux repères. Ce calcul peut s'avérer très complexe, notamment dans le cas de régions irrégulières, comme le montre la figure 10, où il est difficile de déterminer la distance  $d_1$ .

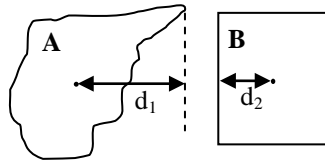


Fig 10. Complexité du Calcul d'une distance

Enfin, une certaine lourdeur peut être constatée lorsqu'une spécification nécessite plus d'une seule relation. Par exemple pour spécifier que le média B recouvre le média A par la droite, l'auteur doit utiliser les relations suivantes :

*B east A*  
*B cover A*

Toutefois, nous estimons que c'est l'approche par régions qui répond mieux à notre domaine d'étude moyennant quelques adaptations.

## 5.2 Modèle spatial proposé

Le modèle que nous proposons fait partie de l'approche par région. Nous allons présenter le modèle proposé en se référant aux caractéristiques mentionnées [71].

### 5.2.1 Degré d'expressivité du modèle

Comme pour le modèle temporel, le degré d'expressivité d'un modèle spatial se mesure par le choix de ses relations et la définition de la distance associée aux relations.

#### 5.2.1.1 Définition de la distance

Une distance est l'écart spatial d'emplacement défini entre deux médias reliés par une relation. Il est clair, qu'une relation avec une distance est plus expressive que la même relation sans distance. Afin d'augmenter le degré d'expressivité du modèle, nous allons, comme pour le modèle temporel, introduire l'idée de la distance flexible. Entre deux médias A et B, une distance flexible est définie comme étant l'écart entre les coordonnées de l'un des angles supérieur gauche ou inférieur droit de A avec les coordonnées de l'un des angles supérieur gauche ou inférieur droit de B et ce en fonction de la relation. Sa valeur est prise dans un intervalle :  $d_f \in [\delta^1, \delta^2] \subset \mathfrak{R}$ ,  $\delta^1 \leq \delta^2$ . Les valeurs des bords de l'intervalle,  $\delta^1$  et  $\delta^2$ , peuvent être spécifiées ou indéterminées. Lorsqu'elles sont spécifiées, elles sont positives, négatives ou nulles, sinon elles sont représentées par le caractère '-'.

Compte tenu du concept de boîte retenu pour la représentation spatiale d'un média, nous définissons au plus deux distances pour une relation:  $d_{fx} \in [\delta_1^1, \delta_1^2]$  et  $d_{fy} \in [\delta_2^1, \delta_2^2]$ , respectivement la distance sur l'axe des x et la distance sur l'axe des y (figure 11).

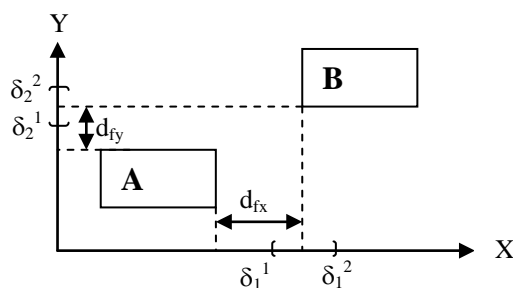


Fig 11. Exemple de définition d'une distance flexible

Ainsi, l'auteur peut spécifier des relations utilisant les expressions *au plus près* et *au plus loin*, expressions qui ne sont proposées dans aucun modèle existant. La figure 12 résume, en fonction de l'intervalle des valeurs de la distance, toutes les expressions possibles associées à une relation.

intervalle	Expressions possibles
$[\delta^1, \delta^2]$	au plus près $\delta^1$ unités et au plus loin $\delta^2$ unités ; $\delta^1 < \delta^2$ .
$[-, \delta]$	au plus loin $\delta$ unités.
$[\delta, -]$	au plus près $\delta$ unités.
$[\delta, \delta]$	exactement $\delta$ unités.
$[-, -]$	distance indéterminée.

Figure 12. Interprétation d'un intervalle

À titre comparatif, le nombre de spécifications, par exemple, de la relation  $B \text{ left\_shift } A$ , dans un modèle avec une distance fixe est de trois :

$B \text{ left\_shift } (d) A$       B est décalé par la gauche de  $d$  unités par rapport à A  
 $B \text{ left\_shift } (0) A$       B et A sont alignés par la gauche.  
 $B \text{ left\_shift } (-) A$       B est décalé par la gauche par rapport à A.

Avec une distance flexible, il est de six :

$B \text{ left\_shift } ([\delta^1, \delta^2]) A$       B est décalé de A au plus près  $\delta^1$  unités et au plus loin  $\delta^2$  unités par la gauche.  
 $B \text{ left\_shift } ([\delta, -]) A$       B est décalé de A au plus près  $\delta^1$  unités par la gauche.  
 $B \text{ left\_shift } ([-, \delta]) A$       B est décalé de A au plus loin  $\delta^2$  unités par la gauche.  
 $B \text{ left\_shift } ([\delta, \delta]) A$       B est décalé de A exactement  $\delta$  unités par la gauche.  
 $B \text{ left\_shift } ([0, 0]) A$       B et A sont alignés par la gauche.  
 $B \text{ left\_shift } ([-, -]) A$       B est décalé par la gauche par rapport à A.

### 5.2.1.2 Choix des relations

Les relations du modèle, données par la figure 13, sont au nombre de 19. Pour simplifier la phase d'édition nous les avons regroupées en quatre classes en fonction de leur sémantique : positionnement qui regroupe les 9 relations définies par Papadias et al (à l'exception de la relation *même\_position* qui peut être exprimée par les autres relations), décalage, répartition et centrage.

#### - Relations de positionnement

Elles traduisent l'emplacement du média secondaire dans le voisinage du média référence. Suivant la valeur de la distance, ces relations expriment trois types d'emplacement : un alignement d'un bord du média secondaire avec le bord opposé du média référence si la distance est nulle, un espacement entre les médias si la distance est positive ou un recouvrement des médias si la distance est négative.

#### - Relations de décalage

Ces relations expriment un décalage entre deux bords identiques de deux médias. Chaque relation peut générer trois emplacements différents en fonction de la valeur de la distance : un décalage dans le sens de la relation, un alignement ou un décalage dans le sens inverse de la relation.

### - Relations de répartition

Elles disposent les médias de façon à ce que la distance entre eux soit la même. À l'opposé des relations précédentes qui sont binaires, ces relations peuvent être spécifiées entre un nombre illimité de médias. Là aussi nous définissons trois emplacements possibles : un espacement entre les médias si la distance est positive, une contiguïté si la distance est nulle ou un recouvrement deux à deux des médias si la distance est négative.

### - Relations de centrage

Ces relations placent les médias en relation sur le même axe médian.

Relation spatiale	Représentation graphique	Interprétation
$B \text{ left } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à gauche de <b>A</b> et son bord supérieur est au plus près $\delta_1^2$ unité et $\delta_2^2$ unité au dessous du bord supérieur de <b>A</b> .
$B \text{ above } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^2$ unité et au plus loin $\delta_2^2$ unité au dessus de <b>A</b> et son bord gauche est au plus près $\delta_1^1$ unité et $\delta_2^1$ unité à droite du bord gauche de <b>A</b> .
$B \text{ right\_top } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à droite de <b>A</b> et au plus près $\delta_2^1$ unité et $\delta_2^2$ unité au dessus de <b>A</b> .
$B \text{ left\_top } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à gauche de <b>A</b> et au plus près $\delta_2^1$ unité et $\delta_2^2$ unité au dessus de <b>A</b> .
$B \text{ right } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à droite de <b>A</b> et son bord supérieur est au plus près $\delta_2^1$ unité et $\delta_2^2$ unité au dessous du bord supérieur de <b>B</b> .
$B \text{ below } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_2^1$ unité et au plus loin $\delta_2^2$ unité au dessous de <b>A</b> et son bord gauche est au plus près $\delta_1^1$ unité et $\delta_2^1$ unité à droite du bord gauche de <b>A</b> .
$B \text{ right\_bottom } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à droite de <b>A</b> et au plus près $\delta_2^1$ unité et $\delta_2^2$ unité au dessous de <b>A</b> .
$B \text{ left\_bottom } (\#_x, \#_y) A$		<b>B</b> se situe au plus près $\delta_1^1$ unité et au plus loin $\delta_2^1$ unité à gauche de <b>A</b> et au plus près $\delta_2^1$ unité et $\delta_2^2$ unité au dessous de <b>A</b> .
$B \text{ right\_shift } (\#_x) A$		Le bord droit de <b>B</b> est décalé d'au moins $\delta_1^1$ et au plus $\delta_2^1$ du bord droit de <b>A</b> .
$B \text{ left\_shift } (\#_x) A$		Le bord gauche de <b>B</b> est décalé d'au moins $\delta_1^1$ et d'au plus $\delta_2^1$ du bord gauche de <b>A</b> .
$B \text{ top\_shift } (\#_y) A$		Le bord supérieur de <b>B</b> est décalé d'au moins $\delta_2^1$ et au plus $\delta_2^2$ du bord supérieur de <b>A</b> .
$B \text{ bottom\_shift } (\#_y) A$		Le bord inférieur de <b>B</b> est décalé d'au moins $\delta_2^1$ et au plus $\delta_2^2$ du bord inférieur de <b>A</b> .


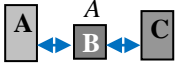


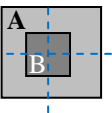
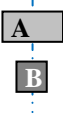

$B$ vertical_repartition ( $\#_y$ ) $A$		Les objets <b>A</b> , <b>B</b> et <b>C</b> sont espacés d'au moins $\delta^2_1$ et d'au plus $\delta^2_2$ sur l'axe vertical.
$B$ horizontal_repartition ( $\#_x$ ) $A$		Les objets <b>A</b> , <b>B</b> et <b>C</b> sont espacés d'au moins $\delta^1_1$ et d'au plus $\delta^1_2$ sur l'axe horizontal.
$B$ right_horizontal_Centring ( $\#_x$ ) $A$		<b>B</b> et <b>A</b> sont centrés horizontalement et <b>B</b> se situe au plus près $\delta^1_1$ unité et au plus loin $\delta^1_2$ unité à droite de <b>A</b> .
$B$ left_horizontal_Centring ( $\#_x$ ) $A$		<b>B</b> et <b>A</b> sont centrés horizontalement et <b>B</b> se situe au plus près $\delta^1_1$ unité et au plus loin $\delta^1_2$ unité à gauche de <b>A</b> .
$B$ total_centring $A$		<b>B</b> et <b>A</b> sont centrés horizontalement et verticalement.
$B$ bottom_vertical_centring ( $\#_y$ ) $A$		<b>B</b> et <b>A</b> sont centrés verticalement et <b>B</b> se situe au plus près $\delta^1_2$ unité et au plus loin $\delta^2_2$ au dessous de <b>A</b> .
$B$ top_vertical_centring ( $\#_y$ ) $A$		<b>B</b> et <b>A</b> sont centrés verticalement et <b>B</b> se situe au plus près $\delta^1_2$ unité et au plus loin $\delta^2_2$ au dessus de <b>A</b> .

Fig 13. Relations spatiales du modèle proposé

Avec :

- Le caractère  $\#_x$  représente l'intervalle de la distance sur l'axe des X.
- Le caractère  $\#_y$  représente l'intervalle de la distance sur l'axe des Y.

Le nombre de spécifications possibles avec une relation à une seule distance est donné par la combinaison des valeurs (nulle, positive, négative et non spécifiée) des bords de son intervalle, ce qui nous donne  $2^4$  combinaisons possibles comme le montre le tableau suivant :

	$\delta^1_1$	$\delta^1_2$	Interprétation
(1)	0	0	La distance est nulle.
(2)	0	> 0	Au plus près 0 et au plus loin $\delta^1_2$
	0	< 0	Impossible ( $\delta^1_2$ doit être supérieur à $\delta^1_1$ )
(3)	0	-	Au plus près 0
	> 0	0	Impossible ( $\delta^1_2$ doit être supérieur à $\delta^1_1$ )
(2)	> 0	> 0	Au plus près $\delta^1_1$ et au plus loin $\delta^1_2$
	> 0	< 0	Impossible ( $\delta^1_2$ doit être supérieur à $\delta^1_1$ )
(3)	> 0	-	Au plus près $\delta^1_1$
	< 0	0	Recouvrement
	< 0	> 0	Possibilité de recouvrement
(4)	< 0	< 0	Recouvrement
(4)	< 0	-	Possibilité de recouvrement
(5)	-	0	Au plus loin 0
(5)	-	> 0	Au plus loin $\delta^1_2$
(4)	-	< 0	Recouvrement d'au moins $\delta^1_2$
(6)	-	-	Aucune préférence pour la distance.

Fig 14. Spécifications possibles avec une distance

Par ailleurs, nous remarquons que 03 de ces combinaisons sont impossibles à spécifier car  $\delta^1_2$  doit être supérieur à  $\delta^1_1$  ce qui nous laisse 13 combinaisons que nous avons regroupé en 06 spécifications selon la sémantique qu'elles expriment (distance nulle, distance dans un intervalle, au plus près, au plus loin, recouvrement et aucune préférence).

Ainsi, le nombre de spécifications avec une relation à deux distances est obtenu en combinant le nombre de spécification sur l'axe horizontal (06) et le nombre de spécification sur l'axe vertical (06), ce qui nous donne  $6^2$  (36) spécifications possibles.

Pour le critère de puissance d'expressivité, on retrouve dans le modèle les 169 relations directionnelles requises pour l'emplacement à travers les relations de positionnement. Dans l'annexe 1, nous donnons pour chacune des 169 relations son équivalente dans notre modèle. La définition de la distance flexible a augmenté d'une manière considérable le degré d'expressivité du modèle. À titre comparatif, la figure 15 donne le nombre de spécifications possibles dans notre modèle et dans un modèle à distance fixe.

Relation spatiale	Nombre de spécifications distance fixe	Nombre de spécifications distance flexible
<i>left</i>	05	36
<i>right</i>	05	36
<i>above</i>	05	36
<i>below</i>	05	36
<i>right_top</i>	05	36
<i>right_bottom</i>	05	36
<i>left_top</i>	05	36
<i>left_bottom</i>	05	36
<i>left_shift</i>	03	06
<i>top_shift</i>	03	06
<i>right_shift</i>	03	06
<i>bottom_shift</i>	03	06
<i>vertical_repartition</i>	03	06
<i>horizontal_repartition</i>	03	06
<i>Right_vertical_Centring</i>	03	06
<i>Left_vertical_Centring</i>	03	06
<i>Bottom_horizontal_centring</i>	03	06
<i>Top_horizontal_centring</i>	03	06
<i>Total_centring</i>	01	01
<b>Total</b>	<b>71</b>	<b>349</b>

Fig 15. Comparaison du degré d'expressivité.

Enfin, la définition des relations de décalage permettent à l'auteur d'exprimer des spécifications qui mettent en relations les mêmes bords de deux médias, et ce en complément aux relations de positionnement qui mettent en relations uniquement les bords opposés de deux médias.

### 5.2.2 Précision des emplacements

En plus du choix des relations et de la distance, la modélisation du voisinage d'un média est un paramètre à considérer dans le degré de précision d'un emplacement. En effet, l'auteur devrait pouvoir placer un média *B* dans le plus large voisinage possible d'un média *A*. Pour cela, nous avons retenu le modèle de A.U. Frank [32], jugé le plus complet pour une forme rectangulaire d'un média, où le voisinage est défini par huit cadrans (nord, sud, est, ouest, nord-est, nord-ouest, sud-est et sud-ouest), comme le montre la figure 16.

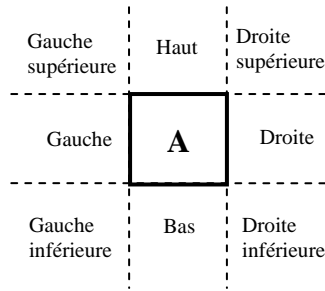


Fig 16. Voisinage d'un média.

Ainsi, l'auteur peut placer un média  $B$  dans tout le voisinage d'un média  $A$  avec les écarts voulus entre les bords des médias. Même le chevauchement sur deux cadrans adjacents est possible, la relation  $B \text{ right\_top} (d_{fx}, d_{fy}) A$ ,  $d_{fy} < 0$ , par exemple, permet de placer  $B$  sur deux cadrans du voisinage de  $A$  (figure 17).

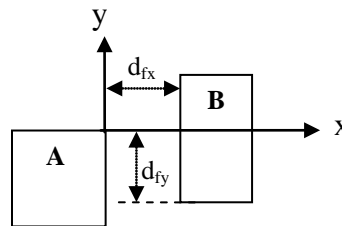


Fig 17. Placement d'un média sur deux cadrans

### 5.2.3 Simplicité d'utilisation et intuitivité

Grâce au regroupement des relations en classes, l'auteur retrouve facilement les relations qu'il désire exprimer. Les relations de répartition et de centrage ont été définies afin de donner plus de simplicité et d'intuitivité au modèle et ce malgré qu'elles peuvent être exprimées, moyennant une certaine complexité, avec les autres relations.

### 5.2.4 Spécification du recouvrement désiré

Comme mentionné, le recouvrement spatial des médias ne doit pas être automatiquement considéré comme une incohérence. Sans définir des relations spécifiques au recouvrement désiré, et par là même réduire le nombre de relations du modèle, l'auteur peut l'exprimer grâce à la distance négative. La relation  $B \text{ right} ([-5, -5], [0, 0]) A$  par exemple (figure 18) permet à  $B$  de recouvrir  $A$  par la droite de 5 unités.

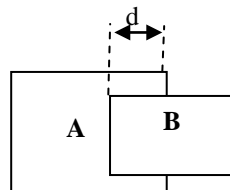


Fig 18. Spécification du recouvrement désiré

Toutefois, certaines valeurs négatives des distances peuvent générer des configurations d'emplacements autres que celles spécifiées par l'auteur. Par exemple, dans la relation  $\text{left\_bottom} (d_{fx}, d_{fy})$ , représentée dans la figure 19.a, si  $d_{fx}$  et  $d_{fy}$  sont négatives et que  $|d_{fx}| \geq A.\text{largeur}$  et  $|d_{fy}| \geq A.\text{hauteur}$ , nous aurons la configuration de la figure 19.b qui correspondrait à la relation  $\text{right\_top} (d_{fx}, d_{fy})$ .

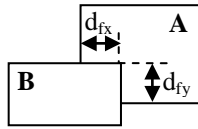


Fig 19a. *left\_bottom* ( $d_{fx}$ ,  $d_{fy}$ ),  $d_{fx}, d_{fy} < 0$

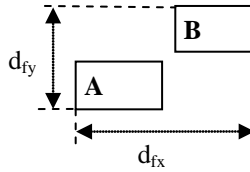


Fig 19b. *right\_top* ( $d_{fx}$ ,  $d_{fy}$ ),  $d_{fx}, d_{fy} < 0$

Fig 19. Génération de configurations autres que celles spécifiées

Afin de résoudre ce problème, nous posons la contrainte suivante sur les valeurs de  $d_{fx}$  et  $d_{fy}$  :  $|d_{fx}| \leq \text{média-référence.largeur}$  et  $|d_{fy}| \leq \text{média-référence.hauteur}$ .

## 6. Conclusion

Dans ce chapitre, nous avons présenté les principaux aspects liés à la modélisation de la spécification des relations temporelles et spatiales dans un document multimédia interactif. En partant des critères recherchés pour un modèle de spécification de relations, nous avons proposé deux modèles où les objectifs attendus sont :

1. D'allier, au mieux, simplicité d'utilisation et degré d'expressivité : En se basant sur une approche à base de relations, la spécification ne nécessite aucun préalable de profil ou des connaissances particulières. Le modèle temporel permet la spécification des 29 relations requises pour le multimédia avec un ensemble simplifié de 13 relations. Le concept de délai flexible, qui nous a permis d'introduire les expressions au plus tôt et au plus tard, a conféré au modèle un degré d'expressivité relativement élevé comparativement aux modèles existants et l'ajout des trois relations d'Allen lui assure une plus grande intuitivité.  
Dans le modèle spatial, les 19 relations définies permettent non seulement l'expression des 169 relations requises pour l'emplacement, mais permettent également des spécifications qui nécessitent plus d'une relation, voire impossibles, dans les modèles existants. L'ensemble des relations conjugué avec la distance flexible a permis d'introduire les expressions au plus près et au plus loin, expressions qu'aucun modèle actuel n'offre.
2. De permettre l'expression de l'indéterminisme : le modèle temporel permet, non seulement, l'expression de l'indéterminisme mais offre également, dans certains cas, des informations utiles pour sa gestion lors des phases d'analyse et de présentation,
3. D'augmenter la précision des emplacements des médias : grâce toujours à la distance flexible et à la définition d'un voisinage qui enveloppe tout le contour d'un média, le modèle spatial offre une grande précision d'emplacement et permet même l'emplacement d'un média sur deux cadrans adjacents.
4. De permettre la spécification du recouvrement désiré : sans définir des relations spécifiques, l'utilisation de la distance négative permet d'exprimer le recouvrement désiré.

Enfin, grâce aux concepts du délai et de la distance flexibles, les spécifications de l'auteur peuvent être moins contraignantes ce qui donne plus de chance à un média d'être *placé* temporairement ou spatialement et ainsi d'éviter à l'auteur des re-spécifications.

Dans le chapitre qui suit, nous allons abordés les mécanismes d'analyse et de synthèse des modèles.

# Chapitre 5

## Analyse et synthèse des spécifications

### 1. Introduction

Comme mentionné, l'écriture du scénario d'un document multimédia interactif ne peut en général suivre une logique d'enchaînement linéaire car souvent des choix déjà faits peuvent ou doivent être revus. D'autre part, les spécifications sont souvent cause de contradictions. Par conséquent, des mécanismes à même de gérer et de résoudre ces conflits doivent être prévus.

Au cours du précédent chapitre, nous avons choisi de fournir un formalisme à base de relations pour le langage de spécification et un formalisme à base de contraintes pour la vérification de la cohérence (analyse) et la génération de solutions d'*emplacements* temporel et spatial (synthèse).

De par le nombre de travaux menés dans le domaine des systèmes de contraintes, nous avons fait le choix d'étudier des techniques de résolution de contraintes pour mettre en œuvre les services de cohérence et de synthèse plutôt que de chercher des solutions ad hoc.

Le présent chapitre est organisé de la manière suivante. Dans sa deuxième section une présentation sommaire des systèmes de contraintes est faite. Elle est suivie par la présentation d'un ensemble de caractéristiques que nous jugeons nécessaires pour un résolveur de contraintes dans un environnement d'édition et de présentation de documents multimédia interactifs, cet ensemble va nous servir de base pour le choix d'un résolveur. Dans la quatrième section, nous présentons les principales approches de résolution de contraintes en les confrontons aux caractéristiques retenues. La cinquième section est consacré au résolveur Cassowary que nous avons adopté, après sa présentation, nous montrons comment qu'il répond aux caractéristiques recherchées pour le domaine de l'EPDMI. La section 6 conclut ce chapitre.

### 2. Présentation des systèmes de contraintes

#### 2.1 Contrainte

Dans le domaine de l'EPDMI, une contrainte est une relation ou une information intrinsèque d'un média qui est formulée à l'aide d'une ou de plusieurs variables. Par exemple, la contrainte *Bord-gauche + largeur = Bord-droite* exprime la relation que le bord droit d'un média doit être espacé de son bord gauche d'un nombre d'unités égal à *largeur*.

Résoudre ou satisfaire une contrainte consiste à trouver, pour chaque variable, une affectation de valeur qui satisfait la condition définie par celle-ci.

#### 2.2 Système de contraintes

Un système de contraintes consiste en un ensemble de variables et en un ensemble de contraintes qui limite les combinaisons de valeurs pour ces variables. Plus formellement, un système de contraintes est défini par un triplet  $\{V, D, C\}$  où :

$V = \{V_1, V_2, \dots, V_n\}$  : est un ensemble fini de variables.

$D = \{D_1, D_2, \dots, D_n\}$  : représente l'ensemble des domaines de valeurs, pour  $i$  dans  $[1, n]$ ,  $D_i$  est le domaine des valeurs possibles pour la variable  $V_i$ .

$C = \{C_1, C_2, \dots, C_n\}$  : est un ensemble de contraintes entre les variables, c'est-à-dire un sous

ensemble de  $D_1 \times D_2 \times \dots \times D_n$ . Chaque contrainte peut être définie en intention par une équation ou en extension par l'ensemble des *tuples* qui la satisfont.

Une solution pour un tel système est une instantiation des variables de  $V$  qui satisfait toutes les contraintes de  $C$ .

### 2.3 Représentation graphique

Un système de contrainte est communément représenté par un graphe non orienté. Le graphe de la Figure 1 correspond au système  $\{C1 : C = A + B, C2 : E = C + D \text{ et } C3 : F = B\}$ . Les cercles représentent les variables et les carrés noirs les contraintes. Les arêtes symbolisent les liens *variables-contraintes*, c'est-à-dire l'appartenance d'une variable à une contrainte. Un tel graphe est appelé *graphe de contrainte* du système.

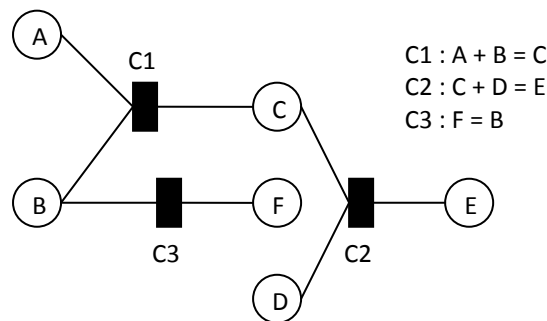


Fig.1 Graphe de contraintes d'un système

### 3. Caractéristiques recherchées pour un résolveur dans le domaine de l'EPDMI

Le choix d'un résolveur se fera en fonction d'un ensemble de caractéristiques, que nous avons retenu pour le domaine de l'EPDMI, elles se présentent comme suit :

1. La gestion des équations et des inéquations : Dans le domaine de l'EPDMI, les contraintes ne se présentent pas uniquement sous la forme d'équations mais également sous forme d'inéquations, on peut par exemple spécifier qu'un média A est à gauche de 5 unités d'un média B. Plusieurs méthodes de résolution ne gèrent que des contraintes de type fonctionnel (équation), c'est à dire chaque variable est déterminée de façon unique en fonction des autres variables de la contrainte. Or, les contraintes d'inégalité ne sont pas fonctionnelles. L'approche doit pouvoir gérer les deux types de contraintes.
2. L'incrémentalité : Dans un contexte d'environnement interactif, l'efficacité de la méthode de résolution est un facteur très important. Cette efficacité se traduit généralement par l'utilisation de techniques incrémentales (elles profitent de la solution courante pour calculer la nouvelle solution). On a en particulier deux cas de figure :
  - À l'édition d'un média, on peut vouloir ajouter, supprimer ou modifier des spécifications et donc on a besoin d'avoir une résolution rapide, ce qui est possible si l'on ne fait que réutiliser au maximum la solution précédente.
  - Quand on déplace un média directement (par exemple, avec une souris), on représente cette interaction en tant qu'une contrainte qui décrit le déplacement d'une partie du média vers la position de la souris (x et y). Dans ce cas, on doit satisfaire le même ensemble de contraintes avec une différence d'une seule contrainte (celle de la position de la souris).

Là où un résolveur non-incrémental reconsidérerait l'ensemble des contraintes, un résolveur incrémental ne réévalue que les contraintes affectées par la dernière perturbation.

3. Le maintien de la solution : Suite à une perturbation du système pour laquelle aucune solution n'a été trouvée, l'approche doit pouvoir revenir à la solution précédente.
4. Le contrôle du choix d'une solution : Une propriété importante des solveurs concerne leur comportement lorsque plusieurs solutions sont possibles (système sous-contraint). Il en résulte alors la nécessité de pouvoir contrôler le choix de la solution. Dans notre cas, celle-ci pourrait être la *prise en compte des préférences de l'auteur* : il peut vouloir, par exemple, donner la priorité à la valeur *pref* d'une durée. Le système de contraintes devra permettre de prendre en compte de telles préférences.
5. La complétude : lorsqu'une solution existe pour le système, l'approche doit pouvoir la trouver.
6. La gestion des cycles : les spécifications cycliques sont des situations très fréquentes dans le domaine de l'EPDMI. Prenons l'exemple de deux contraintes C1 et C2 : {C1:  $A=B+C$ ; C2:  $C = B$ ). Trois méthodes à sortie unique sont associées à la première contrainte:  $A:=B+C$ ,  $B:=A-C$  et  $C:=A-B$ , deux méthodes à sortie unique sont associées à la deuxième contrainte:  $C:=B$  et  $B:=C$ .

Supposons que l'on parte de la solution (A,4), (B,2) et (C,2) et que l'on modifie A avec la valeur 10. Pour maintenir C1, le résolveur doit choisir une méthode où A est une variable d'entrée, soit  $B:=A-C$  soit  $C:=B-A$ , puis propager la nouvelle modification. S'il choisit  $B:=A-C$ , il doit maintenir la contrainte C2 en exécutant une méthode où B est en entrée soit  $C:=B$ . La variable C étant modifiée, le résolveur doit de nouveau maintenir la contrainte C1, ce qui crée un cycle. Une solution serait d'arrêter la propagation après un seul parcours de boucle, mais on obtient alors une solution incohérente: (A,10), (B, 8) et (C,8). Si le résolveur avait choisi au départ d'exécuter la méthode  $C:=A-B$  le résultat aurait été identique. L'approche doit pouvoir gérer les contraintes cycliques.

7. La directionnalité : Ce critère caractérise la capacité d'un résolveur à gérer des contraintes unidirectionnelles (one-way) ou multidirectionnelles (multi-way). Les contraintes unidirectionnelles possèdent une méthode unique de résolution, c'est à dire qu'elles ne peuvent être maintenues que d'une seule façon. Les contraintes multidirectionnelles possèdent au contraire plusieurs méthodes de résolution. Dans notre contexte, il est nécessaire que les contraintes soient multidirectionnelles. Un exemple de ces deux types de contraintes est illustré par la Figure 2. Dans la partie supérieure, le média A et le média B sont alignés sur leur bord gauche par une contrainte unidirectionnelle. L'unique méthode de résolution spécifie que B est aligné sur A. Si l'on déplace A sur l'axe horizontal, alors B suit le mouvement de A (la contrainte est maintenue). Si l'on déplace B sur l'axe horizontal, A reste au contraire immobile. La contrainte d'alignement n'est plus respectée et elle est retirée. Dans la partie inférieure, A et B sont alignés par une contrainte multidirectionnelle avec deux méthodes de résolution : soit que A est aligné sur B, soit que B est aligné sur A. Le résultat obtenu est que le déplacement de n'importe lequel des deux entraîne un déplacement analogue de l'autre. La contrainte reste constamment maintenue.

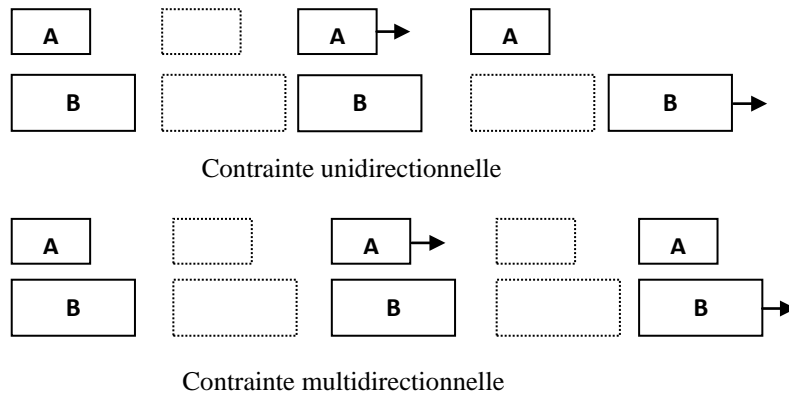


Fig 2. Contrainte unidirectionnelle et multidirectionnelle

8. L'intégration des trois aspects : L'approche retenue pour le traitement des contraintes temporelles, spatial et du recouvrement doit être unique.
9. Les performances temporelles : Dans un environnement interactif le système se doit d'avoir des performances temporelles acceptables.

## 4. Les différentes approches de résolution des systèmes de contraintes

Dans la littérature, il existe trois grandes familles d'approches pour la résolution des systèmes de contraintes. Dans ce qui suit, nous allons donner les principes de leurs fonctionnements ainsi qu'une critique de leurs avantages et insuffisances, par rapport aux critères retenus, qui va nous servir pour le choix de l'une d'elles.

### 4.1 Les approches locales

Dans une approche locale [33][40][78][79][87][99], les contraintes sont traitées individuellement, de façon séquentielle. Elles n'utilisent que des informations qui leur sont disponibles immédiatement, c'est-à-dire les valeurs des variables qu'elles contraignent. Une contrainte peut être vue comme responsable d'elle même, elle a une certaine autonomie et est capable d'action et de réaction. Les résolveurs procèdent en deux étapes. Ils construisent dans un premier temps un graphe de contraintes où chaque variable du problème est représentée par un cercle, chaque contrainte est représentée par un carré, et chaque arc représente l'appartenance d'une variable à une contrainte.

Dans un deuxième temps, à chaque contrainte est associée un ensemble de méthodes qui explicitent comment répercuter la modification d'une variable sur les autres variables présentes dans la contrainte.

Une méthode consiste en zéro ou plusieurs variables en entrée, en une ou plusieurs variables en sortie et en une suite d'instructions qui calculent la ou les variables de sortie en fonction des variables d'entrée de manière à satisfaire la contrainte. Sur l'exemple de la contrainte  $CI : C=A+B$  (fig 3.), on peut lui associer trois méthodes de résolution, chacune stipulant qu'une variable est calculée en fonction des deux autres:  $A:=C-B$  (fig3.a),  $B:=C-A$  (fig3.b) et  $C:=A+B$  (fig3.c). Si la valeur de  $B$  ou  $C$  est modifiée, alors la contrainte peut être maintenue en exécutant la première méthode.

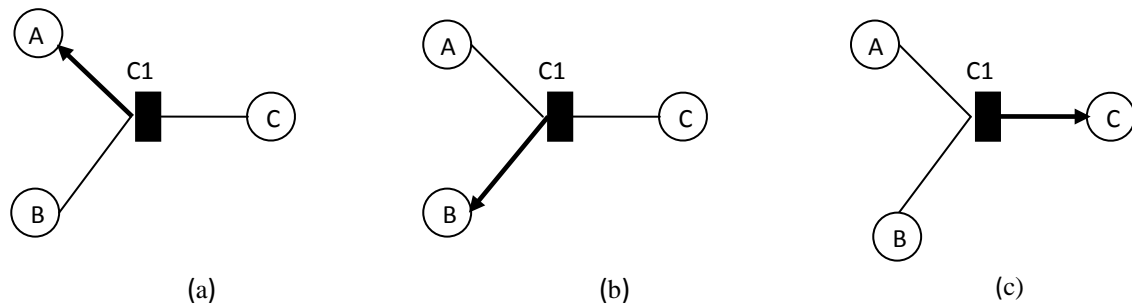


Fig.3 Les différentes méthodes de résolution possible

Une des méthodes largement utilisée dans les applications interactives est la propagation locale. Elle se décline sous deux variantes, la propagation locale par valeurs et la propagation locale par degré de liberté [100]. Partant d'une solution perturbée, la propagation locale est une technique qui rétablit la cohérence d'un système de contraintes en sélectionnant, puis en exécutant une séquence finie de méthodes. L'exécution d'une méthode satisfait la contrainte associée en modifiant la valeur de ses variables de sortie. Les valeurs modifiées entraînent la violation d'autres contraintes et, par propagation, l'exécution d'autres méthodes pour les satisfaire et ce, de manière récursive. L'exécution d'une méthode dont les variables de sortie ne sont reliées à aucune autre contrainte ne provoque pas de propagation.

Cette famille se caractérise par sa simplicité de mise en œuvre, la non restriction des types de domaines des contraintes et ses performances temporelles très acceptables. Toutefois, elle ne satisfait pas tous les critères que nous avons retenus, notamment :

- Son incomplétude, c'est-à-dire que le système peut ne pas trouver de solution alors qu'il peut en existe une. Cela est lié par exemple à une orientation du graphe vers une valuation des variables sans solution.
- L'impossibilité de supporter des cycles dans le graphe de contraintes. Par conséquent de tels systèmes rejettent les contraintes qui introduisent des cycles. Dans notre contexte, les systèmes de contraintes que l'on manipule engendrent des graphes de contraintes fortement cycliques. Des travaux ont été réalisés pour supporter des graphes cycliques. Dans Skyblue [88] par exemple, le traitement des cycles est déferé vers un résolveur spécialisé. Cependant ces techniques ne semblent pas adaptées aux graphes fortement cycliques du fait que toute la résolution du problème est déferée au résolveur spécifique, et de ce fait on ne tire plus profit de la méthode locale.
- La manipulation de contraintes uniquement fonctionnelles. Une contrainte à  $n$  variables est fonctionnelle, si lorsque l'on fixe  $n-1$  variables, la dernière est déterminée de façon unique. Dans notre contexte, les contraintes qui limitent l'intervalle de validité d'une variable ne sont pas fonctionnelles.
- l'impossibilité de traiter des inéquations.
- L'absence de remise en cause du plan calculé au début d'une opération d'édition lors d'une succession d'opérations d'édition identiques [100]. Par exemple, tout au long du déplacement d'un objet dans la vue de présentation ce seront les mêmes déformations qui seront appliquées (Fig 4.). Si nous avons trois objets A, B et C placés les uns à côté des autres, et que nous déplaçons l'objet A vers la droite, on aimerait, dans un premier temps, retailler B, et dans un deuxième temps C, ou alors de répartir la déformation de manière équivalente sur les deux. Dans le cas d'un résolveur local, on aura B qui prendra la taille minimale autorisée par les contraintes, et après tout déplacement sera impossible. Il faudra recalculer un nouveau plan et ajouter une nouvelle contrainte pour dire que B n'est plus modifiable, et ainsi forcer le résolveur à déformer C. Pour éviter

cela, il faut donc changer le plan à chaque fois que cela est nécessaire. Le résolveur devrait permettre de changer localement le plan de manière intrinsèque.

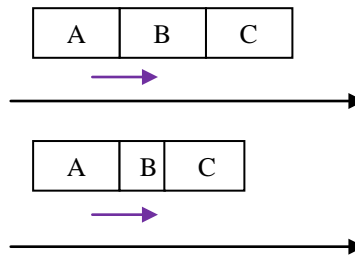


Fig 4. Absence de remise en cause du plan

Malgré leurs limitations, on peut noter que cette technique est très employée dans les applications graphiques. De nombreux systèmes ont utilisé cette technique à l'instar de Indigo [14], UltraViolet [16], SketchPad [22], ICOLA [35], ThingLab [63], DeltaBlue [87] et SkyBlue [88].

## 4.2 Les approches globales

Dans les approches globales, toutes les contraintes du système sont considérées de façon simultanée. Elles parcourent toutes les valeurs possibles de toutes les variables jusqu'à trouver une solution (quand elle existe.) Il existe de nombreux algorithmes [103] qui utilisent cette approche et qui se différencient de part leur manière de parcourir l'ensemble des valeurs. On en distingue essentiellement ceux qui utilisent l'algorithme du backtrack (retour arrière) [10][24][76][111].

### 4.2.1 Algorithme de backtrack [10]

Nous rappelons simplement les caractéristiques essentielles de cet algorithme. Dans l'état initial, aucune variable n'est affectée. À une étape donnée, l'ensemble des variables est divisé en deux groupes, les variables affectées et les variables non affectées. La solution donnée par les variables affectées est cohérente par rapport aux sous-ensembles de contraintes qui ne portent que sur ces variables. La prochaine étape consiste à affecter une nouvelle variable, le système choisit une variable et une valeur dans l'ensemble de celles qui sont possibles, et vérifie la cohérence. Si le système est cohérent, il passe à l'étape suivante, si le système est incohérent, il essaie une autre valeur pour la variable considérée. Si toutes les valeurs ont été considérées, il désaffecte une des variables affectées, et repart à l'étape n-1.

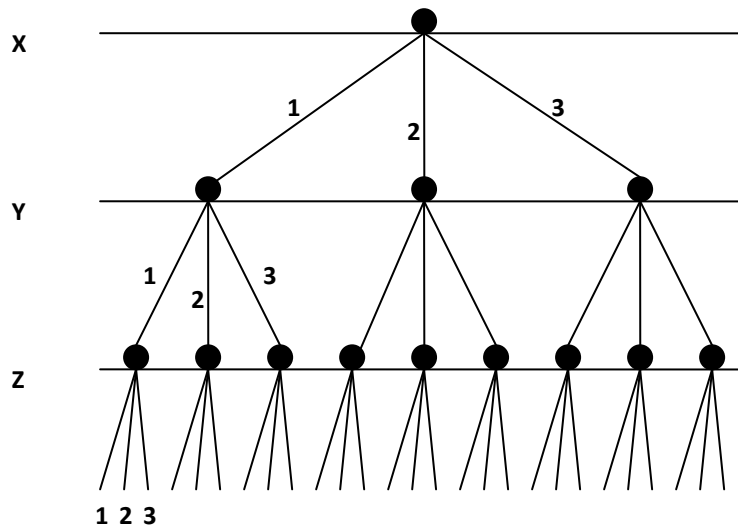


Fig.5 Arbre de recherche

Dans l'exemple de la Figure 5, on peut voir un arbre de recherche dans le cas où nous aurions des contraintes qui portent sur trois variables X, Y et Z (chacune étant définie dans l'intervalle [1...3]). Le système dans un premier temps évalue X, en lui affectant par exemple la valeur 1, dans un deuxième temps il évalue Y puis Z. Si cette évaluation ne satisfait pas le système de contraintes, il choisit une nouvelle valeur pour Z. Si aucune des valeurs de Z ne permet de satisfaire l'ensemble de contraintes, le système choisit une nouvelle valeur pour Y, et ainsi de suite. Il existe des heuristiques qui permettent d'améliorer et de personnaliser cet algorithme de manière à orienter la recherche de solutions (par exemple les heuristiques portant sur le choix de la variable à remettre en cause lors d'un retour arrière).

Les avantages de cette technique sont sa complétude, on trouve toujours la solution lorsqu'elle existe, la possibilité pour l'utilisateur de contrôler la recherche de solution en cas de réponses multiples et la prise en compte des contraintes cycliques.

Ses inconvénients sont l'obligation de travailler sur des domaines finis, or nos systèmes de contraintes manipulent des domaines infinis et le coût généralement très grand de cette technique, qui procède par énumération, lorsque le domaine des variables est trop grand.

### 4.3 Approches basées sur le simplexe

Ces approches se basent sur la méthode du simplexe, qui a été proposée par Dantzig vers les années 40 [23]. Cette méthode simple, robuste et efficace permet de résoudre avec succès des problèmes linéaires comportant plusieurs dizaines de milliers de variables et de contraintes. De nombreuses variantes ont été réalisées de manière à améliorer les performances temporelles de l'approche et dans un deuxième temps à définir une fonction d'optimisation ( $f$ ) appropriée.

Un problème linéaire se pose ainsi :

Trouver les valeurs des variables  $x_1, x_2, \dots, x_n$  qui optimisent la fonction objective  $c_1x_1 + c_2x_2 + \dots + c_nx_n$  et vérifient les contraintes de minimisation (ou de maximisation) :

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\
 \dots & \\
 a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n &\leq b_k \\
 \dots & \\
 a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \quad x_1 \geq 0; x_2 \geq 0; \dots x_n \geq 0
 \end{aligned}$$

Avec :  $a_{11}, \dots, a_{mn}, b_1, \dots, b_m, c_1, \dots, c_n$ , connus.

Ces approches se révèlent mieux adaptées au contexte des applications graphiques notamment pour le placement spatial. En effet, contrairement aux approches locales qui n'acceptent pas les systèmes de contraintes qui présentent des cycles ou des inégalités, celles basées sur le simplexe les gèrent efficacement. De plus, leurs performances temporelles sont proches de celles des approches locales contrairement au backtrack qui est très long et mal adapté à notre contexte.

Toutefois, trois aspects sont mal gérés par les différentes implémentations de cet algorithme :

- L'incrémentalité : la solution courante est calculée indépendamment de la solution précédente.
- La prise en considération des variables négatives : Pour les besoins des applications multimédia, on doit pouvoir manipuler des variables négatives alors que l'algorithme du simplexe ne manipule que des variables positives.
- La définition de la fonction objective appropriée : Dans l'algorithme du simplexe, la fonction objective à optimiser doit être linéaire ; or les comparateurs utilisés pour le multimédia sont tous non linéaires.

Plusieurs travaux ont été menés afin de surmonter ces insuffisances et d'adapter le simplexe aux systèmes graphiques notamment. Parmi ces travaux, nous citons le résolveur Cassowary. Ainsi il respecte la totalité des critères susmentionnés et de ce fait c'est lui que nous avons adopté pour notre système.

Dans ce qui suit, nous allons présenter le principe de fonctionnement de Cassowary et montrer comment qu'il satisfait aux critères fixés pour le domaine de l'EPDMI.

## 5. Le résolveur Cassowary

### 5.1 Principe de fonctionnement

Cassowary est un résolveur de contraintes qui permet de traiter des équations et des inéquations linéaires. Il a été développé à l'université de Washington par Badros et Borning [Badros98]. Il se décompose en deux phases :

#### 5.1.1 La phase de prétraitement

Elle vise à traduire le problème sous une forme ne contenant ni inégalité ni variables négatives, de manière à résoudre l'ensemble de contraintes ainsi obtenues par le simplexe :

- Les inégalités de la forme  $x \geq y$  sont traduites sous la forme  $x = s + y$ , où  $s$  est une variable non négative appelée *variable d'écart (slack)*.
- Les variables négatives  $v$  sont substituées par des équations de la forme  $e = -v$ , toutes les occurrences de  $v$  sont ensuite remplacées par  $(-e)$ . La résolution des variables  $v$  se fera dans une étape postérieure, une fois la valeur de  $e$  connue.

Ensuite le système de contraintes est partagé en deux tableaux  $C_S$ , et  $C_U$ . Le tableau  $C_S$  contiendra les variables contraintes (c'est à dire l'ensemble des variables restreint à ne pas prendre de valeurs négatives), et  $C_U$  celles qui sont non contraintes. Cette première étape a été améliorée pour rendre la deuxième phase plus efficace. Initialement toutes les équations sont placées dans le tableau  $C_S$ . Un filtrage basé sur la méthode d'élimination de Gauss-Jordan a été rajouté. Cette phase a pour but de supprimer toutes les variables non contraintes du tableau  $C_S$  pour les placer dans  $C_U$ . Pour réaliser cela, on cherche dans  $C_S$  une équation contenant une variable  $v$  non contrainte et on retire cette équation de  $C_S$ . On résout l'équation pour  $v$ , en générant une nouvelle équation  $e$  de la forme :  $eI = v$  (ou  $eI$  est une expression). On ajoute  $eI$  à  $C_U$  et on substitue alors toutes les occurrences de  $v$  dans  $C_S$ ,  $C_U$  et  $f$ .

Nous donnons un exemple simple pour illustrer cette première phase. Soit le système de contraintes suivant :

$$\begin{aligned} 2x_1 &= x_2 + x_3 \\ x_2 + 10 &\leq x_3 \\ x_2 &\leq 20 \\ x_3 &\leq 100 \end{aligned}$$

Considérons arbitrairement  $\text{Min } f = x_3 - x_2$  comme fonction objective de ce système. (Nous verrons par la suite la fonction objective réelle utilisée dans Cassowary).

En traduisant les inégalités en égalités, nous obtenons le système suivant :

$$\begin{aligned} \text{Min } f &= x_3 - x_2 \\ 2x_1 &= x_2 + x_3 \\ x_2 + 10 &= x_3 - s_1 \\ x_2 &= 20 - s_2 \\ x_3 &= 100 - s_3 \\ \text{avec } s_1, s_2, s_3 &\geq 0 \end{aligned}$$

Initialement le tableau  $C_S$  de cet exemple correspond au système de contrainte ci-dessous. Les équations contenant des variables non contraintes sont retirées de ce tableau et transférées vers  $C_U$ . A l'exemple de l'équation  $x_3 = 100 - s_3$  qui contient une variable non contrainte à savoir  $x_3$ . Pour transférer cette équation vers  $C_U$ , nous substituons toutes les occurrences de  $x_3$  par la partie droite de l'équation (i.e  $100 - s_3$ ), dans le système de contraintes précédent. Nous obtenons ainsi le système suivant :

$$\begin{aligned} \text{Min } f &= 100 - s_3 - x_2 \\ \left. \begin{aligned} x_3 &= 100 - s_3 \\ 2x_1 &= x_2 + 100 - s_3 \\ x_2 &= 90 - s_3 - s_1 \end{aligned} \right\} C_U \\ \left. \begin{aligned} x_2 &= 20 - s_2 \end{aligned} \right\} C_S \end{aligned}$$

En répétant ce processus pour les autres variables non contraintes, nous obtenons à la fin le système ci-dessous, où toutes les variables de  $C_S$  sont contraintes.

$$\begin{aligned} \text{Min } f &= 80 - s_3 + s_2 \\ \left. \begin{aligned} x_3 &= 100 - s_3 \\ x_2 &= 20 - s_2 \\ x_1 &= 60 - \frac{1}{2} s_2 - \frac{1}{2} s_3 \end{aligned} \right\} C_U \\ \left. \begin{aligned} s_1 &= 70 - s_3 + s_2 \end{aligned} \right\} C_S \end{aligned}$$

### 5.1.2 La phase de résolution

Une solution est obtenue par la méthode du simplexe classique.

Par ailleurs, Cassowary classe les contraintes suivant une hiérarchie de priorité. Les contraintes utilisées dans l'exemple que nous venons de donner sont dites *requisées*, et doivent être satisfaites impérativement. Leur non satisfaction correspond à un échec de la résolution.

En plus des contraintes requises, Cassowary définit un autre type de contraintes internes appelées *contraintes non-requisées*. Ces dernières ont un niveau de priorité inférieur à celui des contraintes requises, et de ce fait elles ne sont satisfaites qu'en dernier lieu, c'est-à-dire après les contraintes requises. Lorsqu'elles sont en contradiction avec une contrainte requise, le résolveur ne les prend pas en considération.

Il existe deux classes de contraintes non-requises :

- Les contraintes d'édition (*edit constraints*) : Elles sont utilisées pour changer la valeur d'une variable.
- Les contraintes de maintien (*stay constraints*): Elles sont utilisées pour contraindre une variable à maintenir sa valeur initiale.

Les contraintes non requises sont représentées sous forme d'équation de la forme :

$v = \alpha + \delta_v^+ - \delta_v^-$  où  $\delta_v^+$  et  $\delta_v^-$  sont des variables non négatives représentant l'écart de  $v$  par rapport la valeur désirée  $\alpha$ . Si la contrainte est satisfaite elles auront toutes les deux la valeur 0. Dans le cas contraire  $\delta_v^+$  sera positive et  $\delta_v^-$  nulle si  $v$  est trop grand par rapport à la valeur souhaitée, et vice-versa si  $v$  est trop petit. Comme le but est de minimiser les valeurs de  $\delta_v^+$  et  $\delta_v^-$ , Cassowary les met dans la fonction objective. Il utilise alors une paire de variables d'erreur, au lieu d'une seule variable, pour satisfaire la restriction de non négativité des variables qui font partie de la fonction objective.

Les contraintes non-requises sont très utiles dans le domaine des applications graphiques. En effet, les contraintes d'édition sont utilisées lors d'un déplacement d'un élément par la souris et sont satisfaites tant que ce déplacement est possible. Lorsque le déplacement n'est plus possible (lorsqu'un élément a atteint un des bords de l'écran par exemple), les contraintes d'édition associée ne sont plus satisfaites et l'élément concerné ne suit pas le mouvement de la souris. Par ailleurs, les contraintes de maintien sont intéressantes dans la mesure où leur utilisation permet aux éléments qui ne sont pas concernés par un déplacement de rester à leurs places.

## 5.2 Prise en comptes des critères pour le domaine de l'EPDMI

### 5.2.1 La gestion des équations et des inéquations

Etant basé sur l'approche du Simplexe, de ce fait Cassowary gère les équations et les inéquations.

### 5.2.2 l'incrémentalité

Dans Cassowary, cette caractéristique est définie par rapport à l'ajout et la suppression de contraintes.

#### - Ajout de contraintes

Pour ajouter une nouvelle contrainte au tableau, en premier lieu Cassowary la convertit : si c'est une inégalité en une équation par l'ajout de variables d'écart. Il utilise le tableau courant pour substituer toutes les variables de base, ce qui donne une équation de la forme  $e = c$  où  $e$  est une expression linéaire et  $c$  une constante. Si  $c$  est négative, alors il multiplie les deux cotés par  $-1$  pour qu'elle devienne positive. Si  $e$  contient une variable non contrainte, il l'utilise pour remplacer ses occurrences dans le tableau et ajouter l'équation dans la partie  $C_U$ . Sinon, il crée une variable artificielle restreinte  $a$ , ajoute l'équation  $a = e - c$  dans la partie  $C_S$  du tableau et considère l'expression  $e - c$  comme étant la fonction objective du système. Le but ici est de minimiser cette fonction. Si le minimum résultant est différent de zéro, le système n'a pas de solution et la contrainte est rejetée. Dans le cas contraire  $a$  est soit de base, soit hors base. Si  $a$  est hors base, sa colonne peut être tout simplement enlevée du tableau. Si elle est de base, la rangée où elle se trouve doit avoir la constante 0 (puisque nous pouvions atteindre la valeur 0 pour notre fonction objective qui est égale à  $a$ ). Si la rangée est simplement  $a = 0$ , le résolveur

la supprime. Sinon la rangée est de la forme  $a = 0 + bx + e$  où  $b \neq 0$ . Dans ce dernier cas Cassowary fait rentrer  $x$  dans la base en utilisant cette rangée et enlève la colonne de  $a$ .

#### - Retrait de contraintes

Après qu'une série de pivotements ait été exécutée, l'information représentée par la contrainte à retirer n'est plus contenue dans une seule rangée. Donc le résolveur a besoin d'identifier les influences de la contrainte dans le tableau pour pouvoir la retirer. Pour cela il utilise une variable appelée *marqueur*, définie à l'origine pour chaque contrainte. Il retrouve par la suite l'effet de la contrainte sur le tableau en recherchant les occurrences de ce marqueur. Pour les contraintes d'inégalités, la variable d'écart  $s$ , ajoutée pour qu'elles deviennent une égalité, peut lui servir comme marqueur, puisque à l'origine  $s$  n'est présente que dans cette équation. L'équation représentant une contrainte non-requise aura une variable d'erreur qui peut servir comme marqueur. Pour les contraintes d'égalités requises, le résolveur ajoute une variable contrainte appelée *variable factice (dummy)* à l'équation originale pour lui servir de marqueur. Cette variable ne rentre jamais dans la base, puisqu'elle doit toujours prendre la valeur 0 pour ne pas avoir d'influences sur le système.

Pour la suppression d'une contrainte, Cassowary fait rentrer la variable marqueur dans la base par une série de pivotements. Ensuite, il enlève tout simplement la rangée où se trouve le marqueur.

### 5.2.3 Le maintien de la solution

Lors d'une perturbation du système (mise à jour), Cassowary maintient la solution courante vers laquelle il peut revenir dans le cas où aucune solution n'existe pour le nouveau système. Pour cela, il retire la dernière contrainte qui a provoqué l'échec. Toutefois, dans sa logique de conception, toute opération de mise à jour introduit une seule contrainte, or dans notre cas, une relation par exemple, peut être convertie en un ensemble de contraintes. A ce niveau, nous avons étendu la solution pour pouvoir retirer un ensemble de contraintes.

### 5.2.4 Contrôle du choix de la solution

Afin de pouvoir contrôler le choix d'une solution (lorsqu'elles sont multiples), Cassowary utilise le concept de contraintes hiérarchique qui permet de classer les contraintes selon le besoin de satisfaction que l'auteur peut requérir. Elle définit pour chaque contrainte une force exprimée en termes de *poids*. Une contrainte de force *requise* par exemple doit être satisfaite. En général, une contrainte d'une force donnée domine complètement les contraintes de force moindres. Explicitement, au moment de la résolution, le système essaye de satisfaire toutes les contraintes. En cas d'échec, le système supprimera certaines contraintes non requises jusqu'à trouver une solution.

Pour cela, il est nécessaire de définir une relation d'ordre sur les contraintes et un comparateur.

Un comparateur comprend une fonction d'erreur qui mesure l'écart entre une solution proposée et la solution recherchée. Le comparateur peut être de deux types, soit local s'il considère l'erreur pour chaque contrainte à l'intérieur d'un niveau de hiérarchie, soit global s'il considère l'erreur combinée pour chacun des niveaux. Pour des systèmes manipulant des contraintes d'inégalité, il est plus approprié d'utiliser un comparateur métrique que celui de prédicat. Ainsi, on peut utiliser [4]:

- Les comparateurs *locally error better* : minimisent la distance des variables modifiées, ils favorisent les modifications locales.
- Les comparateurs *Weighted sum better* : minimisent la somme des poids des contraintes modifiées.
- Les comparateurs *Least-squares-better* : minimisent une distance entre les objets modifiés.

Cassowary utilise la métrique *locally-error-better* qui est plus permissibile car il admet plusieurs solutions aux contraintes.

### 5.2.5 La complétude

Cassowary considère toutes les contraintes de façon simultanée, il parcourt toutes les valeurs possibles de toutes les variables jusqu'à ce qu'il trouve une solution (quand elle existe.)

### 5.2.6 La gestion des cycles

De part l'utilisation de l'algorithme du simplexe, Cassowary prend en charge la gestion des cycles.

### 5.2.7 La directionnalité

Cassowary est multidirectionnel du fait qu'il gère indifféremment les équations linéaires et leurs inverses ( $x=f(y)$  et  $y=f^{-1}(x)$ ).

### 5.2.8 L'intégration des trois aspects

Cette caractéristique a été assurée du fait que nous avons opté pour une conversion en contraintes de toutes les informations du document (informations temporelles, spatiales et celle du recouvrement)

### 5.2.9 Les performances temporelles

Les performances de Cassowary sont jugées assez satisfaisantes, à titre illustratif, un document complexe à 600 contraintes et 700 variables est traité en moins de 30 ms en moyenne.

## 5.3 La fonction objective

À l'inverse de la fonction objective dans le simplexe standard qui doit être une expression linéaire, la fonction objective dans Cassowary n'est pas linéaire [4]. Cependant beaucoup de techniques ont été développées dans la communauté de recherche opérationnelle pour les manipuler. Pour des systèmes manipulant des contraintes d'inégalité, il est plus approprié d'utiliser un comparateur métrique.

Ainsi, la fonction objective du résolveur est formée comme étant la somme des poids des erreurs positives et négatives de toutes les contraintes non requises du système de contraintes. L'idée principale est que chaque contrainte non requise qui est de la forme  $x = v$  sera réécrite dans le résolveur sous forme  $x = v + \delta_v^+ - \delta_v^-$  et ajouter ainsi le terme  $c_w * (\delta_v^+ + \delta_v^-)$  à la fonction objective où  $\delta_v^+$  et  $\delta_v^-$  représentent les variables d'erreurs de la contrainte et  $c_w$  le coefficient reflétant le poids de la contrainte.

## 6. Conclusion

Dans ce chapitre, nous avons présenté brièvement la programmation par contraintes et ses caractéristiques. Après avoir recensé les caractéristiques d'un résolveur jugées nécessaires pour le domaine de l'EPDMI, nous avons étudié et confronté les trois approches de résolution de contraintes aux caractéristiques retenues. Il en est ressorti que l'approche basée sur le simplexe est celle qui répond le mieux à l'ensemble des critères hormis ceux de l'incrémentalité, de la prise en compte des variables négatives et de la définition de la fonction objective. Dans cette famille, l'algorithme Cassowary vient pour pallier à ces insuffisances grâce au simplexe incrémentale et à la linéarisation de la fonction objective.

Le résolveur Cassowary peut être utilisé dans toutes les opérations d'édition existantes dans le domaine de l'EPDMI. Le seul inconvénient invoqué dans la littérature est les performances

temporelles de l'approche, hors à notre avis cette insuffisance date du temps où les machines disposaient de moyennes performances. En effet, nous avons obtenu un temps de résolution très acceptable pour un document à 600 contraintes et 700 variables. De plus, nous avons anticipé la réduction de cet impact par la mise en place d'un ensemble de contrôles et de règles de construction d'un document.

# Chapitre 6

## Gestion du recouvrement spatial

### 1. Introduction

Assurer une gestion du temps et de l'espace ne suffit aucunement à garantir une qualité de service à la présentation. En effet, le problème du recouvrement spatial des médias, ignoré jusqu'à présent des systèmes étudiés, peut considérablement altérer la qualité de service d'une présentation, d'où une intention particulière doit lui être accordée [69].

Contrairement aux documents classiques où les parties de l'écran sont occupées par les mêmes médias pendant toute la présentation, dans un document multimédia ils peuvent se partager un même espace pendant la présentation. De plus, comme la phase d'édition est séparée de la phase de présentation, cela rend inapplicable le principe de WYSIWYG [28] [100]. Il en résulte alors le risque de se retrouver avec une présentation sérieusement altérée (fig 1).

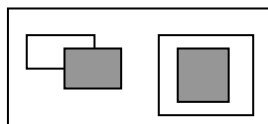


Fig 1. Recouvrement spatial

Nous parlons de recouvrement spatial lorsque deux ou plusieurs médias d'un document occupent, à des instants de présentation communs, entièrement ou partiellement, le même espace de présentation [69].

Hormis la proposition d'une approche qui répond partiellement au problème du recouvrement spatial [7] et une autre destinée au domaine des interfaces graphiques interactives [72], il n'existe pas de solution émergente capable de gérer ce problème. Sa gestion est laissée à la charge de l'auteur qui se retrouve alors contraint, à chaque fois qu'il procède à des opérations d'édition, de vérifier la présentation de son document. À la rencontre de chaque recouvrement, il doit revenir à la phase d'édition pour pouvoir l'éliminer.

Dans ce qui suit, nous commençons par une analyse critique des deux approches proposées dans la littérature avant de présenter notre approche.

### 2. Les approches de gestion du recouvrement spatial

Dans la littérature on trouve deux approches, la première recherche les espaces libres à un moment de la présentation et essaye de placer le média qui a provoqué le recouvrement dans l'un de ces espaces. La deuxième se base sur les contraintes disjonctives pour contraindre les médias à ne pas se recouvrir.

## 2.1 Approche basée sur les rectangles libres

Elle a été mise en œuvre dans un système de génération et de présentation de documents multimédia appelé MPGS [7] utilisant une approche à base de relations pour la spécification des emplacements. L'emplacement spatial d'un média est obtenu à partir des relations spatiales et des *rectangles* disponibles dans l'espace de présentation. Son principe est de déterminer, à tout instant  $t$  de l'intervalle temporel du média à placer, la liste des espaces qui ne sont pas encore occupés. Dans le cas de disponibilité, le média est placé dans l'un de ces espaces en fonction de ses contraintes spatiales. Pour cela l'approche utilise le concept des rectangles libres expliqué par l'exemple de la figure 2 ci-dessous.

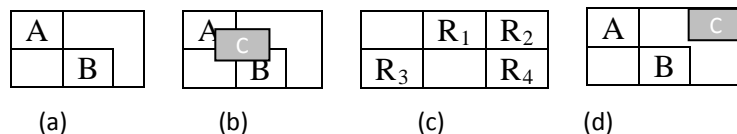


Fig 2. Recherche d'un rectangle libre

Soient deux médias A et B (figure 2a) ayant respectivement les intervalles de présentation [1,10] et [10,20]. Supposons maintenant que l'auteur veut ajouter un média C ayant l'intervalle de présentation [5,15]. A, B et C ont des instants de présentation communs dans l'intervalle [5,15]. Supposons maintenant que l'emplacement de C provoque un recouvrement avec A et B (figure 2b). Pour lever ce recouvrement, le système commence par déterminer les rectangles disponibles durant l'intervalle de C. Pour cela il *retire* les espaces occupés par A et B et délimite les rectangles disponibles à partir de l'espace restant en projetant les bords de A et B sur les bords de l'écran (figure 2c). L'ensemble des rectangles disponibles est alors obtenu à partir des intersections de ces projections, ce qui nous donne  $\{R_1, R_2, R_3, R_4, R_1 \cup R_2, R_2 \cup R_4\}$  (figure 2c). Par la suite, les rectangles non minimaux, c'est-à-dire les rectangles qui sont inclus dans d'autres rectangles, sont éliminés de cet ensemble. Dans notre exemple les rectangles non minimaux sont  $R_1, R_2$  et  $R_4$ . Le système les élimine et l'ensemble ainsi obtenu est  $\{R_3, R_1 \cup R_2, R_2 \cup R_4\}$ . C est placé dans l'un de ces rectangles en fonction de ses relations spatiales et de ses dimensions (figure 2d). Pour cela, à chaque instant  $t \in [5, 15]$  de C, le système détermine les coordonnées et les dimensions de tous les rectangles disponibles à cet intervalle ainsi que l'ensemble des médias en relations spatiales avec C. C est alors placé dans le rectangle libre ayant les dimensions requises et qui respecte ses relations spatiales. Si plusieurs rectangles répondent au placement d'un média, le système choisit celui qui a les dimensions minimales.

## 2.2 Approche basée sur les contraintes disjointives

C'est une approche qui a été proposée par Marriott [72] pour la gestion des interfaces graphiques interactives. Elle se base sur les systèmes de contraintes disjointives. Un système de contraintes est un ensemble de variables et de contraintes qui limitent les combinaisons de valeurs pour ces variables [88][5]. Lorsque la résolution doit satisfaire toutes les contraintes, on dit que les contraintes du système sont conjonctives. À contrario, lorsque la satisfaction d'au moins une contrainte suffit pour sa résolution, les contraintes sont dites disjointives [73]. C'est cette dernière classe de contraintes qui est utilisée dans cette approche. Son principe est de contraindre un média, dont on connaît les coordonnées, à rester à gauche, à droite, au dessus ou au dessous d'un autre média. Ainsi le média peut être placé en dehors des zones occupées par les médias présents. Comme l'approche fait partie de la classe des approches de maintien de solution [21], elle a l'avantage, suite à une opération d'édition provoquant un

échec, dans ce cas un recouvrement sans solution, de revenir à la solution précédente. En considérant le recouvrement comme une incohérence, l'auteur est alors obligé de partir d'une configuration qui ne présente pas de recouvrement. Autrement dit, à chaque opération d'édition, sauf le déplacement, l'auteur doit s'assurer qu'aucun recouvrement n'est provoqué. À partir de là, et pour maintenir cet état de non recouvrement, suite à toute autre opération d'édition ou de déplacement, l'approche génère des contraintes d'inégalité entre le média concerné et le reste des médias. Pour l'exemple, soient les rectangles  $R_1$  ( $l=30$ ,  $h=15$ ) et  $R_2$  ( $l=10$ ,  $h=20$ ) représentés dans la figure 3, en prenant un des rectangles comme référence, le système de contraintes d'inégalité qui ne permet pas le recouvrement de  $R_1$  et  $R_2$  s'écrit :

$$x_2 \geq x_1 + 30 \vee x_1 \geq x_2 + 10 \vee y_2 \geq y_1 + 15 \vee y_1 \geq y_2 + 20$$

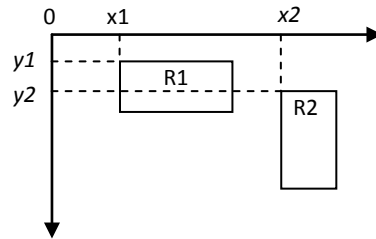


Fig 3. Génération de contraintes disjonctives

Si  $R_1$  est pris comme référence, les contraintes définies correspondent aux cas :  $R_2$  est à droite de  $R_1$ ,  $R_2$  est à gauche de  $R_1$ ,  $R_2$  est au dessous de  $R_1$  et  $R_2$  est au dessus de  $R_1$ .

Il est clair que les contraintes de ce système ne peuvent pas être satisfaites toutes à la fois. En effet, un rectangle ne peut pas être au même temps à gauche et à droite, ou au dessus et en dessous, d'un autre rectangle. De ce fait, la satisfaction d'une seule contrainte garantit une solution au système. La contrainte est alors appelée *contrainte active*. À l'introduction d'une *perturbation* (déplacement, ajout d'un média, etc.), si la contrainte active ne peut plus être satisfaite, le résolveur essaye d'activer une des autres contraintes. Si aucune contrainte ne peut être satisfaite le système n'a alors pas de solution. C'est le cas lorsqu'il n'y a plus d'espace suffisant pour déplacer un média ou lorsqu'il existe des relations spatiales contradictoires avec les contraintes disjonctives.

Pour illustrer ce que nous venons de dire reprenons l'exemple précédent et considérons la configuration où le rectangle  $R_2$  est à droite de  $R_1$  (figure 4a) et que la contrainte  $x_2 \geq x_1 + 30$  est active. Si nous déplaçons  $R_1$  vers la droite,  $R_2$  suivra automatiquement le mouvement dès que son bord gauche est touché par  $R_1$  (figure 4b). Si par contre  $R_1$  est déplacé vers le haut (figure 4c) alors la contrainte  $y_2 \geq y_1 + 15$  est activée en plus de la contrainte précédente  $x_2 \geq x_1 + 30$ . Maintenant, si nous poursuivons le déplacement de  $R_1$  vers la droite (figure 4d), la contrainte  $x_2 \geq x_1 + 30$  sera désactivée, car il n'existe aucun risque de recouvrement entre  $R_1$  et  $R_2$ , et par conséquent  $R_2$  ne suivra pas le mouvement. De là, nous pouvons déduire que l'objectif de l'approche de Marriott est *d'éviter* le recouvrement des médias et non pas forcément le maintien des médias dans leurs emplacements spécifiés par l'auteur.

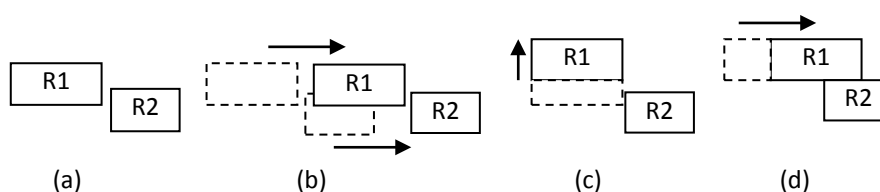


Fig 4. Déplacement d'un média

## 2.3 Discussion

Dans le domaine de l'EPDMI, une approche de gestion du recouvrement spatial doit permettre :

1. Le respect des spécifications des emplacements de l'auteur,
2. le déplacement des médias lors de la phase d'édition,
3. la prise en compte de toutes les opérations d'édition pouvant provoquer un recouvrement spatial,
4. et la prise en compte de la dimension temporelle des médias.

L'approche MPGS prend en compte la dimension temporelle, respecte les spécifications des emplacements et sa mise en œuvre est très intuitive. Néanmoins, elle ne permet pas le déplacement des médias pendant la phase d'édition. Le calcul des rectangles libres à chaque seconde de l'intervalle de présentation d'un média (unité temporelle de base dans le domaine de l'EPDMI) alourdit considérablement les performances temporelles de l'approche. Enfin, l'absence de rectangles de dimensions adéquates à celles du média ayant provoqué le recouvrement, n'implique pas forcément qu'il ne peut pas être placé ; il suffit d'effectuer dans certains cas, quelques déplacements de médias pour dégager l'espace approprié.

Quant à l'approche de Marriott, elle s'inscrit dans la classe des approches de programmation par contraintes, classe qui a déjà fait ses preuves dans le domaine de l'EPDMI [21][4], notamment pour la prise en charge des problèmes de la cohérence temporelle et spatiale [67] et permet le déplacement des médias durant la phase d'édition. Cependant, elle ne tient pas compte de la dimension temporelle, ceci l'amène à considérer que tous les espaces occupés sont inaccessibles pendant toute la durée de la présentation et non pas juste au moment de la présentation des médias concernés. Par conséquent elle suppose que tous les médias ne doivent pas se recouvrir deux à deux et de là, elle génère des systèmes de contraintes entre tous les médias. Enfin, la condition de partir initialement, lors de chaque opération d'édition, avec une configuration qui ne présente pas de recouvrements la rend contraignante, car ceci est la cause essentielle du recouvrement dans le domaine de l'EPDMI.

Toutefois, nous estimons qu'il est intéressant d'adapter l'approche de Marriott à notre domaine d'étude, et ce pour les raisons suivantes. Premièrement, son principe peut être généralisé à toute opération d'édition pouvant provoquer un recouvrement spatial. Deuxièmement, la gestion des dimensions temporelle, spatiale et du recouvrement seront unifiées par l'utilisation des systèmes de contraintes. Enfin, faisant partie des approches de maintien de solution, ceci facilitera l'édition à l'auteur en lui évitant, entre autres, de reprendre son document dès qu'un recouvrement est signalé.

## 3. Approche proposée

Afin de cerner tous les aspects du problème, nous avons adopté la démarche suivante. Nous avons commencé par recenser toutes les opérations susceptibles de provoquer un recouvrement spatial. Nous avons par la suite déterminé, en fonction des modélisations temporelle et spatiale retenues, les informations d'un média (variables et contraintes) nécessaires pour le système de contraintes. Afin de permettre la spécification d'une incrustation ou d'un média de fond, nous avons introduit le concept du recouvrement désiré. Nous avons également proposé un ensemble de solutions pouvant lever ou tolérer un recouvrement non désiré lorsqu'il n'existe pas de solution pour le système de contraintes. Et en dernier lieu, des modifications à l'approche de Marriott, notamment la levée de la condition

initiale de non recouvrement et une génération non systématique de systèmes contraintes, ont été apportées [69].

### 3.1 Opérations pouvant provoquer un recouvrement

À ce niveau, un recensement des opérations pouvant provoquer un recouvrement spatial et/ou qui nécessitent le maintien de solution a été effectué. Nous avons : l'insertion d'un média, d'une relation spatiale ou temporelle ; la modification d'une relation spatiale ou temporelle, d'un attribut temporel, des coordonnées spatiales ou de la taille d'un média et la suppression d'une relation temporelle, spatiale ou d'un média.

### 3.2 Les variables du système de contraintes

Par rapport à la modélisation temporelle et spatiale retenues, nous considérons 9 variables pouvant intervenir dans la gestion du recouvrement, les 6 variables spatiales données par la Figure 5 et les 3 variables temporelles *début*, *durée* et *fin*.

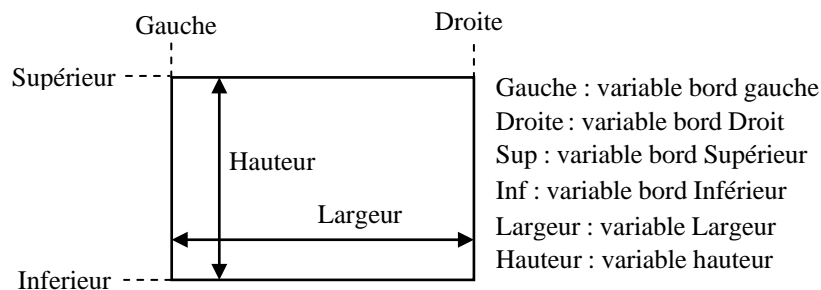


Fig 5. Représentation graphique d'un média

### 3.3 Recouvrement désiré

Il est clair que le recouvrement spatial peut affecter la qualité de service d'une présentation et qu'il faut par conséquent l'éliminer. Néanmoins, dans certains cas, il peut aussi répondre à un besoin bien déterminé de l'auteur. Aussi l'auteur peut vouloir, par exemples, qu'un logo soit incrusté dans une image ou qu'un texte défile sur la partie basse d'une vidéo. A cet effet, nous avons introduit le concept du recouvrement désiré : à sa rencontre le système de gestion du recouvrement l'*ignore*.

### 3.4 Tolérance d'un recouvrement

L'idée est de considérer, à défaut d'une solution d'emplacement (échec du solveur), le cas où l'auteur peut tolérer qu'un média soit partiellement recouvert (Figure 6) [69].

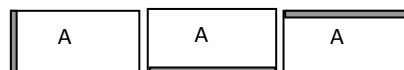


Fig 6. Exemples de recouvrements tolérés

Cette proposition, nous l'avons formalisée de la manière suivante : à la spécification d'un média, l'auteur indique s'il le souhaite, la ou les parties du média où le recouvrement peut être toléré. Par la suite le système traduit cette tolérance en un système de contraintes disjonctives. Pour cela, nous définissons quatre coefficients,  $C_{i, i=1,4}$  tel que  $0 \leq C_i \leq 1$ , représentant le pourcentage de recouvrement toléré associé aux variables *largeur* et *hauteur* du média à partir

de ses quatre bords. Suite à quoi un système de contraintes disjonctives est généré où chaque contrainte traduit la tolérance de recouvrement sur l'un des bords du média concerné.

Dans le cas général, le système de contraintes traduisant la tolérance d'un recouvrement d'un média A par un média B est formulé de la façon suivante :

$$\begin{aligned}
 A_g &\geq B_d - c_1(A_l) \vee \\
 A_d &\leq B_g + c_2(A_l) \vee \\
 A_{sup} &\geq B_{inf} - c_3(A_h) \vee \\
 A_{inf} &\leq B_{sup} + c_4(A_h)
 \end{aligned}$$

Initialement, tous les  $C_i$  sont initialisés à zéro ( $c_1 = c_2 = c_3 = c_4 = 0$ ) ce qui signifie qu'aucun recouvrement n'est toléré. Par contre si  $C_i=1 \forall i=1,4$  ou ( $c_1 + c_3 = 1$ ) ou ( $c_2 + c_4 = 1$ ) alors un recouvrement total du média est toléré, et dans ce cas le système de contraintes associé ne sera pas généré car il est toujours à vrai.

Si nous prenons l'exemple de la figure 7 où l'auteur tolère un recouvrement de 20 % sur la partie supérieur de A et de 10% sur sa partie gauche, on aura alors :  $c_1=0,1$ ,  $c_2=0$ ,  $c_3=0,2$  et  $c_4=0$ . Le système de contraintes associé s'écrit :

$$\begin{aligned}
 A_g &\geq B_d - 0,1(A_l) \vee \\
 A_d &\leq B_g \vee \\
 A_{sup} &\geq B_{inf} - 0,2(A_h) \vee \\
 A_{inf} &\leq B_{sup}
 \end{aligned}$$

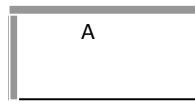


Fig 7. Recouvrement toléré sur A

### 3.5 Réduction des dimensions d'un média

La réduction des dimensions d'un média ou des deux médias concernés par un recouvrement pourrait éliminer ce dernier. Pour illustrer ce cas, prenons l'exemple de la figure 8a où l'insertion de B provoque un recouvrement sur A. Une réduction de la largeur de B élimine le recouvrement (figure 8b).

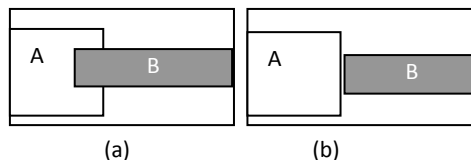


Fig 8.Exemple de réduction d'un média

Cette solution est considérée de la manière suivante : à la spécification d'un média l'auteur introduit s'il le souhaite, les limites de réduction qu'il juge acceptables, car un retailage non contrôlé peut altérer gravement la qualité visuelle du média et par conséquent la qualité de service de la présentation. De plus, pour le maintien de la stabilité des emplacements successifs définis par l'auteur, nous privilégions en premier lieu, la réduction de la taille du média qui a provoqué le recouvrement, puis le second suivi des deux à la fois si c'est

nécessaire. Enfin, pour réduire le moins possible la taille du média, nous procédons par une réduction progressive (pixel par pixel) jusqu'à atteindre les limites spécifiées par l'auteur [65].

### 3.6 Modification des attributs temporels d'un média

En partant de l'axiome qui dit que les médias qui ont des intervalles de présentation disjoints ne risquent jamais de se recouvrir, il suffit donc de rendre disjoints les intervalles des deux médias concernés pour lever le recouvrement. Pour cela nous procédons comme suit : pour les mêmes raisons de stabilité que précédemment, nous décalons l'attribut *début* du média qui a provoqué le recouvrement suivi, si c'est nécessaire, de celui qui a subi le recouvrement. Comme cette opération de décalage peut engendrer une ou plusieurs autres intersections d'intervalles et/ou une incohérence temporelle, elle devra donc être itérée autant de fois. Ceci peut conduire à un cycle infini d'opérations de décalage ou à une présentation dénaturée de son sens originel. Pour cela, nous avons choisi d'arrêter le processus dès que la première modification génère un autre recouvrement ou une incohérence temporelle [65].

## 4. Adaptation de l'approche disjonctive

Afin d'adapter l'approche au domaine de l'EPDMI, nous lui avons apporté deux importantes modifications [69].

### 4.1 La levée de la condition de non recouvrement initial

Comme signalé l'approche de Marriott exige que l'opération d'édition ne provoque pas initialement un recouvrement. Autrement dit, lorsque l'auteur insère un média et que ce dernier provoque d'emblée un recouvrement cette approche est incapable de l'éliminer. Or dans le domaine de l'EPDMI ceci est quasiment la cause essentielle du recouvrement spatial. Afin de pallier à ce problème, l'idée est de générer une contrainte d'inégalité entre les médias qui se recouvrent de façon à déplacer l'un des deux pour éliminer le recouvrement. Une fois ce dernier éliminé, la contrainte d'inégalité est retirée, des contraintes disjonctives entre ces deux médias sont générées et une d'elles est activée afin de maintenir le non recouvrement. Dans le cas où l'espace de présentation restant est insuffisant, aucune contrainte ne peut alors être activée et par conséquent le recouvrement ne peut être éliminé. Pour illustrer l'idée, prenons l'exemple de la figure 9 où l'insertion de B provoque un recouvrement sur A (figure 9b). Pour l'éliminer, et en fonction de la configuration du recouvrement, nous générons par exemple la contrainte d'inégalité  $B_g \geq A_g + A_l$  qui provoque le déplacement de B à droite de A (figure 9c). Par la suite, nous la supprimons et nous ajoutons au système global l'ensemble des contraintes disjonctives suivant :

$$\begin{aligned} A_g &\geq B_d - c_1(A_l) \vee \\ A_d &\leq B_g + c_2(A_l) \vee \\ A_{sup} &\geq B_{inf} - c_3(A_h) \vee \\ A_{inf} &\geq B_{sup} + c_4(A_h) \end{aligned}$$

Par la suite, la contrainte  $A_d \leq B_g + c_2 A_l$  est activée pour le maintien du non recouvrement.

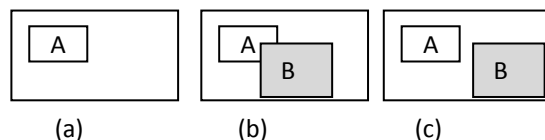


Fig 9. Elimination du recouvrement par les contraintes d'inégalités

## 4.2 Génération sélective des systèmes de contraintes disjonctives

Dans l'approche de Marriott tous les médias d'un document sont contraints à ne pas se recouvrir deux à deux. Or nous avons vu dans le domaine de l'EPDMI, que non seulement le recouvrement peu être désiré par l'auteur, mais également, qu'il ne peut toucher que les médias qui ont des instants de présentation communs. Compte tenu de tout cela, pour augmenter les chances de la levée d'un recouvrement en offrant le maximum d'emplacements possibles et pour améliorer les performances temporelles du système, nous introduisons le concept de *génération sélective* des systèmes de contraintes disjonctives : un système de contrainte disjonctives entre A et B est généré si et seulement si A et B ont des instants de présentation communs, ont une intersection de coordonnées, et que leur recouvrement n'est pas désiré [69].

## 5. Conclusion

Comme nous l'avons signalé tout le long de ce chapitre, le recouvrement spatial est un problème à forte conséquence sur la qualité de service d'une présentation multimédia. Actuellement on trouve très peu de travaux de recherche sur le recouvrement spatial et encore moins dans le domaine de l'EPDMI où sa gestion est le plus souvent laissée à la charge de l'auteur. L'étude et l'analyse des deux approches présentées nous ont conduit à constater que les inconvénients de l'approche des rectangles sont son incapacité à permettre le déplacement d'éléments dans la phase d'édition, son incomplétude et son temps de réponse jugé inacceptable dans le domaine de l'EPDMI. Dans l'approche proposée par Marriott, l'utilisation des systèmes de contraintes disjonctives est très intéressante de part leur formalisme éprouvé et leur application qui a donné des résultats appréciables dans de nombreux domaines de recherche. Cependant, la condition initiale de non recouvrement et l'obligation de définir des systèmes de contraintes disjonctives entre tous les éléments d'un document la rendent inapplication dans le domaine de l'EPDMI.

L'approche proposée, utilise le principe des systèmes de contraintes disjonctives adapté au domaine de l'EPDMI. Pour la levée de la condition initiale de non recouvrement, nous avons proposé une solution formalisée avec les contraintes disjonctives consistant à introduire une contrainte d'inégalité de telle façon à déplacer un des deux éléments pour éliminer le recouvrement. Une fois ce dernier éliminé nous retirons la contrainte d'inégalité et nous forçons le solveur à générer des contraintes disjonctives entre ces deux éléments et d'en activer une. Pour augmenter les chances de placement d'un élément responsable d'un recouvrement et améliorer les performances de calcul du système, nous avons introduit le concept de la *génération sélective* des systèmes de contraintes disjonctives. De plus, pour permettre à l'auteur de spécifier des opérations d'incrustation nous avons défini le concept du recouvrement désiré. En fin, nous proposons, une fois le placement du média n'est plus possible (recouvrement persistant), un ensemble de solutions (le principe de la tolérance, du retailage et de la modification des attributs temporels) capable de lever le recouvrement dans les limites souhaitées par l'auteur.

# Chapitre 7

## Architecture et mise en œuvre

### 1. Introduction

Ce chapitre présente l'architecture du système d'édition et de présentation, dénommé dans ce qui suit MediaStudio, et la mise en œuvre de l'ensemble de ses fonctions qui gèrent la création d'un document/parcours de lecture multimédia interactif. La partie présentation est détaillée dans le chapitre suivant.

La démarche adoptée consiste à mettre en correspondance d'une part le processus d'édition avec la phase de spécification et d'analyse (insertion d'information temporelles, spatiale, de recouvrement, de relations, etc. et la vérification de la cohérence), et d'autre part le processus de présentation avec la phase de résolution de ce problème (la synthèse).

### 2. Architecture de MediaStudio

L'architecture décrit les quatre dimensions d'un document multimédia et leur interaction. Pour garantir l'extensibilité et pour faciliter la portabilité du système sur différentes plateformes, ces fonctions ont été mises en œuvre sous forme d'un ensemble de modules.

#### 2.1 Principe de conception

L'approche adoptée dans cette thèse tente d'allier édition interactive et présentation multimédia avec le domaine des systèmes de contraintes. Du côté de l'édition, ce qui nous paraît intéressant dans une approche interactive similaire au Wysiwyg, c'est la possibilité de voir et de modifier le document sans changer d'environnement : le passage du mode d'édition au mode de présentation, et vice-versa, doit être aussi fluide que possible. L'intérêt d'une telle approche réside dans l'interactivité, qui rend plus facile l'accès au système et permet à l'utilisateur de vérifier en permanence ce qu'il fait. Ce principe, largement adopté pour les documents statiques, est cependant plus complexe à mettre en œuvre pour les documents multimédia. Cela est dû à la nature temporelle très diverse des médias et des dimensions multiples du document. D'un autre côté, les systèmes de contraintes permettent à l'auteur d'exprimer ce qu'il souhaite obtenir, sans se soucier de la complexité du scénario qu'il construit. Les systèmes de contraintes permettent ainsi, par des méthodes plus systématiques d'analyse et de synthèse, de vérifier la cohérence d'un scénario et de rechercher une solution.

En partant de ces constats, le système MediaStudio tente de tirer profit des avantages des systèmes de contraintes pour l'édition multimédia en offrant à l'utilisateur, à travers la même interface graphique, des fonctions d'édition et de présentation.

## 2.2 Architecture générale du système

Nous présentons dans cette section l'architecture générale du système MediaStudio. Cette architecture est organisée sous forme de quatre parties qui interagissent tout au long d'une session d'édition (voir Figure 1 ) :

- **L'interface utilisateur** : elle regroupe les moyens graphiques offerts à l'utilisateur comme les boutons, les formulaires et les palettes pour réaliser les opérations d'édition et de présentation.
- **Le système d'édition** : il offre l'ensemble des fonctions permettant, la création ou la modification d'un document et la définition des relations temporelles, spatiales et des liens hypermédia. Ce module permet aussi le chargement d'un document ainsi que sa sauvegarde. Il regroupe quatre parties :
  - **Le gestionnaire temporel** : il prend en charge toute la gestion des opérations d'édition et de présentation qui concernent la dimension temporelle du document.
  - **Le gestionnaire spatial** : il assure la gestion des opérations d'édition et de présentation des informations spatiales du document.
  - **Le gestionnaire de recouvrement** : il prend en charge la gestion du recouvrement spatial des médias.
  - **Le gestionnaire des parcours** : il offre un ensemble d'opérations permettant la création, la modification et la suppression de parcours de lecture.
- **Le gestionnaire des médias** : il regroupe l'ensemble des opérations permettant la création, la modification, l'initialisation, le lancement et l'arrêt des médias. Il est détaillé au niveau du chapitre suivant.
- **Le système de présentation** : il prend en charge la restitution du document à l'utilisateur en assurant l'ordonnancement des médias, la gestion des ressources et la gestion de l'environnement d'exécution (interactions de l'utilisateur, indéterminisme des durées, etc.). Cette partie fera l'objet du chapitre suivant.

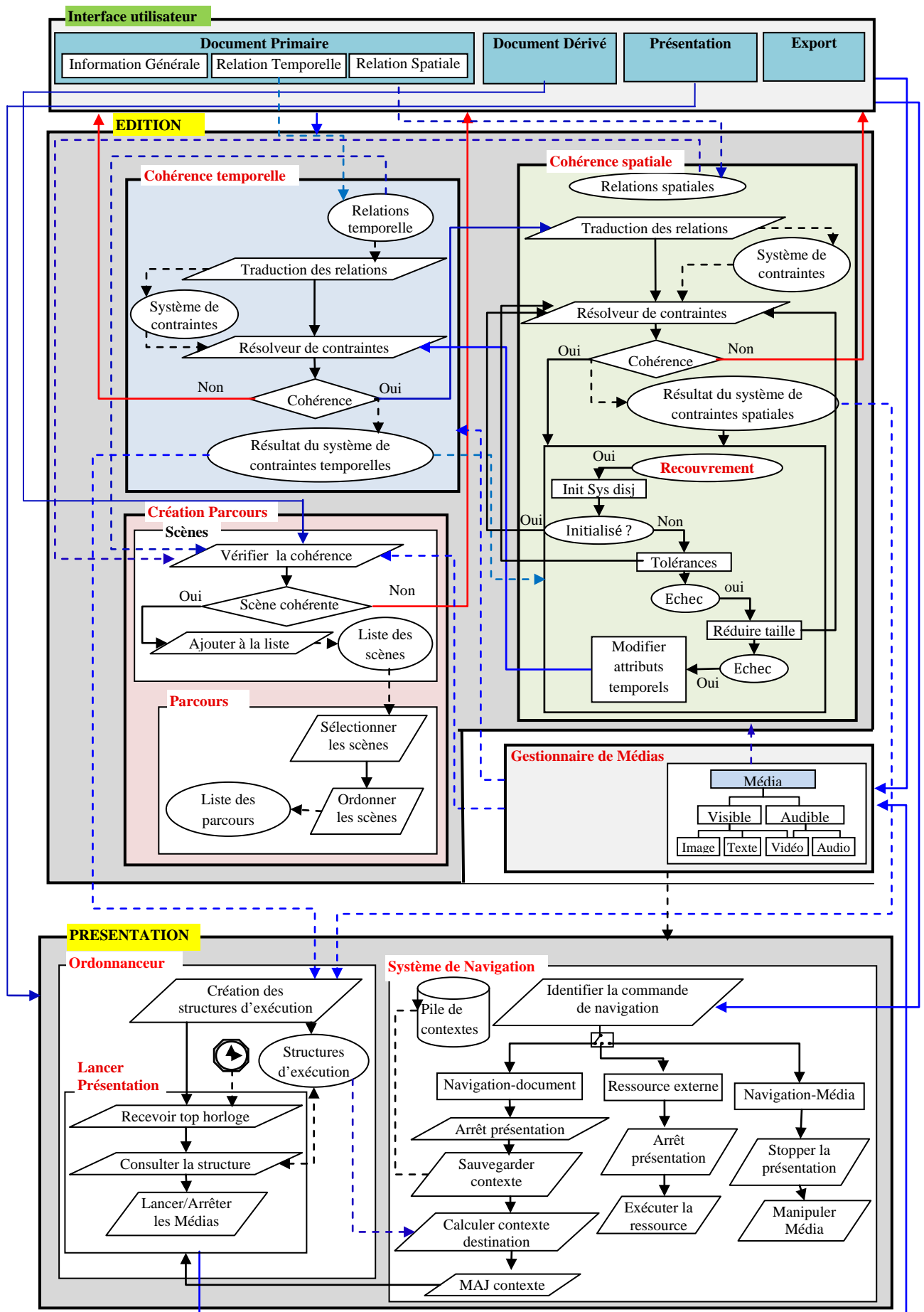


Fig 1. Architecture générale de MediaStudio

### **3. Conditions et règles d'édition**

Avant de présenter la mise en œuvre de MediaStudio, nous allons commencer par présenter les règles de construction d'un document ou d'un parcours de lecture ainsi et que les contrôles a priori retenus.

#### **3.1 Paramétrage et contrôle a priori**

##### **3.1.1 Paramétrage par défaut du système**

- L'unité de mesure d'une durée ou d'un délai est la seconde.
- Les durées minimale, préférentielle et maximale d'un média dynamique sont égales à sa durée intrinsèque.
- Le début d'un média est initialisé à l'instant 0 et ses coordonnées sont initialisées au point (0,0).
- La validation d'un document est lancée par défaut après chaque opération d'édition sinon elle peut être activée à tout moment par l'auteur.
- Initialement les médias ne tolèrent aucun recouvrement et aucun retailage.

##### **3.1.2 Les contrôles a priori**

Afin d'alléger la phase d'analyse, un ensemble de contrôles est effectué a priori :

- La durée d'un média et ses dimensions spatiales ne peuvent être négatives ou nulles.
- Les contraintes induites par les relations temporelles et spatiales sont plus fortes que celles des informations d'emplacement et d'instant de début d'un média.
- L'identificateur d'un média est son nom, il est unique.
- Les médias audio n'ont ni de relations spatiales ni d'attributs spatiaux.
- Tout média doit avoir une relation spatiale et une relation temporelle avec au moins un autre média.
- Entre deux médias, il ne peut y avoir plus d'une relation temporelle ou plus d'une relation spatiale.
- Avant toute présentation ou exportation d'un document, la vérification de sa validité est effectuée.
- Un média dérivé est composé à partir des médias physiques de même type.
- Une fois le média inséré son type ne peut plus être modifié.

#### **3.2 Nombre de relations entre les médias**

Afin de simplifier l'édition, de ne pas surcharger le système de contraintes et d'anticiper sur le contrôle de la cohérence, nous avons adopté le principe de ne permettre qu'une seule relation temporelle ou spatiale entre deux médias. Après une étude empirique, et notamment avec les relations les plus utilisées dans le domaine de l'EPDMI, nous avons constaté qu'une spécification au delà de deux relations entre deux média peut le plus souvent être ramenée à une seule relation ou bien qu'elle génère une incohérence. À titre illustratif, la figure 2 donne quelques exemples des relations les plus fréquemment utilisées et leurs équivalences ou incohérences signalées.

Spécification	Equivalence/Incohérence
A cobegin (#) B and A coend (#) B	B delayed (#,#) A
B cobegin (#) A and A coend (#) B	A while (#, #) B
A cobegin (#) B and A endin (#,#) B	A overlaps (#,#,#) B
A cobegin (#) B and A coend ([0, 0]) B	B while (#,[0, 0]) A
A cobegin (#) B and B beforeendof (#) A	B startin (+,+) A
A beforeendof (#) B and B beforeendof (#) A	A cross (#, #) B
A cobegin (#) B and A overlaps (#, #, #) B	A overlaps (#, #, #) B
A cobegin (#) B and A While ([0, 0], #) B	Incohérence
A startin (#,#) B and A cobegin ([0, 0]) B	Incohérence
B total_centring A and B right (# <sub>1</sub> , # <sub>2</sub> ) A	Incohérence
B top_shift (# <sub>2</sub> ) A and B below (# <sub>1</sub> , # <sub>2</sub> ) A	Incohérence
B left_Top (# <sub>1</sub> , # <sub>2</sub> ) A and B right_bottom (# <sub>1</sub> , # <sub>2</sub> ) A	Incohérence
B right([ $\delta^1_1$ , $\delta^1_2$ ], [-, -]) A and B above([-, -], [ $\delta^2_1$ , $\delta^2_2$ ]) A and B left_shift([A.largeur, A.largeur]) A.	B right_top ([ $\delta^1_1$ , $\delta^1_2$ ],[0, 0])A
B left([ $\delta^1_1$ , $\delta^1_2$ ], [-, -]) A and B left_shift([-A.largeur,- A.largeur]) A	Incohérence

Fig 2. Exemple de spécifications ramenées à une relation ou provoquant une incohérence

### 3.3 Concept d'activation/désactivation

Pour simplifier d'avantage l'édition, nous permettant à l'auteur de spécifier un média dans le cas où il n'est pas encore fixé sur ses informations (début, durée, relations, etc.). Pour cela nous définissons le concept d'activation/désactivation. Dans ce cas, le média est mis à *désactivé*.

La présence d'un média avec un état *désactivé* dans un document indique que ce dernier n'est pas valide et par conséquent il ne peut être joué. Dans ce cas, le média est maintenu, mais ignoré, jusqu'à ce que l'auteur le complète ou le supprime. L'état d'un média est mis à *activé*, lorsque l'auteur complète ses informations.

### 3.4 Règles de construction d'un parcours de lecture

Afin de s'assurer de la cohérence de la construction d'un parcours de lecture par rapport aux informations du document primaire, un ensemble de règles de constructions doit être observé :

- Si le document primaire est modifié, tous ses documents dérivés, scènes et parcours sont mis à jour.
- Un document dérivé contient au moins une scène.
- Une scène contient au moins un média.
- Un parcours contient au moins une scène.
- Dans un parcours de lecture, un média ne doit pas appartenir à plus d'une scène.
- Une scène peut appartenir à plusieurs parcours du même document dérivé.
- Quels que soient deux médias d'une même scène, sauf si elle est composée d'un seul média, il faut qu'ils soient liés par une relation temporelle directe ou indirecte, si elle est indirecte elle doit passer uniquement par les médias de la scène.

## 3.5 Les règles d'édition

### 3.5.1 L'insertion

#### - **D'un média**

À l'insertion d'un média, l'auteur doit préciser son nom et son type. Le reste des informations générales, temporelles et spatiales peuvent être fournies immédiatement ou ultérieurement.

Une fois que le chemin du média est indiqué, le système extrait automatiquement les informations relatives à sa hauteur et sa largeur (image et vidéo) ainsi que sa durée intrinsèque (audio et vidéo) qui peuvent être modifiées par l'auteur.

Les dimensions (hauteur et largeur) de la boîte d'affichage d'un média dérivé vidéo sont celles des plus grands médias desquels il dérive. Pour une image dérivée, sa hauteur est la somme des hauteurs des médias placés verticalement et sa largeur est la somme des largeurs des médias placés horizontalement. La durée d'un média dérivé vidéo ou audio est la somme des durées de ses parties.

#### - **D'une relation**

Une relation peut être spécifiée à l'insertion d'un média ou à tout moment de l'édition. Si une relation existe déjà entre les deux médias, une erreur est signalée et l'opération est annulée.

#### - **D'un lien**

Durant la phase d'édition, l'auteur peut insérer des liens de type saut temporel ou référence en indiquant respectivement la destination ou le chemin. Le lecteur peut définir uniquement des liens de type saut temporel lors de la construction de ses parcours de lecture.

Le lien est matérialisé par un contour spécifique.

### 3.5.2 La Création

#### - **D'un document dérivé**

Du document primaire, un ensemble de documents dérivés peut être défini par l'auteur ou par le lecteur.

#### - **D'une scène**

À la création d'une scène, l'utilisateur doit lui fournir un nom et sélectionner les médias la composant.

#### - **D'un parcours de lecture**

À la création d'un parcours, l'utilisateur lui fournit un nom et choisit l'ensemble des scènes le composant. Si la destination d'un saut temporel n'existe pas dans le parcours de lecture, l'utilisateur en est avisé et le lien est ignoré.

### 3.5.3 La modification

#### - **Des informations générales**

Le nom: sa modification ne nécessite aucune action, le système adresse les médias par un identificateur interne.

Le chemin : sa modification peut nécessiter la modification des attributs spatiaux (la hauteur et la largeur) et l'attribut temporel durée. Dans ce cas, le système retire les contraintes correspondantes et les redéfinit.

#### - **Des informations temporelles et spatiales**

L'auteur peut modifier les attributs temporels (le début, la fin ou l'intervalle de durée) ou spatiaux (les coordonnées, la hauteur ou la largeur) d'un média. Le système retire les contraintes correspondantes et les redéfinit.

- **De l'emplacement d'un média**  
L'auteur peut modifier l'emplacement d'un média par déplacement sur écran (clavier ou souris). Dans ce cas, le système essaie de placer le média à l'emplacement spécifié, en cas d'échec ce dernier est replacé dans sa position initiale.
- **Des informations du recouvrement**  
À la modification des informations de recouvrement, les systèmes disjonctifs correspondants sont retirés, réécrits et réintroduits dans le système de résolution.
- **D'une relation et de ses attributs (délai ou distance)**  
Dans ce cas, les contraintes de la relation sont retirées du système et les contraintes de la nouvelle relation sont redéfinies.
- **D'un média référence**  
La modification d'un média référence passe par plusieurs étapes :
  - Si la nouvelle référence possède déjà une relation avec le média secondaire, l'auteur en est informé et l'opération est annulée.
  - Si l'ancien média référence risque une suppression (il n'a plus de relations), l'auteur en est avisé. Dans le cas où il persiste, l'ancienne référence et l'ancienne relation sont supprimées. La nouvelle relation est définie avec la nouvelle référence.
  - Si la référence possède d'autres relations, l'ancienne relation est supprimée et la nouvelle relation est définie avec la nouvelle référence.
- **D'un lien de navigation**  
L'auteur peut modifier la destination d'un lien en sélectionnant la nouvelle cible.
- **D'un document dérivé**  
La modification d'un document dérivé porte sur l'ajout ou la suppression de médias ou le changement de son nom.
- **D'une scène**  
L'utilisateur peut changer le nom de la scène, lui ajouter ou lui supprimer des médias. La scène est mise à jour. Tous les parcours la contenant sont à leur tour mis à jour.
- **D'un parcours de lecture**  
L'auteur ou le lecteur peut changer le nom de son parcours, modifier les délais de ses scènes séquentielles ou ajouter une scène à un parcours. Il peut aussi supprimer une scène du parcours et dans ce cas, l'ordonnancement du parcours est mis à jour et s'il existe dans le parcours des liens qui pointent vers cette scène ou un de ses médias, ils seront ignorés.

### 3.5.4 Suppression

- **D'un média**  
Si la suppression du média peut entraîner la suppression de d'autres médias (ceux qui n'auront plus de relations ni temporelles ni spatiales), alors l'auteur en est informé, s'il persiste le média, ses contraintes et ses relations sont supprimés et les médias qui n'ont plus de relation spatiale ou de relation temporelle sont également supprimés. Tous les liens définis à partir et vers ce média seront supprimés. Sinon le média, ses contraintes et ses relations seront supprimées et les tous les liens définis à partir et vers ce média sont supprimés
- **D'une relation**  
La relation est supprimée et ses contraintes sont retirées du résolveur. Dans le cas où un des médias de la relation risque une suppression (il ne possède que cette relation), l'auteur est avisé et peut annuler l'opération.
- **D'un document dérivé**  
La suppression d'un document dérivé revient à supprimer automatiquement toutes ses scènes et tous les parcours construits à partir de celles-ci (voir suppression de parcours).

- **D'une scène**  
À la suppression d'une scène, le document dérivé est mis à jour et l'ordonnancement de tous les parcours la contenant est mis à jour. Tous les liens définis à partir et vers cette scène seront supprimés.
- **D'un parcours de lecture**  
L'auteur et le lecteur peuvent supprimer leur propres parcours et cela sans aucune incidence.

### 3.5.5 Gestion des documents

- **La session**  
Une session correspond à une instanciation de tous les modules du système opérant sur le même document ainsi que l'ensemble des ressources nécessaires à son édition et sa présentation. Les informations d'une session sont enregistrées dans un fichier temporel avant d'être transférées au document en cours lors de sa sauvegarde.  
Une session peut être de deux types : auteur ou lecteur. L'identification se fait lors de l'ouverture de la session en indiquant le type de la session, le login et le mot de passe. Une session auteur, lui donne accès à ses propres documents et à tous les modules du système (édition et présentation). Avec une session lecteur, l'utilisateur peut accéder, aux documents primaires non protégés pour en définir des documents dérivés, à ses propres documents dérivés et parcours de lecture, et jouer les documents existants. À sa fermeture, dans le cas où un document n'a pas été sauvegardé, l'auteur en est averti.
- **Un nouveau document**  
La création d'un nouveau document correspond à l'ouverture d'une nouvelle session auteur. Si la création du nouveau document s'effectue lors d'une session, l'auteur est appelé à sauvegarder son travail de la session courante pour qu'une nouvelle session puisse être initialisée.
- **Ouverture d'un document**  
L'ouverture d'un document revient à charger son fichier de sauvegarde. Au chargement, le système vérifie l'existence de toutes les ressources du document. En cas où il se heurte à une adresse invalide (la ressource n'existe plus), une alerte critique est émise. Si la session est ouverte par un lecteur, elle sera automatiquement fermée.
- **Sauvegarde d'un document**  
La sauvegarde d'un document revient à enregistrer sa structure dans un format interne au système.
- **Importation et exportation de document et de parcours de lecture**  
Pour assurer l'importation et l'exportation des documents/parcours de lecture, nous avons défini dans MediaStudio, un format XML pivot et un schéma XML (voir annexe 2) pour leur validation.  
À l'importation d'un document XML, ce dernier est *parsé*, s'il n'est pas valide il est rejeté. Sinon le système reconstitue le document dans le format interne.  
Seuls les documents globalement cohérents (temporellement, spatialement et ne possédant pas de recouvrement non désiré) et par conséquent valides schéma XML, peuvent être exportés. Lors de l'exportation, le fichier XML est accompagné par une application (visionneuse) pour le jouer.

## 4. Représentation des contraintes dans MediaStudio

Afin de bénéficier des avantages d'une représentation en intervalle (la projection des médias dans le temps correspond à des intervalles, la représentation est plus compacte et elle répond mieux aux critères d'encapsulation et de modularité) et de ceux d'une représentation en

instants (elle fournit un très bon support d'exécution), au sein de l'ordonnanceur, la structure temporelle et spatiale d'un document est maintenue à deux niveaux :

- Au niveau de la spécification, sous forme d'un ensemble de relations à base d'intervalles.
- À un niveau interne, où les relations sont traduites sous forme de systèmes de contraintes à base d'instants.

L'approche adoptée dans MediaStudio repose donc principalement sur une mise en correspondance entre une représentation de haut niveau et une gestion interne de plus bas niveau fondée sur les instants.

## 4.1 Traduction des informations en contraintes

Comme mentionné, les relations des modèles temporel et spatial ainsi que les informations intrinsèques des médias sont traduites en contraintes.

### 4.1.1 Traduction des relations temporelles

La figure 3 ci-dessous donne la traduction des relations temporelles en contraintes.

Relations temporelles	Traduction en contraintes
B starts A	$B.begin = A.begin$
B finishes A	$B.end = A.end$
B equals A	$B.begin = A.begin \text{ And } B.end = A.end$
B cobegin (#) A	$B.begin + \delta^1 \leq A.begin \text{ And } B.begin + \delta^2 \geq A.begin; \delta^i \geq 0$
B before (#) A	$B.end + \delta^1 \geq A.begin \text{ And } B.end + \delta^2 \leq A.begin; \delta^i \geq 0$
B beforeendof (#) A	$B.begin + \delta^1 \leq A.end \text{ And } B.begin + \delta^2 \geq A.end; \delta^i > 0$
B coend (#) A	$B.end + \delta^1 \leq A.end \text{ And } B.end + \delta^2 \geq A.end; \delta^i \geq 0$
B while (#, #) A	$(B.begin \geq A.begin + \delta_1^1 \text{ And } B.begin \leq A.begin + \delta_1^2) \text{ And } (B.end + \delta_2^1 \leq A.end \text{ And } B.end + \delta_2^2 \geq A.end); \delta_i^k > 0$
B delayed (#, #) A	$(B.begin \geq A.begin + \delta_1^1 \text{ And } B.begin \leq A.begin + \delta_1^2) \text{ And } (B.end \geq A.end + \delta_2^1 \text{ And } B.end \leq A.end + \delta_2^2); \delta_i^k > 0$
B startin (#, #) A	$B.begin \geq A.begin + \delta_1^1 \text{ And } B.begin \leq A.begin + \delta_1^2) \text{ And } (B.begin + \delta_2^1 \leq A.end \text{ And } B.begin + \delta_2^2 \geq A.end); \delta_i^k > 0$
B endin (#, #) A	$(B.end \leq A.debut + \delta_1^1 \text{ And } B.end \geq A.debut + \delta_1^2) \text{ And } (B.end \geq A.end + \delta_2^1 \text{ And } B.end \leq A.end + \delta_2^2); \delta_i^k > 0$
B cross (#, #) A	$(B.begin + \delta_1^1 \leq A.end \text{ And } B.begin + \delta_1^2 \geq A.end) \text{ And } (B.end \geq A.begin + \delta_2^1 \text{ And } B.end \leq A.begin + \delta_2^2); \delta_i^k > 0$
B overlaps (#, #, #) A	$(B.begin + \delta_1^1 \leq A.begin \text{ And } B.begin + \delta_1^2 \geq A.begin) \text{ And } (B.end \geq A.begin + \delta_2^1 \text{ And } B.end \geq A.begin + \delta_2^2) \text{ And } (B.end + \delta_3^1 \leq A.end \text{ And } B.end + \delta_3^2 \geq A.end); \delta_i^k > 0$

Fig 3. Traduction des relations temporelles en contraintes

## 4.1.2 Traduction des relations spatiales

La figure 4 ci-dessous donne la traduction des relations spatiales en contraintes.

Relation spatiale	Traduction en Contraintes
B left ( $\#_x, \#_y$ ) A	A.Gauche $\geq$ B.Droite + $\delta^1_1$ And A.Gauche $\leq$ B.Droite + $\delta^1_2$ And B.sup $\geq$ A.sup + $\delta^2_1$ And B.sup $\leq$ A.sup + $\delta^2_2$
B above ( $\#_x, \#_y$ ) A	B.Droite $\geq$ A.Gauche + $\delta^1_1$ And B.Droite $\leq$ A.Gauche + $\delta^1_2$ And B.sup $\geq$ A.sup + $\delta^2_1$ And B.sup $\leq$ A.sup + $\delta^2_2$
B right_top ( $\#_x, \#_y$ ) A	B.Inf + $\delta^2_1 \leq$ A.Sup And B.Inf + $\delta^2_2 \geq$ A.Sup And B.Droite $\geq$ A.Droite + $\delta^2_1$ And B.Droite $\leq$ A.Droite + $\delta^2_2$
B left_top ( $\#_x, \#_y$ ) A	B.Sup $\geq$ A.Inf + $\delta^2_1$ And B.sup $\leq$ A.Inf + $\delta^2_2$ And B.Droite $\geq$ A.Droite + $\delta^2_1$ And B.Droite $\leq$ A.Droite + $\delta^2_2$
B left_shift ( $\#_x$ ) A	B.Droite $\geq$ A.Gauche + $\delta^1_1$ And B.Droite $\leq$ A.Gauche + $\delta^1_2$ And B.Inf + $\delta^2_1 \leq$ A.Sup And B.Inf + $\delta^2_2 \geq$ A.Sup
B right_shift ( $\#_x$ ) A	B.Droite $\geq$ A.Gauche + $\delta^1_1$ And B.Droite $\leq$ A.Gauche + $\delta^1_2$ And B.Sup $\geq$ A.Inf + $\delta^2_1$ And B.sup $\leq$ A.Inf + $\delta^2_2$
B vertical_repartition ( $\#_y$ ) A	A.Gauche $\geq$ B.Droite + $\delta^1_1$ And A.Gauche $\leq$ B.Droite + $\delta^1_2$ And B.Inf + $\delta^2_1 \leq$ A.Sup And B.Inf + $\delta^2_2 \geq$ A.Sup
B right_vertical_Centring ( $\#_x$ ) A	A.Gauche $\geq$ B.Droite + $\delta^1_1$ And A.Gauche $\leq$ B.Droite + $\delta^1_2$ And B.Sup $\geq$ A.Inf + $\delta^2_1$ And B.sup $\leq$ A.Inf + $\delta^2_2$
B left_vertical_Centring ( $\#_x$ ) A	B.Gauche $\geq$ A.Gauche + $\delta^1_1$ And B.Gauche $\leq$ A.Gauche + $\delta^1_2$
B total_centrering A	B.Droite + $\delta^1_1 \geq$ A.Droite And B.Droite + $\delta^1_2 \leq$ A.Droite
B right ( $\#_x, \#_y$ ) A	B.Sup + $\delta^2_1 \leq$ A.Sup And B.Sup + $\delta^2_2 \geq$ A.Sup
B below ( $\#_x, \#_y$ ) A	B.Inf $\geq$ A.inf + $\delta^2_1$ And B.Inf $\leq$ A.inf + $\delta^2_2$
B right_bottom ( $\#_x, \#_y$ ) A	B.Hmilieu = A.Hmilieu And A.Gauche $\geq$ B.Droite + $\delta^1_1$ And A.Gauche $\leq$ B.Droite + $\delta^1_2$
B left_bottom ( $\#_x, \#_y$ ) A	B.Hmilieu = A.Hmilieu And B.Droite $\geq$ A.Gauche + $\delta^1_1$ And B.Droite $\leq$ A.Gauche + $\delta^1_2$
B top_shift ( $\#_y$ ) A	B.Vmilieu = A.Vmilieu And B.Inf + $\delta^2_1 \leq$ A.Sup And B.Inf + $\delta^2_2 \geq$ A.Sup
B bottom_shift ( $\#_y$ ) A	B.Vmilieu = A.Vmilieu And B.Sup $\geq$ A.Inf + $\delta^2_1$ And B.sup $\leq$ A.Inf + $\delta^2_2$
B horizontal_repartition ( $\#_x$ ) A	B.Hmilieu = A.Hmilieu And B.Vmilieu = A.Vmilieu
B bottom_horizontal_centrering ( $\#_y$ ) A	B.Droite $\geq$ A.Gauche + $\delta^1_1$ And B.Droite $\leq$ A.Gauche + $\delta^1_2$ And C.Droite $\geq$ B.Gauche + $\delta^1_1$ And C.Droite $\leq$ B.Gauche + $\delta^1_2$
B top_horizontal_centrering ( $\#_y$ ) A	B.Sup $\geq$ A.Inf + $\delta^2_1$ And B.sup $\leq$ A.Inf + $\delta^2_2$ And C.Sup $\geq$ B.Inf + $\delta^2_1$ And C.sup $\leq$ B.Inf + $\delta^2_2$

Fig 4. Traduction des relations spatiales en contraintes

### **4.1.3 Traduction des informations intrinsèques**

En plus de la traduction des relations temporelles et spatiales en contraintes, et pour s'assurer de la cohérence des informations intrinsèques des médias, un ensemble de contraintes est ajouté au système. Pour un média A, nous avons les contraintes suivantes :

#### **4.1.3.1 Contraintes temporelles intrinsèques**

$$A.end = A.begin + A.dur$$

$$A.MaxDur \geq A.NomDur$$

$$A.MinDur \leq A.NomDur$$

#### **4.1.3.2 Contraintes spatiales intrinsèques**

$$A.width = A.right - A.left$$

$$A.height = A.bottom - A.top$$

$$A.Hmilieu = (A.top + A.bottom) / 2$$

$$A.Vmilieu = (A.left + A.right) / 2$$

$$A.left \geq 0$$

$$A.top \geq 0$$

$$A.right \leq \text{screenwidth (largeur de l'espace de travail)}$$

$$A.bottom \leq \text{screenheight (hauteur de l'espace de travail)}$$

## 5. Mise en œuvre

La partie d'édition et de présentation de MediaStudio est construite avec visuel C++/MFC, pour la partie résolveur, nous avons effectué une traduction de sa version originale en GCC vers le langage C++. Dans ce qui suit, nous allons présenter l'essentiel de la mise œuvre du système.

### 5.1 Interface utilisateur de MediaStudio

L'interface d'édition et de présentation de MediaStudio (Figure 5) est composée d'une fenêtre principale organisée sous forme d'un ensemble de *widgets* permettant à l'utilisateur de réaliser les fonctions d'édition et de présentation. À travers cette interface, l'utilisateur peut créer, charger un document, le modifier ou pré-visualiser la construction de son document (document primaire et parcours de lecture). Enfin, il peut lancer et interagir avec la présentation de son document.

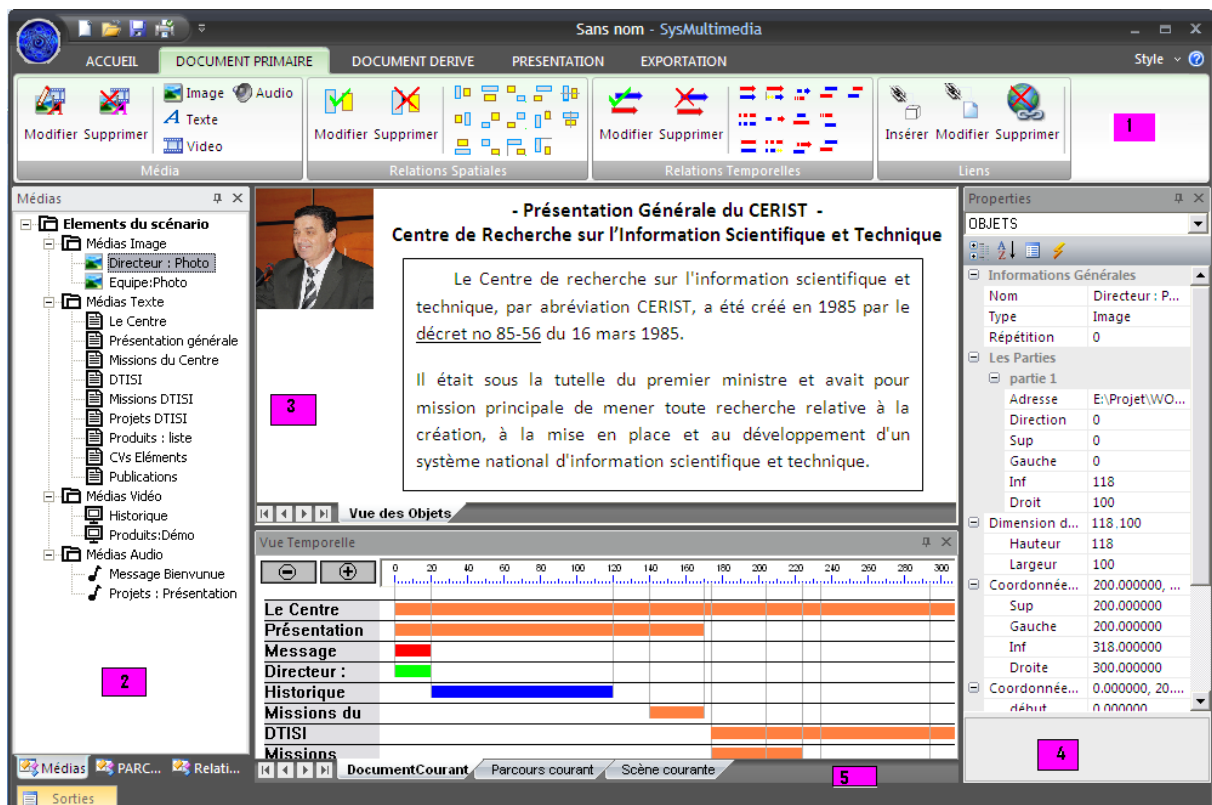


Fig 5. Interface utilisateur de MediaStudio

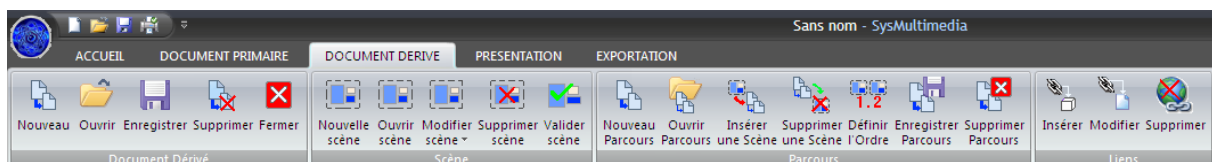


Fig 6. Anglet document dérivé

La fenêtre principale est composée de cinq zones (numérotées sur la figure 5) qui offrent les fonctions d'édition suivantes :

Zone 1 : Elle est composée d'un panneau regroupant cinq onglets principaux :

- Accueil : activé par défaut, il comprend les fonctions de création, d'ouverture, de sauvegarde d'un document et de sortie d'une session.
- Document primaire : il regroupe l'ensemble de boutons et des formulaires permettant l'insertion, la modification ou la suppression de médias, de relations et de liens.
- Document Dérivé : il regroupe l'ensemble de boutons permettant de créer, d'ouvrir ou de modifier un document dérivé. De créer, d'ouvrir, de sauvegarder ou de modifier un parcours de lecture.
- Présentation : Permet de lancer une prévisualisation ou la présentation d'un document (primaire ou parcours). La présentation détaillée de cette partie sera traitée dans le chapitre suivant.
- Exportation : permet de sauvegarder, sous un format XML valide-schéma, un document primaire ou un parcours de lecture avec une visionneuse (application permettant la présentation du document sur une autre machine).

Zone 2 : Il y est affiché, sous forme d'arborescence, les médias, les relations et les parcours de lecture en cours d'édition. Elle permet de sélectionner un élément pour éventuellement le supprimer ou pour modifier ses attributs qui s'affichent au niveau de la zone 4. À ce niveau, l'auteur dispose également de la possibilité de modifier les attributs d'un élément à travers une boîte accessible via un menu textuel.

Zone 3 : Une surface de restitution graphique et d'édition du média en cours d'édition ou du média sélectionné dans la zone 2 avec l'ensemble des médias se présentant durant sa période de présentation. L'auteur peut y sélectionner un média pour modifier ses attributs, le déplacer ou le supprimer.

Zone 4 : Elle affiche l'ensemble des attributs d'un média sélectionné où l'auteur peut modifier directement ses informations.

Zone 5 : C'est la zone d'affichage de la vue temporelle globale d'un document où l'auteur a un aperçu de l'enchaînement temporel de son document qu'il peut modifier.

## 5.2 L'édition

Dans notre système, deux types d'utilisateurs peuvent interagir à ce niveau avec le système. L'auteur d'un document qui va s'intéresser au contenu et à l'organisation de son document, et le lecteur qui va s'intéresser à la création de ses parcours de lecture.

À partir de ces deux facettes, nous pouvons considérer, pour l'auteur, que le processus d'édition d'un document est une combinaison des sept fonctions suivantes :

- Saisie et structuration : cette fonction permet d'insérer de nouveaux médias dans le document et à organiser son contenu.
- Composition temporelle : elle consiste à définir les relations temporelles entre les médias du document.
- Placement spatial : cette opération permet de spécifier le positionnement géométrique des médias en termes de relations spatiales.
- Mise en place d'une structure de navigation : c'est la fonction qui permet de définir la dimension hypermédia du document à travers des mécanismes de sauts temporels et de références.
- Définition d'un document dérivé : elle consiste à choisir un ensemble de médias du document primaire et à les regrouper en un document dérivé.
- Définition des scènes : elle consiste à regrouper, selon les conditions prédéfinies, un ensemble de médias d'un document dérivé en scènes.
- Composition du parcours de lecture : elle permet de choisir un ensemble de scènes et à les ordonnancer temporellement dans le respect des conditions prédéfinies.

L'ordre d'accès à ces fonctions est totalement libre pour l'auteur.

Pour le lecteur, à partir d'un document primaire, il dispose des trois fonctions suivantes :

- Définition d'un document dérivé : elle consiste à choisir un ensemble de médias du document primaire et à les regrouper en un document dérivé.
- Définition des scènes : c'est à ce niveau que l'utilisateur procède au regroupement, selon les conditions prédéfinies, des médias d'un document dérivé en scènes.
- Composition du parcours de lecture : elle consiste à choisir un ensemble de scènes et à les ordonner temporellement dans le respect des conditions prédéfinies.

Les différentes fonctions d'édition et de présentation sont rendues accessibles à travers l'interface utilisateur décrite ci-dessous.

## 5.3 Vérification de la cohérence

### 5.3.1 Le résolveur cassowary

L'une des caractéristiques retenue pour le choix du résolveur Cassowary est le fait qu'il puisse, indifféremment, modéliser sous forme de contraintes et résoudre les problèmes issus aussi bien des modules de formatage spatial, temporel que du gestionnaire du recouvrement.

Dans ce qui suit, nous donnons l'implémentation adoptée pour le résolveur ainsi que les révisions qui lui ont été apportées.

L'implémentation de tous les modules du système respecte une architecture en classes. On en distingue principalement les classes contraintes des systèmes disjonctifs, des formateurs temporel et spatial ainsi que la classe du résolveur.

Dans ce qui suit, nous allons donner les principales procédures utilisées au niveau du résolveur.

#### 5.3.1.1 Procédures principales du résolveur

- `AddConstraint(CContrainte*constraint)` : Ajout incrémental de contrainte linéaire à la table de contraintes. La classe de contraintes prévoit un champ spécifiant la force (le poids) de la contrainte.
- `RemoveConstraint (CContrainte*constraint)` : Suppression de la contrainte de la table de contraintes.
- `AddEditVar(CVariable v, CForce&strength)` : Ajout d'une contrainte *edit* sur la variable *v* de poids *strength* à la table. Ainsi, la méthode `suggestValue()` peut être utilisée sur cette variable après un `BeginEdit()`.
- `RemoveEditVar(CVariable v)` : Suppression de la variable *edit* *v* de la table. L'appel de la fonction `EndEdit()` supprime automatiquement toutes les variables *edit* et signale la fin d'une opération d'édition.
- `BeginEdit()` : Préparation de la table pour recevoir les nouvelles valeurs des variables d'édition courantes. La fonction `AddEditVar()` doit être utilisée avant l'appel de la fonction `BeginEdit()`. `SuggestValue()` et `Resolve()` doivent être utilisées uniquement après `BeginEdit()` et avant `EndEdit()`.
- `SuggestValue(CVariable v, double value)` : spécifier la valeur *value* désirée pour la variable *v*. Avant que l'on puisse faire appel à `SuggestValue()`, la variable *v* doit au préalable être définie comme étant celle d'une contrainte *edit* (soit via `AddConstraint()` ou `AddEditVar()`).
- `EndEdit()` : Signifie la fin d'une opération d'édition marquant la suppression de toutes les contraintes d'édition de la table.

- `Resolve()` : Essaye de résoudre la table des contraintes avec les nouvelles valeurs spécifiées. Les appels à cette méthode doivent être inclus entre un `BeginEdit()` et un `EndEdit()` et après que tous les appels à `SuggestValue()`.
- `AddStay(CVariable v)` : Exprime le désir de voir la valeur de la variable  $v$  inchangée et du fait, définit une contrainte stay de poids faible sur  $v$ .
- `SetAutoSolve(boolean f)` : permet de choisir le mode de fonctionnement du résolveur. Quand le paramètre est à vrai, la résolution et l'optimisation seront automatiques à chaque ajout ou retrait de contrainte. Par défaut, cette fonction est activée à moins que l'on fasse appel à cette fonction avec le paramètre à faux et alors, il faudra appeler explicitement `Solve()` ou `Resolve()` pour la résolution et l'optimisation.
- `Solve()` : Optimise la table de contraintes (la fonction `Optimize()`) et assigne aux variables des contraintes leurs nouvelles valeurs (la fonction `SetExternalVariables()`). Cette fonction n'est appelée qu'en cas de résolution non automatique du résolveur.
- `Reset()` : Réinitialise le résolveur à partir des contraintes initiales.
- `AddDisjSystem(CSystemeDisjonctif*sysDisj)` : Ajoute un système disjonctif à la table des systèmes disjonctifs. Cette fonction active les contraintes du système qui sont d'emblée satisfaites en essayant de les ajouter une à une dans la table de contraintes.
- `RemoveDisjSystem(CSystemeDisjonctif*const pdisj)` : Enlève le système disjonctif de la table des systèmes disjonctifs et aussi toutes les contraintes qui lui sont associées et qui sont activées et donc, se trouvant dans la table des contraintes.
- `ActiveNewConstraint(CSystemeDisjonctif*pdisj)` : Active une nouvelle contrainte lorsque la contrainte actuellement ne peut plus être satisfaite.
- `DisjSolve()`: Résout la table des systèmes disjonctifs en cherchant puis en insérant pour chaque système disjonctif la contrainte qui satisfait le système global.

### 5.3.2 Vérification de la cohérence des relations

Elle se fait après chaque opération d'édition pouvant affecter la cohérence du document ou à tout moment à la demande de l'auteur. Aucune distinction n'est faite entre la vérification des relations temporelles et des relations spatiales, encore une fois un haut niveau d'intégration est assuré à ce niveau.

La figure 7 suivante donne la représentation détaillée du processus de vérification de la cohérence temporelle/spatiale.

Le système commence par récupérer les attributs temporels/spatiaux des médias à partir du gestionnaire d'objets et les traduit en variables du résolveur. Ensuite, il traduit les relations temporelles/spatiales spécifiées par l'auteur et les relations intrinsèques des médias en contraintes. Après avoir généré le système de contraintes, le résolveur Cassowary essaye de trouver une solution. Dans le cas où le système est cohérent, les attributs temporels/spatiaux des médias sont mis à jour. Dans le cas contraire, un message d'erreur est affiché pour l'auteur.

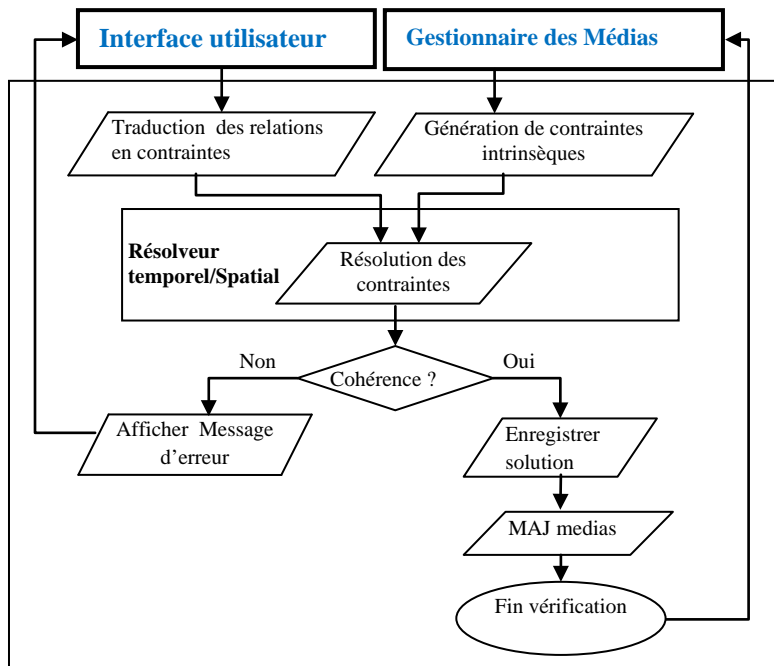


Fig 7. Procédure de vérification de la cohérence des relations

### 5.3.3 Vérification de la cohérence des parcours de lecture

La vérification de la cohérence des parcours de lecture passe par celle des scènes. Le système commence par vérifier que quelque soient deux médias d'une scène, ils doivent être liés par une relation directe ou indirecte. Si toutes les scènes vérifient cette condition alors le système s'assure que l'ordre d'enchaînement des scènes du parcours est le même que celui du document primaire.

## 5.4 Gestion du recouvrement spatial

Une des extensions et améliorations qui sont venues enrichir le système de résolution de Cassowary est incontestablement la prise en charge des systèmes disjonctifs.

Au niveau du solveur, une table particulière est maintenue pour sa gestion. À l'ajout d'un système, il crée une nouvelle entrée dans cette table et y sauvegarde la structure du système. D'autre part et parallèlement à cet ajout, il identifie la contrainte initiale active et lui crée une nouvelle entrée dans la table des contraintes du système et résout puis optimise la table. En cas d'échec à l'un des niveaux du traitement, le système disjonctif ne sera pas ajouté et la contrainte est retirée de la table des contraintes.

### 5.4.1 Procédures de mise en œuvre

Dans un souci du respect des spécifications de l'auteur et de l'intégrité des médias ainsi que d'optimisation des performances temporelles de l'approche, il est clair que la prise en compte des solutions proposées (tolérance, réduction des dimensions, etc.) doit non seulement obéir à un ordre d'exécution bien établi mais également qu'elles soient le dernier recours après l'échec d'une solution d'emplacement. Compte tenu de tout cela nous proposons pour la gestion du recouvrement l'algorithme suivant.

Soit B le média objet de l'opération d'édition, la procédure *Gestion\_Recouvrement (B,A)* est appelée avec le média A si et seulement si :

intervalle temporel (B)  $\cap$  intervalle temporel (A)  $\neq 0$  ET recouvrement (B, A) non désiré.

```

Procédure Gestion_Recouvrement (B,A)
{
  If (recouvrement (B,A))
  {
    Initialisation_sys_cont_disj(B,A) ;
    If (Echec_initialisation)
    {
      If ((c1 OR c2 OR c3 OR c4) <> 0) Tolerance (B,A) ;
      If (Echec =vrai) Reduction_taille (B,A) ;
      If (Echec=vrai)
      {
        Modif_attribut_temporels (B,A) ;
        Resolution_sys_tempo() ;
        If (non_coherence) Then Echec :=Vrai
      }
      If (Echec =vrai) Message (' le recouvrement ne peut être éliminé');
    }
    Else
    {
      If (Non_sys_cont_disj (B,A) ) Generation_sys_cont_disj (B,A) ;
      Resolution_Disj(C,active[]) ;
    }
  }
  Else
  {
    If ((∃ intervalles_communs (B,A)) AND (∃ sys_cont_disj (B,A))) Resolution_Disj(C,active[]);
    If ((∃ intervalles_communs(B,A)) AND (Non_sys_cont_disj(B,A)))
    {Generation_sys_cont_disj (B,A); Resolution_Disj(C,active[])};
    If ((Non intervalles_communs(B,A)) AND (∃ sys_cont_disj )) Supprimer_sys_cont_disj(B,A);
  }
}

```

**Resolution\_Disj (C, active[])**

```

{
  termine:=Faux;
  Repeat
  {
    If (termine ==Faux) Solution = resolveur (C0 ∧ D1active[1] ∧ ... ∧ Dnactive[n]);
    termine =True;
    If (Solution == echec)
    {
      For (i=1; i<=n; i++)
      {
        Indice_courante = active[i];
        Activer_nouvelle_contrainte (Di,courante,liste_contraintes_testees);
        liste_contra_testées = liste_contraintes_testées + active[i] ;
        If (active[i]!= indice_courant) {termine = Faux; Break;}
      }
    }
  }Until termine ;
  Return(Solution); /* Echec ou réussite*/
}

```

**Activer\_nouvelle\_contrainte(D<sub>i</sub>,courante, liste\_contraintes\_testees)**

```

{
  Soit Di de la forme C1 ∨... ∨ C8
  For ( k=1 ;k<=8;k++)
  If ((Ck!= courante) AND (Ck ∉ liste_contraintes_testees ))
    Return(k) endif ;
  Return(indice_courant) ;
}

```

**Tolerance (A,B)**

```

{
  Echec = Vrai ;
  If (∃ sys_cont_disj(A,B) )
  {
    Generation sys_contraintes_disj(A,B,Ci,i=1-4) ;
    Resolution_Disj(C,active[]) ;
    If (reussite) Echec =Faux ;
  }
  Else
  {
    Generation_sys_cont_disj (A,B) ;
    For (i=1;i<=8;i++)
    {
      Activer_contrainte(i);
      Resolution_Disj(C,active[])
      If (reussite) { Echec =Faux; Break ;}
    }
  }
}

```

```

Reduction_taille (A,B)
{
  For (j=1;i<=3;j++)
  {
    Choisir_Media() ;
    While (limite non atteinte)
    {
      Reduire_taille(Media);
      If ( $\exists$  sys_cont_disj(A,B))
      {
        Resolution_Disj(C,active[]);
        If (reussite) { Echec =Faux ; Break;}
      }
      Else
      {
        Generation_sys_cont_disj (A,B) ;
        For (i=1;i<=8;i++)
        {
          Activer_contrainte(i);
          Resolution_Disj(C,active[])
          If (reussite) {Echec =Faux ; Break ;}
        }
      }
      If (reussite ) Break ;
    }
    If (reussite ) Break ;
  }
}

```

Lorsqu'un recouvrement est détecté, la procédure *Initialisation\_sys\_cont\_disj(A,B)* est appelée pour créer, dans le cas où il n'existe pas, le système de contraintes disjonctives entre A et B et d'activer l'une des ses contraintes; ou alors pour mettre à jour le système existant en activant la contrainte courante si celle-ci satisfait toujours la solution sinon pour activer une autre.

Dans le cas de la création d'un nouveau système, la procédure crée également une contrainte d'inégalité de façon à déplacer l'un ou les deux médias pour éliminer le recouvrement. Si le recouvrement est éliminé alors la contrainte ajoutée est supprimée, le système associé est généré et une de ses contraintes est activée. Par la suite, ce système est ajouté au système global C.

La résolution du système de contraintes global C est assurée par la procédure *Resolution\_Disj(C, active[])*. Elle prend en entrée C, composé de l'ensemble des contraintes conjonctives issu des relations spatiales et intrinsèques des médias, et des ensembles de contraintes disjonctives issus de  $D_1...D_n$  et du tableau *active[]*. Celui-ci est composé des indices des contraintes actives, tel que pour chaque disjonction  $D_i$ , *active[i]* correspond à l'indice de la ou des deux contraintes actives dans  $D_i$ . Il varie entre 1 et 8, nombre qui correspondant aux quatre contraintes du système disjonctif et aux quatre de leurs combinaisons significatives (non contradictoires parmi les combinaisons possibles des contraintes disjonctives).

La procédure *Activer\_nouvelle\_contrainte(D\_i,courante,liste\_contraintes\_testées)* est appelée pour activer la contrainte actuelle ou une nouvelle dans le cas où l'actuelle ne satisfait plus le système. Dans ce dernier cas la résolution du système est exécutée. Ce processus est itéré jusqu'à aboutir à une solution du système (placement du média) ou à une déclaration d'un échec (recouvrement persistant). Par contre, s'il y'a échec d'initialisation, autrement dit le recouvrement ne peut pas être éliminé par déplacement des médias, on fait alors appel à la procédure *Tolerance(A,B)* si au moins un des coefficients de tolérances de A est différent de zéro. Elle régénère le système de contraintes disjonctives, lorsqu'il existe, avec les coefficients de tolérance préalablement introduits et fait appel à la procédure de résolution du système global. Lorsque le système de contraintes disjonctives n'existe pas, il est généré avec les coefficients de tolérance.

La procédure *Activer\_contrainte(i)* est exécutée dans l'ordre, avec une des contraintes ou une de leurs combinaisons, jusqu'à trouver celle qui le satisfait ou bien un échec est signalé une fois toutes les contraintes et leurs combinaisons testées.

Lorsque la procédure *Tolerance(A,B)* n'arrive pas à éliminer le recouvrement, on fait appel à la procédure *Reduction\_taille(B,A)* qui réduit progressivement, dans les limites permises, respectivement la taille de B de A ou de A et B. Suite à quoi la procédure *Resolution\_Disj(C, active[])* est appelée dans le cas où le système de contraintes disjonctives existe. Par contre s'il n'existe pas, il est alors généré et la procédure *Activer\_contrainte(i)* est exécutée.

Enfin, si le recouvrement n'est pas éliminé jusqu'à ce niveau, la procédure *Modif\_attribut\_temporels(A,B)*, qui décale respectivement le début du média B ou de A ou les deux, de façon à rendre leurs intervalles de présentation disjoints, est exécutée. La validation de la cohérence temporelle est alors invoquée.

Si aucun de ces mécanismes ne permet d'éliminer le recouvrement alors l'auteur en sera informé.

Maintenant, si l'opération d'édition n'engendre pas de recouvrement spatial, la procédure de gestion du recouvrement est exécutée pour maintenir le scénario dans un état de non recouvrement.

## **6. Conclusion**

La réalisation de MediaStudio a comporté une part importante de développement pour le mettre en œuvre. Elle a contribué au raffinement des modèles et des approches adoptées et d'une manière globale à toute la réflexion.

L'originalité du système MediaStudio se situe à plusieurs niveaux : d'abord il constitue une réponse quasi-globale à tous les problèmes de l'édition et de la présentation d'un document multimédia interactif en étant d'emblée conçu dans cette perspective. L'incrémentalité des gestionnaires temporel et spatial, qui permettent à l'auteur d'être avisé à chaque opération d'édition de la validité des opérations qu'il effectue.

Son architecture modulaire a été définie de manière à faciliter l'expérimentation de divers algorithmes intervenant depuis la construction jusqu'à la présentation d'un document multimédia. Son utilisation n'exige aucun pré-requis de formation ou de connaissances particulières.

# Chapitre 8

## Systeme de présentation multimédia

### 1. Introduction

Avec ce chapitre, nous arrivons à l'ultime phase de la mise en œuvre de notre SEPDMI. La présentation est l'action qui consiste à restituer le contenu d'un document multimédia dans le respect des spécifications de l'auteur. La prise en compte de ces spécifications soulève de nombreux problèmes dans la réalisation des fonctions du système de présentation comme la synchronisation et la gestion de l'indéterminisme (interactions utilisateurs).

Ce système utilise la représentation interne du document, structurée sous forme d'un graphe d'exécution et produite par le système d'édition, pour restituer à l'utilisateur le contenu du document à travers les dispositifs de la machine comme l'écran et le haut parleur.

### 2. Caractéristiques d'une présentation multimédia

Nous présentons ici les principales propriétés qui caractérisent les présentations multimédia. Ces caractéristiques doivent être prises en charge lors de la présentation d'un document [57].

#### 2.1. Hétérogénéité des médias

Les médias ont des caractéristiques hétérogènes dues à la différence de leurs natures (Vidéo, audio, image et texte). Celles-ci peuvent être classées selon un certain nombre de critères dont les plus importants sont :

- **Le mode de perception de l'information** : Il spécifie la manière par laquelle l'utilisateur perçoit les médias. Ils peuvent être audibles ou visible.
- **Le comportement temporel** : la façon selon laquelle les médias se comportent temporellement pendant leur présentation. On peut identifier deux types de comportements :
  - *Médias discrets* : Médias n'ayant pas de durées intrinsèques comme le texte et les images.
  - *Médias continus* : Médias constitués d'un flot continu de données restitué pendant une durée bien déterminée qui est la *durée calculée* de ce média.

#### 2.2. Dynamicité de la présentation multimédia

Contrairement à la présentation d'un document statique où la lecture est linéaire, la présentation d'un document multimédia peut avoir plusieurs traces d'exécution dues aux interactions utilisateur qui peut arrêter, avancer, reculer et sauter temporellement à travers le document multimédia.

#### 2.3. Synchronisation multimédia

Pour rappel, un document multimédia se distingue d'un document classique par la prise en compte de la dimension temporelle. Cette dimension peut être perçue à un niveau individuel indépendant pour les médias continus, et peut être aussi considéré à un niveau commun en

prenant en compte le déroulement logique de la présentation, et éventuellement l'interaction avec l'utilisateur.

La synchronisation est une caractéristique très importante dans les applications multimédia. Elle définit la sémantique de la présentation. Deux types de synchronisation peuvent être définis.

### **2.3.1. Synchronisation inter médias**

Concerne le positionnement temporel d'un média par rapport aux autres. C'est cette information qui est généralement spécifiée par l'auteur d'un document multimédia. Cette synchronisation est définie sous forme de relations temporelles entre les instants de début ou de fin des médias.

### **2.3.2. Synchronisation environnementale**

Ce type de synchronisation définit le comportement de la présentation vis-à-vis des actions de l'utilisateur. Ce dernier agit par le biais de liens ou d'interfaces de contrôle de la présentation. Par exemple, cette synchronisation décrit quel comportement doit avoir l'application si l'utilisateur fait une pause, un avancement rapide ou un redémarrage du scénario.

## **3. Fonctions du système de présentation multimédia**

À partir des caractéristiques identifiées ci-dessus, nous présentons ici les différentes fonctions permettant leur prise en charge :

- Gestion de l'ordonnancement et de la synchronisation multimédia : La synchronisation consiste à exécuter la spécification de l'auteur en respectant les contraintes et les délais.
- Gestion des médias : Permet le traitement homogène (création, lancement et arrêt) de tous les types des médias.
- Gestion des interactions utilisateur : Elle consiste à gérer les différents types de navigation et de manipulation d'un document offerts au lecteur.

### **3.1. Architecture du système de présentation**

Les différentes fonctions définies ci-dessus sont organisées autour de trois modules principaux comme décrit dans la figure 1 suivante :

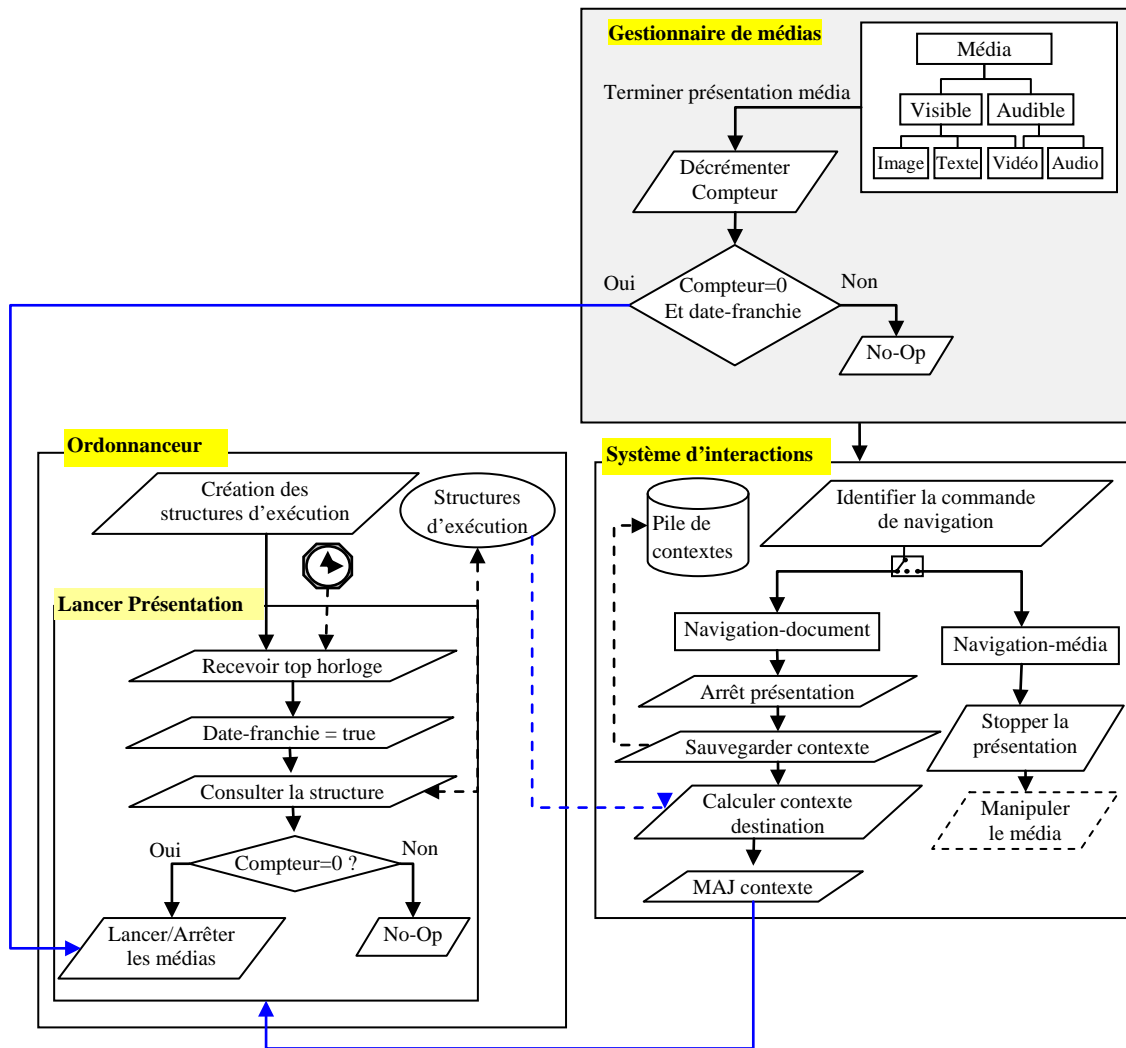


Fig 1. Architecture du système de présentation.

### 3.2. L'ordonnanceur

L'ordonnanceur est le module qui se charge du lancement et de l'arrêt d'un média à l'arrivée de son instant de début ou de fin. C'est lui qui décide quel média ou quel ensemble de médias doit être lancé/arrêté à un instant donné. L'ordonnanceur utilise l'ensemble des informations fournies par la structure d'exécution et l'horloge pour décider quelle action effectuer à chaque instant *significatif* de la présentation. Un instant significatif correspond à un instant de la présentation où au moins un média démarre ou termine sa présentation.

#### 3.2.1. La structure d'exécution

Utilisée par l'ordonnanceur, cette structure est organisée sous forme d'un graphe (fig 2.) où :

- Les nœuds représentent les instants significatifs de la présentation.
- Les arcs modélisent les médias. La position de l'arc dans le graphe donne l'ordonnancement temporel du média correspondant par rapport aux autres médias.

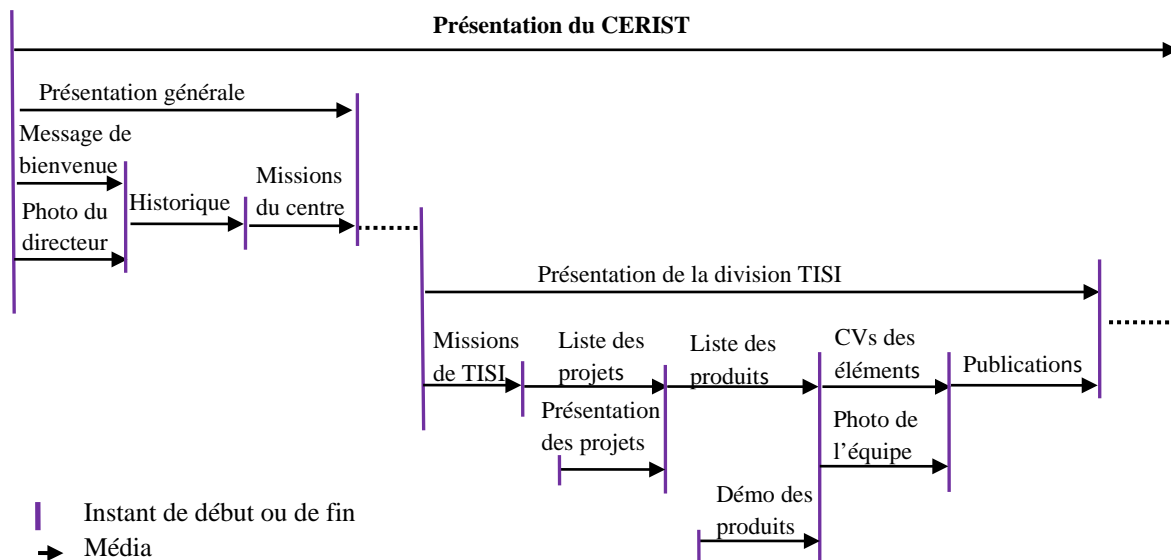


Fig 2. Exemple de Structure d'exécution

Pour chaque nœud du graphe nous avons retenu les informations suivantes :

- La liste des arcs entrants dans ce nœud.
- La liste des arcs sortants de ce nœud.
- La date de franchissement de ce nœud : elle correspond à la date de début des médias symbolisés par les arcs sortants et la date de fin des médias symbolisés par les arcs entrants.
- Des pointeurs vers les nœuds précédents et suivants.
- Un compteur de synchronisation initialisé au nombre de médias se terminant à ce nœud. Lorsqu'un média entrant se termine, le compteur est décrémenté d'une unité. Ainsi, le nœud n'est franchi que lorsque la valeur du compteur est à zéro, c'est-à-dire, que tous les médias entrants ont terminé leurs présentations.
- Un booléen *date-franchie* qui indique si la date de franchissement du nœud est arrivée. Il est initialisé à faux.

Chaque arc du graphe contient les informations suivantes :

- Le nom de l'arc qui est celui du média représenté.
- Un pointeur vers son nœud origine.
- Un pointeur vers son nœud destination,
- La durée de l'arc qui est celle du média.

Pour la génération de la structure d'exécution d'un parcours de lecture, on commence par la génération des structures de chaque scène du parcours de la même manière que précédemment, par la suite elles seront concaténées en fonction de l'ordre de leurs définitions dans le parcours de lecture.

Dans tout ce qui suit, aucune différence n'est faite entre un document primaire et un parcours de lecture, ils sont tous les deux désignés par le terme document.

### 3.2.2. Fonctionnement de l'ordonnanceur

Le fonctionnement de l'ordonnanceur est illustré par la figure 3 en suivant les étapes numérotées ci-dessous de (1) à (8).

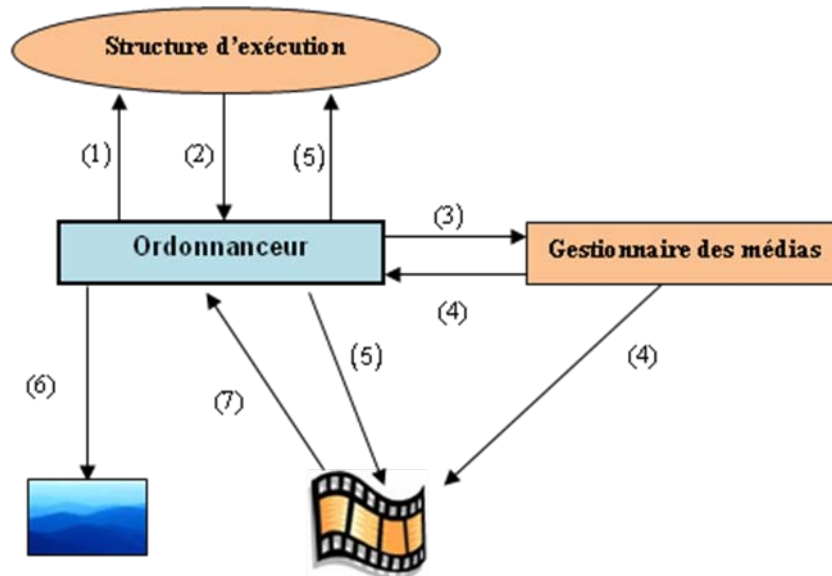


Fig 3. Fonctionnement de l'ordonnanceur

- 1) L'ordonnanceur consulte la structure d'exécution pour savoir quel est le nœud de début de la présentation.
- 2) La structure d'exécution retourne un pointeur vers le nœud de début.
- 3) L'ordonnanceur invoque le gestionnaire des médias pour créer les médias sortants de ce nœud.
- 4) Le gestionnaire des médias crée les médias demandés et retourne leurs identificateurs à l'ordonnanceur.
- 5) L'ordonnanceur envoie aux médias créés des commandes de démarrage. À l'arrivée de la fin d'un média, l'ordonnanceur vérifie le compteur de synchronisation du nœud d'arrivée.
- 6) Si le compteur est à zéro, l'ordonnanceur arrête les médias entrants, lance les médias sortants et réinitialise l'horloge globale.
- 7) Sinon, l'ordonnanceur attend un signal du média indiquant sa fin. Si ce nœud est le dernier du graphe, alors la présentation est arrêtée sinon les étapes (3), (4), (5) et (6) sont exécutées.

### 3.2.3. La gestion de la synchronisation

La présentation d'un document est une séquence d'actions déclenchée aux instants significatifs. Dans notre système, la gestion de la synchronisation est centralisée au niveau de l'ordonnanceur qui assure les actions à exécuter à chaque arrivée d'un instant significatif.

L'ordonnanceur utilise une horloge globale initialisée en se basant sur les informations fournies par la structure d'exécution. La période de l'horloge globale est calculée comme étant la différence entre la date du nœud suivant et celle du nœud courant. De plus, chaque média possède une horloge interne, lancée à son démarrage, qui scrute la durée de sa présentation.

À chaque top d'horloge globale, sa période est recalculée, un signal est envoyé à l'ordonnanceur pour exécuter la procédure *Synchroniser* ci-dessous, qui vérifie si les médias entrant ont terminé leurs présentations et ce en consultant le compteur de synchronisation.

```
Procédure Synchroniser ()
{
    Date-Franchie = true ;
    If (compteur == 0)
    {
        Arrêter-Médias (Liste-Médias-Entrants) ;
        Lancer-Médias (Liste-Médias-Sortants) ;
        Initialiser-Horloge-Globale () ;
    }
}
```

Lorsque la valeur du compteur est à zéro, les médias sortant sont lancés puis l'horloge globale est réinitialisée pour la prochaine étape. Sinon, à la fin de chaque média l'ordonnanceur est invoqué pour exécuter la procédure *Decrementer-Compteur* suivante.

```
Procédure Decrementer-Compteur ()
{
    Compteur = Compteur -1;
    If (compteur == 0 And Date-Franchie)
    {
        Gestionnaire-Sync .Lancer-Médias (Liste-Médias-Sortants) ;
        Gestionnaire-Sync .Initialiser-Horloge-Globale () ;
    }
}
```

### 3.3. Gestionnaire des médias

Le gestionnaire des média est invoqué par l'ordonnanceur afin de créer, de lancer, d'arrêter ou d'affecter des valeurs aux attributs d'un média.

Il gère une hiérarchie de classes d'objets média (figure 4) qui définissent les caractéristiques et les actions de présentation de chaque type de média.

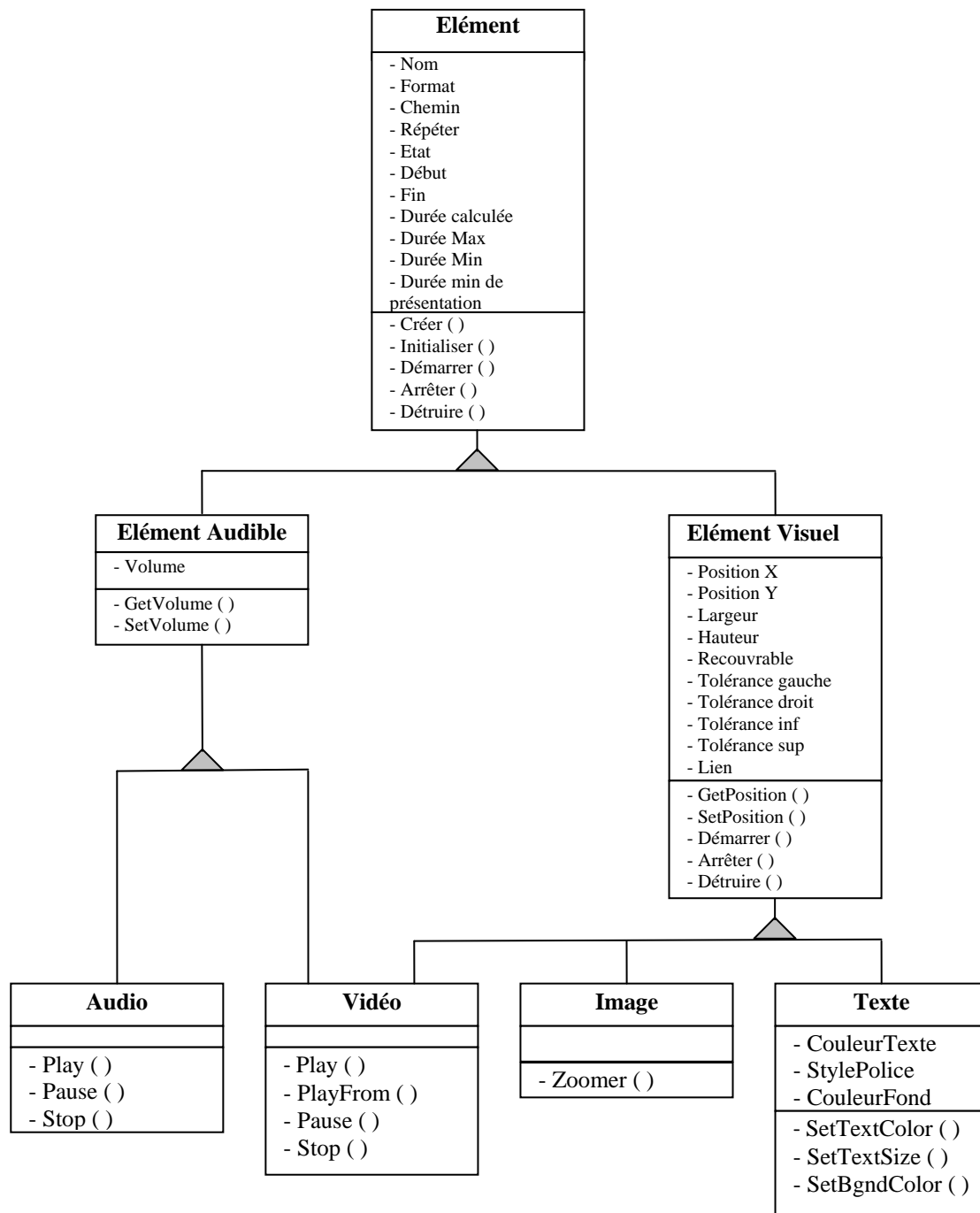


Fig 4. Hiérarchie de classes des médias

### 3.4. Gestionnaire des interactions

Dans notre système, deux types d'interactions sont considérés: la navigation et les opérations de manipulations. Le gestionnaire d'interactions a pour objectif de mettre en œuvre ces interactions. Pour cela, il doit savoir à tout instant de la présentation quels sont les médias actifs ainsi que leurs états d'avancement. Cette information est contenue dans une structure de données dénommée "*contexte de présentation*" [57].

### 3.4.1. Le contexte de présentation

Le contexte de présentation (CP) à un instant donné donne toutes les informations nécessaires pour reconstituer la présentation à partir de cet instant. Il est représenté par l'ensemble des médias actifs à cet instant et leurs état d'avancement par rapport au début de la présentation. Le gestionnaire des interactions utilise la structure d'exécution pour récupérer la liste des médias actifs à cet instant et calcule leurs états d'avancement pour construire le contexte de présentation. Pour l'exemple, prenons la structure d'exécution suivante.

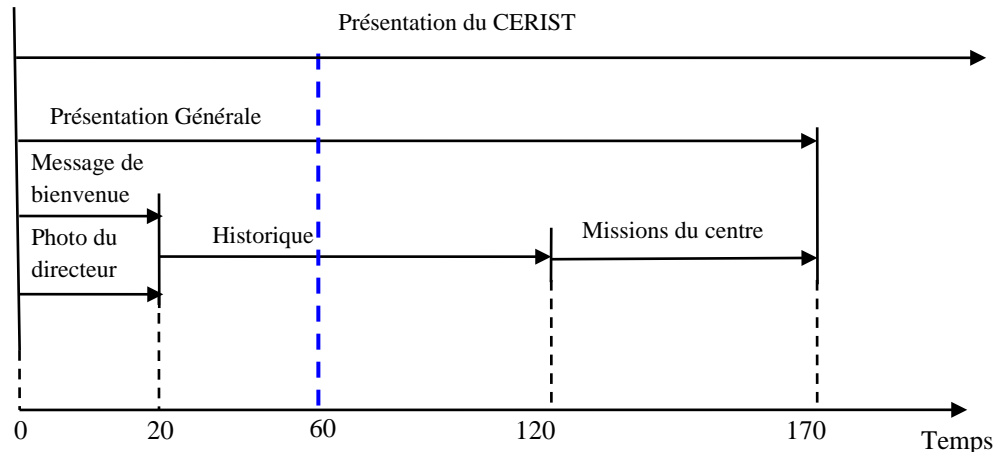


Fig 5. Exemple Contexte de Présentation

Le contexte de présentation à l'instant  $t=60$  sec est :

$CP_{t=60} = \{(Présentation\ du\ CERIST, 60), (Présentation\ générale, 60), (Historique, 40)\}$

### 3.4.2. Manipulation du contexte de présentation

Afin de réaliser les fonctions de navigation et d'effectuer des sauts temporels à travers le document, un ensemble d'actions doit être entrepris par le gestionnaire des interactions :

- Mise à jour du contexte de présentation : à chaque changement d'état d'au moins un média et à chaque opération de navigation, le contexte de présentation doit être mis à jour.
- Sauvegarde et restauration : lors de chaque saut temporel, le contexte de présentation doit être sauvegardé pour être restauré au retour de la destination. Les contextes sont sauvegardés dans une pile appelés *pile de contextes*. Ceci permettra, après plusieurs sauts successifs, de restaurer le dernier contexte sauvegardé.
- Calcul : à l'arrivée à destination, le gestionnaire des interactions détermine quels sont les médias actifs à cet instant, calcule leurs états d'avancement et constitue le contexte de destination.

### 3.4.3. Navigation

À ce niveau, deux types de navigations sont gérés.

#### 3.4.3.1. Navigation par liens

La destination d'un lien peut être :

- Intra-document (média, scène, instant du document),
- extra-document (ressource externe).

#### **- Lien intra-document**

La destination d'un lien intra-document se traduit toujours en un instant. Dans le cas d'un média ou d'une scène elle sera traduite en son instant de début. Dans ce cas, le module de navigation effectue les opérations suivantes :

1. Sauvegarde du contexte courant dans la pile du contexte de présentation.
2. Envoie d'une commande à l'ordonnanceur pour arrêter la présentation.
3. Calcul du contexte de présentation correspondant à la destination du lien (instant) en utilisant les informations de la structure d'exécution.
4. Mise à jour du contexte actuel de présentation en lui affectant le contexte de destination calculé.
5. Envoie d'une commande de lancement de la présentation à l'ordonnanceur.

Lorsque l'utilisateur clique sur le bouton retour de l'interface, le contexte sauvegardé est restauré et la présentation est reprise à partir de cet instant.

#### **- Lien extra-document**

Lorsque la destination d'un lien est un document externe, celui-ci est affiché dans une autre fenêtre indépendante de celle du document en cours qui est par conséquent mis en pause. Si le type de document référencé n'est pas reconnu par le système, l'utilisateur est invité à introduire l'application permettant sa lecture.

#### **3.4.3.2. Navigation libre**

À partir de l'interface utilisateur, le lecteur dispose d'une vue générale de son document composée de l'ensemble des médias ou des scènes. Il peut visualiser le média ou la scène de son choix en cliquant dessus. Ce processus est similaire à celui de la navigation extra document. La destination est présentée à partir de son début suivant les étapes suivantes :

1. Sauvegarde du contexte courant dans la pile du contexte de présentation.
2. Envoie d'une commande à l'ordonnanceur pour arrêter la présentation.
3. Jouer la destination suivant sa nature :
  - si la destination est un média, celui-ci est joué,
  - si la destination est une scène, sa structure d'exécution correspondante est chargée. La scène est alors jouée.
4. À l'arrivée de la fin du média ou de la scène, le contexte sauvegardé est restauré et la présentation est poursuivie à partir du point d'interruption.

#### **3.4.4. Manipulation d'un média**

Elle correspond aux opérations d'arrêt, d'avance ou de pause, que peut effectuer l'utilisateur à l'intérieur d'un média. Ces opérations induisent des situations d'indéterminisme, leur gestion est présentée dans la section 4 ci-dessous.

## **4. Gestion de l'indéterminisme**

L'indéterminisme d'une présentation correspond à une situation où des enchainements de média sont définis en fonction de données connues uniquement au moment de la présentation. Dans MediaStudio, cette situation est liée à deux types d'interactions de l'utilisateur.

Le premier est lorsqu'il arrête complètement (stop) un média, dans ce cas, la présentation peut arriver à un point de synchronisation impliquant ce média (sa fin ou un instant précis de sa présentation entrainera le démarrage ou l'arrêt d'un autre média) et par conséquent, la synchronisation définie ne peut plus être respectée.

Le deuxième type, est lorsqu'il effectue des opérations de manipulation intra-média : avance, retour ou pause.

#### 4.1. À l'arrêt complet d'un média

Dans ce cas nous avons un indéterminisme en avance. Le média en question, autre que l'audio, est ignoré et son espace d'affichage reste vide. Il n'y aura aucune incidence sur la présentation des médias avec lesquels il est synchronisé, ils seront présentés normalement à leurs instants de débuts et arrêtés à leurs instants de fins. Toutefois, et afin d'éviter l'attente, lorsque le média arrêté ne possède aucune intersection temporelle avec aucun autre média, une translation temporelle vers le début du prochain média sera effectuée.

#### 4.2. Lors de manipulation d'un média

Dans ce cas, la présentation du document est figée et ne sera actif que le média objet de l'opération. Elle sera reprise sur action de l'utilisateur et ce à partir de son instant d'arrêt. Cette manipulation est gérée par l'algorithme suivant :

```
If (MediaSelectionne.interaction == navigationIntraMedia)
{
  MediaSelectionne.SauvegarderOffset (EtatAvancement);
  For each media in ListeMediasActifs
  {
    If(media <> mediaSelectionne)
    {
      Media.Pause();
      Media.horloge.Pause();
    }
    HorlogeGlobale.Pause() ;
  }
}
If (MediaSelectionne.interaction == ArrêtMedia)
{
  MediaSelectionné.Arreter() ;
  Nœud = MediaSelectionne.NoëudDestination();
  GrapheExecution.Nœud.Compteur = GrapheExecution. Nœud.Compteur -1;
}
```

La procédure *NœudDestination()* permet à un media de trouver son nœud de destination dans le graphe d'exécution. Après une navigation intra-média, la présentation du document ne sera reprise que sur intervention de l'utilisateur. À ce point le code suivant est exécuté :

```

MediaSelectionné..etatAvancement=MediaSelectionne.RestaurerOffset();
Media.initialiseHorloge(MediaSelectionne..etatAvancement);
Media.horloge.Demarrer();
For each media in listeMediasActifs
{
    Media.Reprendre();
    Media.horloge.Reprendre();
}
If( mediaSelectionne.FinAtteinte)
{
    Nœud = MediaSelectionne.NoëudDestination();
    GrapheExecution.Nœud.Compteur = GrapheExecution. Nœud.Compteur
+1;
}
HorlogeGlobale.Reprendre();

```

Les procédures *SauvegarderOffset()* et *RestaurerOffset()* permettent respectivement à un média de sauvegarder son état d'avancement (le temps écoulé de sa durée) et de le restaurer. La procédure *InitialiserHorloge()* permet à un média de configurer son horloge à démarrer à partir de n'importe quel instant de sa présentation.

```

Procedure InitialiserHorloge(int instant)
{
    TempsRestant = Media.Fin - instant;
    Horloge.periode = tempsRestant;
}

```

## 5. Conclusion

Dans ce chapitre, nous avons présenté l'architecture, le fonctionnement ainsi que la mise en œuvre du système de présentation de MediaStudio. Le rôle de ce système est d'effectuer tous les traitements de présentation requis pour jouer et interagir avec une présentation. Nous avons montré comment le graphe d'exécution était exploité pour l'ordonnancement de la présentation, pour la gestion de l'indéterminisme à travers le formatage dynamique, ainsi que pour la gestion de la navigation et de la manipulation des médias.

# Chapitre 9

## Conclusion

### 1. Rappel des objectifs

Mon travail de thèse porte sur la spécification et la validation des relations temporelles et spatiales ainsi que la gestion du recouvrement spatial dans un système d'édition et de présentation de documents multimédia qui intègre des éléments de type texte, image, audio, vidéo ainsi que les interactions de l'utilisateur.

La construction d'un document multimédia est soumise à des contraintes multiples provenant de la nature diverse des médias, de leur combinaison, et de leur intégration homogène au sein des dimensions d'un document. À ces contraintes s'ajoutent celles de l'organisation d'une application qui est impliquée dans plusieurs couches d'un système.

La plupart des systèmes actuels proposent des méthodes d'édition fondées sur une approche impérative (langages) ou sur la datation explicite (les timelines). Dans ces systèmes, la construction d'un document est un processus long, source de nombreuses erreurs et difficile à maîtriser pour un auteur non spécialisé. Les documents produits sont souvent complexes, difficiles à maintenir et peu compatibles entre eux.

Par ailleurs, il a été constaté que la quasi-totalité des systèmes et des standards étudiés se sont essentiellement focalisés sur la dimension temporelle et ce au détriment des autres dimensions. Or un système d'édition et de présentation devrait être d'emblée considéré à travers ses quatre dimensions afin de non seulement mutualiser sa conception et sa réalisation mais également d'appréhender, notamment, le problème du recouvrement spatial qui, comme montré, peut altérer considérablement la qualité de service d'une présentation alors que celle-ci est l'objectif recherché à travers ces travaux.

Enfin, les documents produits, dans les systèmes et les standards étudiés, ne peuvent pas répondre au besoin, qui se fait de plus en plus ressentir, de l'adaptation du contenu selon un profil particulier.

L'étude des systèmes disponibles montre, en premier lieu, la nécessité de redonner le contrôle aux auteurs et de les assister par la conception d'outils qui allient simplicité de création d'un document/parcours de lecture et puissance d'expressivité, détectent les incohérences, gèrent le recouvrement spatial et permettent d'obtenir une restitution conforme à celle que l'auteur souhaite. De plus, il faut reconsidérer le statut du lecteur en lui donnant la possibilité d'interagir avec le système pour définir ses propres parcours de lecture.

## 2. Bilan scientifique

Dans cette perspective, nous avons analysé la nature des documents multimédia et nous avons passé en revue les principales approches de l'édition multimédia à travers les méthodes de construction de scénario et les standards proposés. Nous avons étudié les langages de spécification de scénarios multimédia, ainsi que les techniques d'analyse et de synthèse les plus utilisés.

Notre démarche a été arrêtée par rapport aux objectifs de proposer un modèle de document intégrant les quatre dimensions d'un document multimédia et permettant la définition de parcours de lecture selon le profil. De proposer des modèles à base de relations pour la spécification des informations temporelles et spatiales. D'adapter une approche pour la vérification de la cohérence de la spécification et la génération de solution applicable aux deux modèles. De proposer une gestion automatisée du recouvrement spatial se basant sur la même approche de vérification de la cohérence, moyennant quelques adaptations. Et enfin d'offrir des mécanismes de navigation pour l'exploitation de la présentation, notamment sa dimension temporelle.

Le premier résultat est un modèle de document gérant les quatre dimensions, duquel l'auteur comme le lecteur peuvent définir des parcours de lecture selon un profil particulier sans qu'ils ne soient conçus comme autant de documents car ils partagent les mêmes données avec le document primaire. L'autre apport du modèle est la définition de médias dérivés à partir de médias physiques permettant ainsi l'utilisation de parties de médias et évitant par la même leur création.

Le deuxième résultat est la proposition de deux modèles (temporel et spatial) pour la spécification des relations comportant respectivement l'ensemble des relations temporelles et spatiales jugées utiles pour le multimédia. Ils se caractérisent par une simplicité d'utilisation et une puissance d'expressivité relativement importante comparée à celles des modèles à base de relations rencontrés dans la littérature. Grâce à l'introduction des concepts du délai et de la distance flexibles, les modèles permettent la spécification d'expressions qu'aucun modèle actuel ne peut offrir.

Le modèle temporel permet, non seulement, l'expression de l'indéterminisme, mais offre également, dans certains cas, des informations utiles pour sa gestion lors des phases d'analyse et de présentation. Le choix du voisinage d'un média conjugué à la distance flexible permet des emplacements d'une grande précision et permet même l'emplacement d'un média sur deux cadrans adjacents. Sans lui définir des relations spécifiques, l'utilisation de la distance négative permet d'exprimer le recouvrement désiré. Enfin, avec le délai et de la distance flexibles, les spécifications de l'auteur sont moins contraignantes, ce qui donne plus de chance à un média d'être *placé* temporellement ou spatialement et d'éviter ainsi à l'auteur des re-spécifications.

La procédure de gestion du recouvrement spatiale constitue notre troisième résultat. L'auteur est d'une part complètement déchargé de la contrainte qui l'obligeait à faire des allers/retours entre les phases d'édition et de présentation pour vérifier la présence éventuelle de recouvrements non désirés et d'y pallier le cas échéant. D'autre part, elle constitue un garant d'une qualité de service acceptable de la présentation.

La gestion de la cohérence et la recherche de solutions des systèmes de contraintes temporelle, spatial et du recouvrement non désiré par le même résolveur (cassowary), constitue un autre résultat qui nous a permis d'unifier et d'optimiser leurs gestions au niveau de MediaStudio.

L'autre résultat est la définition d'un schéma XML, normalisant toutes les informations contenues dans un document multimédia, servant à convertir le document/parcours de lecture globalement cohérent (temporellement, spatialement et sans recouvrements non désirés) en document XML lors de son exportation et à valider un document XML lors de son importation pour sa présentation. Une visionneuse (application) accompagne le document XML lors de son exportation.

Le développement de MediaStudio constitue notre dernier résultat. Doté d'un ensemble de fonctions qui gèrent la construction d'un document multimédia interactif depuis son édition jusqu'à sa présentation où les limites entre ces deux phases ont été sensiblement réduites.

### **3. Bilan et évaluation de la réalisation**

Cette étude n'offrait un réel intérêt que si elle était concrétisée par la réalisation d'un prototype validant les idées qui en émergeaient. La réalisation de MediaStudio a comporté une part importante de développement pour le mettre en œuvre. Elle a contribué au raffinement des modèles et des approches adoptées et d'une manière globale à toute la réflexion.

L'originalité du système MediaStudio se situe à plusieurs niveaux : d'abord il constitue une réponse quasi-globale à tous les problèmes de l'édition et de la présentation d'un document multimédia interactif en étant d'emblée conçu dans cette perspective. Son architecture modulaire a été définie de manière à faciliter l'expérimentation de divers algorithmes intervenant depuis la construction jusqu'à la présentation d'un document multimédia. Son utilisation n'exige aucun pré-requis de formation ou de connaissances particulières. Aucune limite dans la construction d'un document n'est imposée hormis l'option de ne définir qu'une seule relation entre deux média, option que nous avons jugée utile pour simplifier la construction d'un document et par conséquent son analyse et sa maintenance et ce sans limitation réelle du pouvoir d'expression de l'auteur ; et la non prise en compte de l'indéterminisme du aux ressources externes (accès distant aux média) car nous avons conçu le système dans une logique de données locales.

Le prototype fonctionne de manière satisfaisante, avec des performances très acceptables même devant des scénarios complexes (grand nombre de média, des spécifications cycliques, etc.).

### **4. Perspectives**

Cette contribution apporte des réponses aux problèmes qui se posent à chaque niveau de la construction d'un document multimédia (spécification, analyse, synthèse et présentation). Elle constitue une base qu'il faudra encore améliorer et compléter. En plus de la consolidation de quelques propositions faites dans cette thèse, le travail qui reste à faire peut s'orienter sur deux directions : la recherche et le développement.

## **4.1 Aspect recherche**

### **Définition de modèles prédéfinis de documents**

Une façon d'assister l'auteur dans la conception de ses documents, notamment ceux qui sont d'une certaine complexité, est la prédéfinition de modèles de documents en fonctions d'un domaine ou d'une thématique particulière (cours interactifs, présentation médicale, visite virtuelle de patrimoine, guide touristique, etc.) qui leurs serviront comme bases de conception.

### **Adaptation dynamique du document**

Une autre forme d'adaptation qui peut être traitée par les contraintes est l'adaptation dynamique du document au cours de la présentation. Cette adaptation se fait par rapport aux conditions du réseau, de la charge du système, de la taille de l'écran, des préférences et contraintes du lecteur, etc. À partir d'un document formaté partiellement statiquement, le système de présentation va appliquer une (ou plusieurs) phases de formatage dynamique pour adapter la présentation aux nouvelles conditions.

### **Documents Multi-langages**

Une des applications intéressantes qui tire partie d'un formatage souple est de pouvoir réaliser des documents multilingues, formatés en fonction de la langue. Par exemple, un média audio n'aura pas la même durée en arabe ou en français. De la même façon, un texte arabe n'aura pas la même longueur que sa traduction française. De ce fait, avoir un document contenant de la flexibilité et un outil permettant de formater ce document est un apport intéressant pour l'auteur et le lecteur, qui laisse ainsi le soin au formateur de calculer au mieux le placement des médias, en conservant si possible la durée intrinsèque de ces médias. L'auteur ne spécifie ainsi qu'un document. Le système le formatera alors de façon optimale pour chacune des langues.

### **Le recouvrement temporel non désiré des médias**

En plus de la gestion du recouvrement spatial non désiré, il serait intéressant d'étudier dans quelles mesures le recouvrement temporel des média (une vidéo qui se joue en même temps qu'un commentaire sonore ou une musique, etc.) peut avoir des conséquences sur la qualité de service d'une présentation et de proposer des solutions.

## **4.2 Aspect développement**

### **Edition des documents en XML**

Disposant d'un schéma XML pour les documents multimédia, il serait intéressant d'envisager une édition en XML des documents pour les auteurs initiés. L'intérêt est de leur simplifier la création et de gérer plus rapidement les erreurs de conception qui seront, en grande partie, détectées et signalées par un parseur XML intégré.

### **Application du concept Software as a Services (Saas)**

La tendance actuelle dans le développement d'applications essaye de suivre le concept du (Saas) pour ainsi rendre l'application partageable via le web. L'intérêt avancé de cette approche est qu'elle induirait des coûts moins importants que les applications traditionnelles (TCO, total cost of ownership ou ASP, Application service Provider).

# Bibliographie

- [1] Allen.J. F. "Maintaining Knowledge about Temporal Intervals", *Comm. ACM*, 26(11): 832–843, 11 1983.
- [2] Badros G. J., Nichols J. et Borning A., "Scwm--An Intelligent Constraint-Enabled Window Manager", *AAAI Spring Symposium on Smart Graphics*, mars 2000.
- [3] Badros G.J., "UW Cassowary Constraint Solving Toolkit" University of Washington February 2003 [www.cs.washington.edu/research/constraints/cassowary/](http://www.cs.washington.edu/research/constraints/cassowary/)
- [4] Badros et Borning. "The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation", Technical Report UW-CSE-98-06-04, University of Washington, June 1998
- [5] 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.
- [6] Baykan A., Fox M., "Spatial layout by disjunctive constraint satisfaction", *AIEDAM* 11, 245–262, January 1997.
- [7] 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.
- [8] Bertino E. and Ferrari E., "Temporal synchronization models for multimedia data", *TKDE*, 10(4):612–631, 1998.
- [9] Berkom, "GLUE Project", <http://www.fokus.gmd.de/ovma/berglass/entry.html>, 1998.
- [10] Bitter J. R. and Reingold E. M. "Backtrack programming techniques", *Communication of the ACM*, 18(11):651-656, 1975
- [11] Bitzer H.W., Gran C. et Hofrichter K., "MAJA : MHEG Applications Using JAVA Applets", In *proceedings of Real Time Multimedia and the World Wide Web, W3C Workshop (RTMW'96)*, Sophia-Antipolis, France, octobre 1996.
- [12] Blakowski G., Steinmetz R., "A Media Synchronisation Survey: Reference Model, Specification, and Case Studies", *IEEE Journal Of Selected Areas In Communications*, vol. 14, num. 1, pp. 5-34, janvier 1996.
- [13] Boll W. Klas and U. Westermann, "A Comparison of Multimedia Document Models Concerning Advanced Requirements," Technical Report-Ulmer Informatik-Berichte Nr. 99-01, Univ. Ulm, Germany, 1999.
- [14] Borning A., Richard Anderson, and Bjorn Freeman-Benson," Indigo : A local propagation algorithm for inequality constraint", in *proceeding of the 1996 ACM symposium on user interface software and technology*, pages 129-136. 1996.
- [15] Borning A., K. Marriott, P. Stuckey, and Y. Xiao., "Solving linear arithmetic constraints for user interface applications: Algorithm details", Tech report 97-06-01, Dept. Computer Science & Engr, Univ of Washington, Seattle, WA, July 1997.
- [16] Borning A. et Freeman-Benson B., "Ultraviolet: A Constraint Satisfaction Algorithm for Interactive Graphics", *Constraints, Special Issue on Constraints, Graphics, and Visualization*, Vol. 3 No. 1, pp. 9-32, avril 1998.

- [17] Buchanam (C.), Zellweger (P.T.), "Specifying Temporal Behavior in Hypermedia Documents ", Proc. of the ACM Conf. on Hypertext, pp. 262-271, décembre 1993.
- [18] Buchanan M. Cecilia and Polle T. Zellweger. "Automatic Temporal Layout Mechanisms", In 1st ACM Intl. Conference on Multimedia, pages 341 – 350, 8 1993.
- [19] Bulterman. C. A., "Embedding Video in Hypermedia Documents: Supporting Integration and Adaptive Control", ACM Transactions on Information Systems, pp. 1-30, octobre 1995.
- [20] Carcone L., Jourdan M., Roisin C. "Présentation de documents multimédia basée sur les contraintes", Workshop on Electronic Page Models - LAMPE, Lausanne (Suisse), vol., 22-23 Septembre 1997.
- [21] Carcone L., "Formatage spatial dans un environnement d'édition/présentation de documents multimédia", Mémoire CNAM, Conservatoire National des Arts et Métiers, Grenoble, décembre 1997.
- [22] Crilly, "The geometer sketchPad 6.0", Math & Stats, Vol. 6, Num. 2, pp 34.36, 1995.
- [23] Dantzig G. B. "Programming of Interdependent Activities. II. Mathematical Model", *Econometrica* 17, 200-211, 1949.
- [24] Dechter R. and Frost, D., "Backjump-based Backtracking for Constraint Satisfaction Problems", in *Artificial Intelligence*, Volume 136, Number 2, April 2002, pp. 147-188(42)
- [25] Deroose S. J. et Durand D. G., "Making Hypermedia Work: A User's Guide to HyTime", Kluwer Academic Publishers, mars 1994.
- [26] Drapeau George D. "Synchronization in the MAEStro Multimedia Authoring Environment", In 1st Intl. ACM Conference on Multimedia, pages 331 – 340, 8 1993.
- [27] Duke D.J., Duce D.A., Herman I. et Faconti G., "Specifying the PREMO Synchronization Objects", num. ERCIM-01/97-R048, ERCIM Technical Report, 1997.
- [28] Duverger A., "Recherche de solution au problème de chevauchement de labels", Rapport d'ingénieur Cnam, IIE, Mai 2000.
- [29] Egenhofer M., "Reasoning about Binary Topological Relations", In the Proceedings of the Second Symposium on the Design and Implementation of Large Spatial Databases. Springer Verlag LNCS. 1991.
- [30] Egenhofer. M.J. "Spatial SQL: a query and presentation language", *IEEE Transaction on knowledge and data engineering*. 6(1), 1994
- [31] Erfle R., "Specification of temporal constraints in multimedia documents using HyTime", *Electronic Publishing*, vol. Vol. 6, num. 4, pp. 397-411, December 1993.
- [32] Frank, A. U., "Qualitative spatial reasoning: cardinal directions as an exemple", *International journal of geographical information science*, 10(3): 269-290, 1996.
- [33] Freeman-Benson Bjorn N., Maloney John, and Borning Alan. An incremental constraint solver. *Communications of the ACM*, 33(1):54–63, January 1990.
- [34] Goldfarb C. F., "The SGML Handbook", Oxford University Press, Oxford, 1990.

- [35] Gregory M. Oster, Anthony J. Kusalik, ``An Overview of ICOLA and PDI: Incremental, Constraint-Based Graphics for Visualization'', *Proceedings of International Workshop on Constraints for Graphics and Visualization*, num. CGV'95, pp. 119-134, September 1995.
- [36] Hafid A., Bochmann G. et Dssouli R., ``Distributed Multimedia Application and Quality of Service: A Review'', *Electronic Journal on Networks and Distributed Processing*, vol. No.6, , pp. 1-50, février 1998.
- [37] Hardman L., ``Modelling and Authoring Hypermedia Documents'', Ph.D. Thesis, University of Amsterdam (Universiteit van Amsterdam), mars 1998.
- [38] Herman I., Reynolds G.J. et Vanl Loo J., ``PREMO: an emerging standard for multimedia presentation'', num. CS-R9554, Computer Science/Department of Interactive Systems, CWI, Netherlands, 1995.
- [39] Hoepner. P., ``Synchronisation der Praesentation von Multimediaobjekten-Modell und Beispiele'', In J.Encarnacao, editor, *Informatik-Fachberichte 293, Telekommunikation und multimediale Anwendungen der Informatik*, pages 455–464. GI, Springer-Verlag, 10 1991.
- [40] Hudson Scott E. and Shamim P.Mohamed., ``Interactive specification of flexible user interface displays'', *ACM Transactions on Information Systems*, 8(3):269–288, July 1990.
- [41] ISO/IEC JTC1/SC18/WG8 N1920, ``Information Technology: Hypermedia/Time-based Structuring Language (HyTime)'', Second edition, ISO/IEC, août 1997. <http://www.ornl.gov/sgml/wg8/docs/n1920/html/n1920.html>.
- [42] ICONAUTHOR 6.0, ``User's guide'', Aim Tech, 1997: <http://www.aimtech.com/products/iconauthor>.
- [43] ISO/IEC JTC1/SC24, ``Presentation Environments for Multimedia Objects [PREMO], Part 1: Fundamentals of PREMO'', num. 14478-1:199x(E), ISO/IEC, septembre 1996.
- [44] ISO/IEC JTC1/SC24, ``Presentation Environments for Multimedia Objects (PREMO); Part 2: Foundation Component'', num. 14478-2:199x(E), ISO/IEC, septembre 1996.
- [45] ISO/IEC JTC1/SC24, ``Presentation Environments for Multimedia Objects (PREMO); Part 3: Multimedia Systems Services'', num. 14478-3:199x (E), ISO/IEC, septembre 1996.
- [46] ISO/IEC JTC1/SC24, ``Presentation Environments for Multimedia Objects (PREMO); Part 4: Modelling, Rendering, and Interaction Component'', num. 14478-4:199x(E), ISO/IEC, septembre 1996.
- [47] Jourdan M., N. Layada et L. Sabry-Ismaïl, ``Madeus: An Authoring Environment for Interactive Multimedia Documents'', *IEEE International Conference on Multimedia Computing and Systems*, Ottawa (Canada) juin 1997
- [48] Jourdan M., N. Layaïda, C. Roisin, ``A survey on authoring techniques for temporal scenarios of multimedia documents'', vol. *Handbook of Internet and Multimedia Systems and Applications*, part 1: Tools and Standards, ed. Borko Furht, CRC Press. April 1998. pp. 469-490.
- [49] Jourdan M., Roisin C. et Tardif L., ``Multiviews Interfaces for Multimedia Authoring Environments'', *International Conference on Multimedia Modeling (MMM'98)*, pp. 72-79, Lausanne, Switzerland, octobre 1998.

- [50] Jourdan M., Roisin C., Tardif L., “Édition et Visualisation Interactive de Documents Multimedia ”, Electronic Publishing'98, St Malo, France, avril 98.
- [51] Jourdan M., Layaïda N., Roisin C., “Le temps dans les documents”, Techniques de l'ingénieur, 1999.
- [52] Kim M. Y., Song J., “Multimedia Documents with Elastic Time”, Proc. of the 3rd ACM Conf. on Multimedia, pp. 143-154, San Francisco, novembre 1995.
- [53] Kretz Francis and Francoise Calaitis., "Standardizing Hypermedia Information Objects", IEEE Communications Magazine, 5 1992.
- [54] Layaïda N., J. R. Von Ossenbruggen. “SMIL 2.0 Language Profile”, W3C Recommendation, World Wide Web Consortium. 7 August 2001. <http://www.w3.org/TR/smil20/smil20-profile.html>
- [55] Layaïda N., L. Sabry-Ismail, C. Roisin, “Dealing with uncertain durations in synchronized multimedia presentations”, Multimedia Tools and Applications Journal, Kluwer Academic Publishers, vol. 18, num. 3, December 2002. pp. 213-231
- [56] Layaïda N. et Sabry-Ismail L., “Madeus: un modèle de document multimédia structuré”, TSI, 15:9, pp. 1227-1257, septembre 1996.
- [57] Layaida N., “Madeus: un système d'édition et de présentation de documents structurés multimédia”, Thèse PhD, Université Joseph Fourier, Grenoble-France, Juin 1997.
- [58] Legall D., “MPEG : A Video Compression Standard for Multimedia Applications”, Communications of the ACM, vol. Vol. 34, num. No. 4, pp. 45-68, avril 1991.
- [59] Little T. D. C. and A. Ghafoor., “Synchronisation and Storage Models for Multimedia Objects”, IEEE Journal on Selected Areas in Communications, 8(3):413–427, 3 1990.
- [60] Loay Sabry-Ismail, “Schéma d'exécution pour les documents multimédia distribués”, Thèse de Doctorat l'Université Joseph Fourier Grenoble, France 1999
- [61] Macromedia Director, "User's Guide", Macromedia Inc., 1995.
- [62] Macromedia, “Flash et Director”, <http://www.macromedia.com>, 1998.
- [63] Maloney J.H., Borning A. et Freeman-Benson B.N., "Constraint technology for user-interface construction in ThingLab II", SIGPLAN Notices vol. 24, issue 10, pp. 381-388, 1989.
- [64] Maredj A., Alimazighi Z., Tonkin N. et Sadallah M., “La Gestion du Recouvrement Spatial dans les Documents Multimédia”, 5ieme Colloque International TELECOM'2005 & 4ieme Journée Franco-Maghrébines des Micro-ondes et leurs Applications, 2005.
- [65] Maredj A., Z. Alimazighi et A. Hamour, “Contraintes disjonctives pour la gestion du recouvrement spatial dans les documents multimédia”, in Information-Interaction- intelligence, A Journal in information Engineering Sciences, France, Volume 6, N°1, pages 9-29 Décembre 2006.
- [66] Maredj A., Alimazighi Z., Tonkin N. et Sadallah M., “Adaptation de l'approche de Mariott pour les Documents Multimédia”, 4ime Conférence internationale JTEA2006.
- [67] Maredj A., Hammour A. et Ider D., “Une approche basée sur le simplexe pour le formatage spatial de documents multimédia interactifs”, Conférence Internationale: SETIT04 2004.

- [68] Maredj A., Alimazighi Z., Tonkin N. et Sadallah M., “Extension of the Wahl-Rothermel temporal model for the multimedia documents”, The first International Conference on information and communication Technology and accessibility, ICTA 2007.
- [69] Maredj A., Alimazighi Z. et A. Hamour, “Gestion du recouvrement spatial dans les documents multimédia : Approche et Evaluation”, in TSI : Technique et Science Informatique, Hermes Lavoisier, Volume 27-N°1-2, pages 29-50-2008.
- [70] Maredj A., Alimazighi Z., Tonkin N. et Sadallah M., “Un Modèle logique de documents pour les interactions multimédia”, Conférence Internationale JTEA 2008.
- [71] Maredj A., Alimazighi Z., Tonkin N. et Sadallah M., “A Flexible Distance for the spatial placement in a Multimedia Document”, The International Conference on Information & Communication Technologies: from Theory to Applications – ICTTA 2008.
- [72] 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.
- [73] Marriott Kim and Stuckey Peter J. “Programming with Constraints: An Introduction”, MIT Press, Cambridge, Massachusetts, 1998.
- [74] Meyer-Boudnik T. et Effelsberg W., “MHEG Explained”, IEEE Multimedia Magazine, vol. 2, num. 1, pp. 26-38, Spring 1995.
- [75] MHEG. ISO/IEC/WD. Information Technology – Coded, “Representation of Multimedia and Hypermedia Information Objects”, Working Draft 5, ISO/IEC, 3 1992.
- [76] Miguel I. and Shen Q. “Solution Techniques for Constraint Satisfaction Problems: Advanced Approaches”, Artificial Intelligence Review, 15(4):239-293, 2001
- [77] Minglu L., YONGQIANQ S., HUANYE S., “Temporal Representation for Multimedia Systems”, Time 96 International Conference, Florida, USA, 19-20 mai 1996.
- [78] Myers Brad A., Giuse Dario, Dannenberg Roger B., Vander Zanden Brad, Kosbie David S., Marchal Philippe, Ed Pervin, Mickish Andrew, and Kolojejchick John A., “The Garnet toolkit reference manuals: Support for highly-interactive graphical user interfaces in Lisp”, Technical Report CMU-CS-90-117, Computer Science Dept, Carnegie Mellon University, March 1990.
- [79] Myers Brad A, McDaniel Richard G, Miller Robert C, Ferreny Alan S, Faulring Andrew, Kyle Bruce D, Mickish Andrew, Klimovitski Alex, and Doane Patrick., “The Amulet environment: New models for effective user interface software development”, IEEE Transactions on Software Engineering, 23(6):347–365, June 1997.
- [80] Papadias D., Y. Theodoridis, T. Sellis et M. Egenhofer., “Topological relations in the world of minimum bounding rectangles : A study with R-trees”, Dans Proc. Of the ACM conference of the modelling of data (SIGMOD), Mai 1995.
- [81] Papadias D. et Y. Theodoridis., “Spatial relations, minimum bounding rectangles and spatial data structures”, International journal of geographic information science, 11(2), 1997.

- [82] Papadias D., Y. Theodoridis, E. Stefanakis., "Multi-dimensional range query processing with spatial relations", *Geographical systems*, 4(4), 1997.
- [83] Perez-Luque M. J., Little T. D. C., "A Temporal Reference Framework for Multimedia Synchronisation", *IEEE Journal on Selected Areas in Communications*, vol. 14, num. 1, pp. 36-51, janvier 1996.
- [84] Michael M. Richter., "Prinzipien der Kuenstlichen Intelligenz", B.G Teubner Stuttgart, 1989.
- [85] Rossum van G., J. Jansen, K. Mullender et D. Bulterman, "CMIFed : a presentation Environment for Portable Hypermedia Documents", *Proc. of the ACM Multimedia Conference, California*, 1993.
- [86] Safar M., C. Shahabi. "2D topological and direction relations in the world of minimum bounding circles", in *Processing of IDEAS International database engineering ans application symposium*, Aout, 1999.
- [87] Sannella M., Maloney J., Freeman-Benson J. and Borning A., "Multi-way versus One-way constraints in user interfaces: Experience with the DelatBlue algorithm", *Software-Pratice and Experience*, Vol. 32(5), pp529-566, May 1993.
- [88] Sannella M., "The Skyblue Constraint Solver and Its Applications", Vijay Saraswat and Pascal van Hentenryck, editors, *Proceedings of the 1993 Workshop on Principles and Practice of Constraint Programming*, pp. 385-406, MIT Press, 1995.
- [89] Schloss G. et Wynblatt M., "Building Temporal Structures in a Layered Multimedia Data Model", in *proceedings of ACM Multimedia 94*, pp. 271-278, acm press, California, USA, octobre 1994.
- [90] Senac P., "Contribution à la modélisation des systèmes multimédias et hypermédias", Thèse de doctorat en informatique, Université Paul Sabatier de Toulouse, 12 juin 1996.
- [91] Senac P., Diaz M., Leger A. et de Saqui-Sannes P., "Modeling Logical and Temporal Synchronization in Hypermedia Systems", *IEEE Journal of Selected Areas on Communications*, vol. 14, num. 1, pp. 84-103, 1996.
- [92] SMDL, "Standard Music Description Language", <http://www.sub.su.se/henrik/tomasf/smdl.htm>, 1998.
- [93] Smil 2.0 "Players", *Proceedings, IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 2, 6-9 July 2003, pp. 761-764.
- [94] Smil 2.1, W3C Recommendation, "Synchronized Multimedia Integration Language (SMIL) 2.1 Specification", <http://www.w3.org/TR/SMIL2/>, Décembre 2005.
- [95] Smil 1.0 W3C Recommendation, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", <http://www.w3.org/TR/REC/smil>, 15 June 1998.
- [96] Song J., Doganata Y., Kim M. et Tantawi A., "Modeling Timed User-Interactions in Multimedia Documents", *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, USA, novembre 1996.
- [97] Steinmetz R., "Analyzing the Multimedia Operating System", *IEEE Multimedia Magazine*, vol. Vol. 2 , num. No. 1, pp. 68-84, 1995.
- [98] Steinmetz R., "Human perception of jitter and media synchronization", *IEEE Journal on Selected Areas in Communications*, vol. 14, num. 1, pp. 61-72, janvier 1996.

- [99] Sutherland Ivan. Sketchpad: "A Man-Machine Graphical Communication System", PhD thesis, Department of Electrical Engineering, MIT, January 1963.
- [100] Tardif L., "Kaomi : Réalisation d'une boîte à outils pour la construction d'environnements d'édition de documents multimédias", Thèse de doctorat, Institut National Polytechnique de Grenoble, décembre 2000.
- [101] Vanbeek P., "Reasoning about qualitative temporal information", Proceedings of AAAI, vol. Boston, MA, , pp. 728-734, 1990
- [102] Van Rossum G., Jansen J., Mullender K. And Bulterman D., "CMIFed : a presentation Environment for Portable Hypermedia Documents", Proc. of the ACM Multimedia Conf., California, 1993.
- [103] Verfaillie G. et Schiex T., "Maintenance de solution dans les problèmes dynamiques de satisfaction de contraintes: bilan de quelques approches", *Revue d'intelligence artificielle*, volume 9, n°3, pp. 269-309, 1995.
- [104] Vilain M. and H.A. Kautz., "Constraint propagation algorithms for temporal reasoning", in AAAI-86 Philadelphia, PA, pages 132 –144, 1986.
- [105] Vizirgiannis M., Y. Theodoridis et T. Sellis., "Spatio-temporal composition and indexing for large multimedia applications", ACM Springer-Verlag Multimedia Journal, 6(4), 1998.
- [106] W3C, "Synchronized Multimedia Integration Language (SMIL) 1.0", P. Hoschka. W3C Recommendation 15 juin 1998, disponible à : <http://www.w3.org/TR/REC-smil>.
- [107] Wahl T. and Kurt Rothermel., "Representing Time in Multimedia Systems", Technical Report 12, Universitaet Stuttgart, 1993.
- [108] Wahl T., Rothermel K., "Representing Time in Multimedia Systems", IEEE Proceedings of the International Conference on Multimedia Computing and Systems, pp. 538-543, 14-19 mai 1994.
- [109] Wallace G., "The JPEG Still Picture Compression Standard", Communications of the ACM, vol. Vol. 34, num. No. 4, pp. 30-44, avril 1991.
- [110] Wirag S., "Modeling of Adaptable Multimedia Documents", Interactive Distributed Multimedia Systems and Telecommunication Services; 4th International Workshop, IDMS'97, pp. 420-429, Darmstadt, septembre 1997.
- [111] Xinguang Chen and Peter van Beek., "Conflict-Directed Backjumping Revisited", Journal of Artificial Intelligence Research, 14:53-81, 2001
- [112] Yang. C. C., "Design of the Data-Retrieving Engine for Distributed Multimedia Presentations", Proceedings, IEEE International Conference on Multimedia and Expo (ICME), 2001.
- [113] W. Tien, and Y. C. Wang, "Modeling of the Non-deterministic Synchronization Behaviors in SMIL2.0 Documents", Proceedings, IEEE International Conference on Multimedia and Expo (ICME), Vol. 3, 6-9 July 2003.
- [114] Yang. C. C, C. K. Chu, and Y. C. Wang, "Dividable Dynamic Timeline-based Authoring for SMIL2.0 presentations", in Proceedings of IEEE International Conference on Multimedia and Expo (ICME), 2004.

- [115] Zechinilli J. L. Martini. "Construction et manipulation de présentations spatio-temporelles multimédias à partir de serveurs de médias repartis : Application aux données sur le web". Thèse de doctorat. Université Joseph Fourier 2001.

# ANNEXES

## Annexe 1

Nous donnons dans le tableau ci-dessous les 169 relations directionnelles possibles entre deux médias.

	R <sub>i_1</sub>	R <sub>i_2</sub>	R <sub>i_3</sub>	R <sub>i_4</sub>	R <sub>i_5</sub>	R <sub>i_6</sub>	R <sub>i_7</sub>	R <sub>i_8</sub>	R <sub>i_9</sub>	R <sub>i_10</sub>	R <sub>i_11</sub>	R <sub>i_12</sub>	R <sub>i_13</sub>
R <sub>1_j</sub>													
R <sub>2_j</sub>													
R <sub>3_j</sub>													
R <sub>4_j</sub>													
R <sub>5_j</sub>													
R <sub>6_j</sub>													
R <sub>7_j</sub>													
R <sub>8_j</sub>													
R <sub>9_j</sub>													
R <sub>10_j</sub>													
R <sub>11_j</sub>													
R <sub>12_j</sub>													
R <sub>13_j</sub>													

Tableau des 169 relations spatiales utiles pour le multimédia

Le tableau suivant donne l'équivalence des 169 relations directionnelles dans le modèle que nous avons proposé :

Sur l'axe horizontal	Sur l'axe vertical	Relation équivalente dans le modèle proposé
Before	before	Left_Top ( $d_1, d_2$ ) avec $d_1, d_2 > 0$
	meet	Left_Top ( $d_1, 0$ ) avec $d_1 > 0$
	Overlaps	Left_Top ( $d_1, d_2$ ) avec $d_1 > 0, d_2 < 0$
	Finishes	Left_Top ( $d_1, -A.h$ ) avec $d_1 > 0$
	during ( $B.h > A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	Starts	Left_Top ( $d_1, -B.h$ ) avec $d_1, d_2 > 0$
	equals ( $B.h = A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	starts <sup>-1</sup>	Left_Top ( $d_1, -B.h$ ) avec $d_1$
	during <sup>-1</sup> ( $B.h < A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	finishes <sup>-1</sup>	Left_Top ( $d_1, -A.h$ ) avec $d_1 > 0$
	overlaps <sup>-1</sup>	Left_Bottom ( $d_1, d_2$ ) avec $d_1 > 0$ et $d_2 < 0$
	meet <sup>-1</sup>	Left_Bottom ( $d_1, 0$ ) avec $d_1 > 0$
	before <sup>-1</sup>	Left_Bottom ( $d_1, d_2$ ) avec $d_1 > 0$ et $d_2 > 0$
meet	before	Left_Top ( $0, d_2$ ) avec $d_2 > 0$
	meet	Left_Top ( $0, 0$ )
	Overlaps	Left_Top ( $0, d_2$ ) $d_2 < 0$
	Finishes	Left_Top ( $0, -A.h$ )
	during ( $B.h > A.h$ )	Left ( $0, (A.h - B.h)/2$ )
	Starts	Left_Top ( $0, -B.h$ ) avec $d_1, d_2 > 0$
	equals ( $B.h = A.h$ )	Left ( $0, (A.h - B.h)/2$ )
	starts <sup>-1</sup>	Left_Top ( $0, -B.h$ )
	during <sup>-1</sup> ( $B.h < A.h$ )	Left ( $0, (A.h - B.h)/2$ )
	finishes <sup>-1</sup>	Left_Top ( $0, -A.h$ )
	overlaps <sup>-1</sup>	Left_Bottom ( $0, d_2$ ) avec $d_2 < 0$
	meet <sup>-1</sup>	Left_Bottom ( $0, 0$ )
	before <sup>-1</sup>	Left_Bottom ( $0, d_2$ ) avec $d_2 > 0$
Overlaps	before	Left_Top ( $d_1, d_2$ ) avec $d_1 < 0, d_2 > 0$
	meet	Left_Top ( $d_1, 0$ ) avec $d_1 < 0$
	Overlaps	Left_Top ( $d_1, d_2$ ) avec $d_1 < 0, d_2 < 0$
	Finishes	Left_Top ( $d_1, -A.h$ ) avec $d_1 < 0$
	during ( $B.h > A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 < 0$
	Starts	Left_Top ( $d_1, -B.h$ ) avec $d_1 < 0, d_2 > 0$
	equals ( $B.h = A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 < 0$
	starts <sup>-1</sup>	Left_Top ( $d_1, -B.h$ ) avec $d_1 < 0$
	during <sup>-1</sup> ( $B.h < A.h$ )	Left ( $d_1, (A.h - B.h)/2$ ) avec $d_1 < 0$
	finishes <sup>-1</sup>	Left_Top ( $d_1, -A.h$ ) avec $d_1 < 0$
	overlaps <sup>-1</sup>	Left_Bottom ( $d_1, d_2$ ) avec $d_1 < 0$ et $d_2 < 0$
	meet <sup>-1</sup>	Left_Bottom ( $d_1, 0$ ) avec $d_1 < 0$
	before <sup>-1</sup>	Left_Bottom ( $d_1, d_2$ ) avec $d_1 < 0$ et $d_2 > 0$
Finishes	before	Left_Top ( $-A.l, d_2$ ) avec $d_2 > 0$
	meet	Left_Top ( $-A.l, 0$ )
	Overlaps	Left_Top ( $-A.l, d_2$ ) avec $d_2 < 0$
	Finishes	Left_Top ( $-A.l, -A.h$ )
	during ( $B.h > A.h$ )	Left ( $-A.l, (A.h - B.h)/2$ )
	Starts	Left_Top ( $-A.l, -B.h$ )
	equals ( $B.h = A.h$ )	Left ( $-A.l, (A.h - B.h)/2$ )
	starts <sup>-1</sup>	Left_Top ( $-A.l, -B.h$ )
	during <sup>-1</sup> ( $B.h < A.h$ )	Left ( $-A.l, (A.h - B.h)/2$ )
	finishes <sup>-1</sup>	Left_Top ( $-A.l, -A.h$ )
	overlaps <sup>-1</sup>	Left_Bottom ( $-A.l, d_2$ ) avec $d_2 < 0$
	meet <sup>-1</sup>	Left_Bottom ( $-A.l, 0$ )
	before <sup>-1</sup>	Left_Bottom ( $-A.l, d_2$ ) avec $d_2 > 0$

during ( $B.l > A.l$ )	before	Above( $(A.l - B.l)/2, d_2$ ) avec $d_2 > 0$
	meet	Above( $(A.l - B.l)/2, 0$ )
	Overlaps	Above( $(A.l - B.l)/2, d_2$ ) avec $d_2 < 0$
	Finishes	Above( $(A.l - B.l)/2, -B.h$ )
	During	Centring (1, 1)
	Starts ( $A.h < B.h$ )	Above( $(A.l - B.l)/2, -A.h$ )
	Equals ( $A.h = B.h$ )	Above( $(A.l - B.l)/2, -A.h$ )
	Starts <sup>-1</sup>	Above( $(A.l - B.l)/2, -A.h$ )
	during <sup>-1</sup> ( $A.h < B.h$ )	Centring (1, 1)
	finishes <sup>-1</sup> ( $A.h < B.h$ )	Above( $(A.l - B.l)/2, -B.h$ )
	overlaps <sup>-1</sup>	Below( $(A.l - B.l)/2, d_2$ ) avec $d_2 < 0$
	meet <sup>-1</sup>	Below ( $(A.l - B.l)/2, 0$ )
	before <sup>-1</sup>	Below ( $(A.l - B.l)/2, d_2$ ) avec $d_2 > 0$
	Starts	before
meet		Left_Top (-B.l, 0)
Overlaps		Left_Top (-B.l, $d_2$ ) avec $d_2 < 0$
Finishes		Left_Top (-B.l, -A.h)
during ( $B.h > A.h$ )		Left (-B.l, $(A.h - B.h)/2$ )
Starts		Left_Top (-B.l, -B.h)
equals ( $B.h = A.h$ )		Left (-B.l, $(A.h - B.h)/2$ )
starts <sup>-1</sup>		Left_Top (-B.l, -B.h)
during <sup>-1</sup> ( $B.h < A.h$ )		Left (-B.l, $(A.h - B.h)/2$ )
finishes <sup>-1</sup>		Left_Top (-B.l, -A.h)
overlaps <sup>-1</sup>		Left_Bottom (-B.l, $d_2$ ) avec $d_2 < 0$
meet <sup>-1</sup>		Left_Bottom (-B.l, 0)
before <sup>-1</sup>		Left_Bottom (-B.l, $d_2$ ) avec $d_2 > 0$
equals ( $A.l = B.l$ )		before
	meet	Left_Top (-B.l, 0)
	Overlaps	Left_Top (-B.l, $d_2$ ) avec $d_2 < 0$
	Finishes	Left_Top (-B.l, -A.h)
	during ( $B.h > A.h$ )	Left (-B.l, $(A.h - B.h)/2$ )
	Starts	Left_Top (-B.l, -B.h)
	equals ( $B.h = A.h$ )	Left (-B.l, $(A.h - B.h)/2$ )
	Starts <sup>-1</sup>	Left_Top (-B.l, -B.h)
	during <sup>-1</sup> ( $B.h < A.h$ )	Left (-B.l, $(A.h - B.h)/2$ )
	finishes <sup>-1</sup>	Left_Top (-B.l, -A.h)
	overlaps <sup>-1</sup>	Left_Bottom (-B.l, $d_2$ ) avec $d_2 < 0$
	meet <sup>-1</sup>	Left_Bottom (-B.l, 0)
	before <sup>-1</sup>	Left_Bottom (-B.l, $d_2$ ) avec $d_2 > 0$
	starts <sup>-1</sup>	before
meet		Right_Top (-A.l, 0)
Overlaps		Right_Top (-A.l, $d_2$ ) avec $d_2 < 0$
Finishes		Right_Top (-A.l, -A.h)
during ( $B.h > A.h$ )		Right (-A.l, $(A.h - B.h)/2$ )
Starts		Right_Top (-A.l, -B.h)
equals ( $B.h = A.h$ )		Right (-A.l, $(A.h - B.h)/2$ )
starts <sup>-1</sup>		Right_Top (-A.l, -B.h)
during <sup>-1</sup> ( $B.h < A.h$ )		Right (-A.l, $(A.h - B.h)/2$ )
finishes <sup>-1</sup>		Right_Top (-A.l, -A.h)
overlaps <sup>-1</sup>		Right_Bottom (-A.l, $d_2$ ) avec $d_2 < 0$
meet <sup>-1</sup>		Right_Bottom (-A.l, 0)
before <sup>-1</sup>		Right_Bottom (-A.l, $d_2$ ) avec $d_2 > 0$

during <sup>-1</sup> (B.l < A.l)	before	Right_Top (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0, d <sub>2</sub> > 0
	meet	Right_Top (d <sub>1</sub> , 0) avec d <sub>1</sub> < 0
	Overlaps	Right_Top (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0, d <sub>2</sub> < 0
	Finishes	Right_Top (d <sub>1</sub> , -A.h) avec d <sub>1</sub> < 0
	during (B.h > A.h, B.l < A.l)	Centring (1, 1)
	Starts	Right_Top (d <sub>1</sub> , -B.h) avec d <sub>1</sub> < 0, d <sub>2</sub> > 0
	equals (B.h = A.h, B.l < A.l)	Centring (1, 1)
	starts <sup>-1</sup>	Right_Top (d <sub>1</sub> , -B.h) avec d <sub>1</sub> < 0
	during <sup>-1</sup> (B.h < A.h, B.l < A.l)	Centring (1, 1)
	finishes <sup>-1</sup>	Right_Top (d <sub>1</sub> , -A.h) avec d <sub>1</sub> < 0
	overlaps <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0 et d <sub>2</sub> < 0
	meet <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , 0) avec d <sub>1</sub> < 0
	before <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0 et d <sub>2</sub> > 0
	finishes <sup>-1</sup>	before
meet		Right_Top (-B.l, 0)
Overlaps		Right_Top (-B.l, d <sub>2</sub> ) avec d <sub>2</sub> < 0
Finishes		Right_Top (-B.l, -A.h)
during (B.h > A.h)		Right (-B.l, (A.h - B.h)/2)
Starts		Right_Top (-B.l, -B.h)
equals (B.h = A.h)		Right (-B.Largeur, (A.h - B.h)/2)
starts <sup>-1</sup>		Right_Top (-B.l, -B.h)
during <sup>-1</sup> (B.h < A.h)		Right (-B.l, (A.h - B.h)/2)
finishes <sup>-1</sup>		Right_Top (-B.l, -A.h)
overlaps <sup>-1</sup>		Right_Bottom (-B.l, d <sub>2</sub> ) avec d <sub>2</sub> < 0
meet <sup>-1</sup>		Right_Bottom (-B.l, 0)
before <sup>-1</sup>		Right_Bottom (-B.l, d <sub>2</sub> ) avec d <sub>2</sub> > 0
overlaps <sup>-1</sup>		before
	meet	Right_Top (d <sub>1</sub> , 0) avec d <sub>1</sub> < 0
	Overlaps	Right_Top (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0, d <sub>2</sub> < 0
	Finishes	Right_Top (d <sub>1</sub> , -A.h) avec d <sub>1</sub> < 0
	during (B.h > A.h)	Right (d <sub>1</sub> , (A.h - B.h)/2) avec d <sub>1</sub> < 0
	Starts	Right_Top (d <sub>1</sub> , -B.h) avec d <sub>1</sub> < 0, d <sub>2</sub> > 0
	equals (B.h = A.h)	Right (d <sub>1</sub> , (A.h - B.h)/2) avec d <sub>1</sub> < 0
	starts <sup>-1</sup>	Right_Top (d <sub>1</sub> , -B.h) avec d <sub>1</sub> < 0
	during <sup>-1</sup> (B.h < A.h)	Right (d <sub>1</sub> , (A.h - B.h)/2) avec d <sub>1</sub> < 0
	finishes <sup>-1</sup>	Right_Top (d <sub>1</sub> , -A.h) avec d <sub>1</sub> < 0
	overlaps <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0 et d <sub>2</sub> < 0
	meet <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , 0) avec d <sub>1</sub> < 0
	before <sup>-1</sup>	Right_Bottom (d <sub>1</sub> , d <sub>2</sub> ) avec d <sub>1</sub> < 0 et d <sub>2</sub> > 0
	meet <sup>-1</sup>	before
meet		Right_Top (0, 0)
Overlaps		Right_Top (0, d <sub>2</sub> ) d <sub>2</sub> < 0
Finishes		Right_Top (0, -A.h)
during (B.h > A.h)		Right (0, (A.h - B.h)/2)
Starts		Right_Top (0, -B.h) avec d <sub>1</sub> , d <sub>2</sub> > 0
equals (B.h = A.h)		Right (0, (A.h - B.h)/2)
starts <sup>-1</sup>		Right_Top (0, -B.h)
during <sup>-1</sup> (B.h < A.h)		Right (0, (A.h - B.h)/2)
finishes <sup>-1</sup>		Right_Top (0, -A.h)
overlaps <sup>-1</sup>		Right_Bottom (0, d <sub>2</sub> ) avec d <sub>2</sub> < 0
meet <sup>-1</sup>		Right_Bottom (0, 0)
before <sup>-1</sup>		Right_Bottom (0, d <sub>2</sub> ) avec d <sub>2</sub> > 0

before <sup>-1</sup>	before	Right_Top ( $d_1, d_2$ ) avec $d_1, d_2 > 0$
	meet	Right_Top ( $d_1, 0$ ) avec $d_1 > 0$
	Overlaps	Right_Top ( $d_1, d_2$ ) avec $d_1 > 0, d_2 < 0$
	Finishes	Right_Top ( $d_1, -A.h$ ) avec $d_1 > 0$
	during ( $B.h > A.h$ )	Right ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	Starts	Right_Top ( $d_1, -B.h$ ) avec $d_1, d_2 > 0$
	equals ( $B.h = A.h$ )	Right ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	starts <sup>-1</sup>	Right_Top ( $d_1, -B.h$ ) avec $d_1$
	during <sup>-1</sup> ( $B.h < A.h$ )	Right ( $d_1, (A.h - B.h)/2$ ) avec $d_1 > 0$
	finishes <sup>-1</sup>	Right_Top ( $d_1, -A.h$ ) avec $d_1 > 0$
	overlaps <sup>-1</sup>	Right_Bottom ( $d_1, d_2$ ) avec $d_1 > 0$ et $d_2 < 0$
	meet <sup>-1</sup>	Right_Bottom ( $d_1, 0$ ) avec $d_1 > 0$
	before <sup>-1</sup>	Right_Bottom ( $d_1, d_2$ ) avec $d_1 > 0$ et $d_2 > 0$

## Annexe 2

Nous donnons ci-dessous, le Schéma XML défini pour la validation des documents XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="DOCUMENT_MULTIMEDIA">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="DOCUMENT_PRIMAIRE"/>
        <xs:element ref="DOCUMENT_DERIVE" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--=====-->
  <!--===== LES TYPES =====-->
  <!--=====-->
  <xs:simpleType name="ObjectType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Video"/>
      <xs:enumeration value="Audio"/>
      <xs:enumeration value="Image"/>
      <xs:enumeration value="Texte"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="TypeDestination">
    <xs:choice>
      <xs:element name="objet" type="xs:string"/>
      <xs:element name="instant" type="xs:nonNegativeInteger"/>
    </xs:choice>
  </xs:complexType>

  <xs:simpleType name="TypeParamètre">
    <xs:restriction base="xs:string">
      <xs:pattern value="[\-+]?[0-9]*|-\-"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="TypeParamètrePositif">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]*|-\-"/>
    </xs:restriction>
  </xs:simpleType>
  <!--=====-->
  <!--===== RELATIONS SPATIALES =====-->
  <!--=====-->
  <xs:element name="RelationSpatiale">
    <xs:complexType>
      <xs:sequence>
        <xs:choice>
          <xs:element name="A_Droite">
            <xs:complexType>
              <xs:attribute name="Dmin" type="TypeParamètre" use="required"/>
              <xs:attribute name="Dmax" type="TypeParamètre" use="required"/>
            </xs:complexType>
          </xs:element>
          <xs:element name="A_Gauche">
            <xs:complexType>
```



```

<xs:element name="DécalageInférieur">
  <xs:complexType>
    <xs:attribute name="Dmin" type="TypeParamètre " use="required"/>
    <xs:attribute name="Dmax" type="TypeParamètre " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="DécalageSupérieur">
  <xs:complexType>
    <xs:attribute name="Dmin" type="TypeParamètre " use="required"/>
    <xs:attribute name="Dmax" type="TypeParamètre " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="RépartitionHorizontale">
  <xs:complexType>
    <xs:attribute name="D" type="TypeParamètre " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="RépartitionVerticale">
  <xs:complexType>
    <xs:attribute name="D" type="TypeParamètre " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="répartitionVerticale"/>
<xs:element name="CentrageVertical"/>
<xs:element name="CentrageHorizontal"/>
<xs:element name="LargeurEgale"/>
<xs:element name="HauteurEgale"/>
</xs:choice>
<xs:element name="ObjetRéférence" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<!--=====-->
<!--===== RELATIONS TEMPORELLES=====-->
<!--=====-->
<xs:element name="RelationTemporelle">
  <xs:complexType>
    <xs:sequence>
      <xs:choice>
        <xs:element name="Starts"/>
        <xs:element name="Finishes"/>
        <xs:element name="Equals"/>
        <xs:element name="CoBegin">
          <xs:complexType>
            <xs:attribute name="Dmin" type="TypeParamètrePositif " use="required"/>
            <xs:attribute name="Dmax" type="TypeParamètrePositif " use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="Before">
          <xs:complexType>
            <xs:attribute name="Dmin" type="TypeParamètrePositif " use="required"/>
            <xs:attribute name="Dmax" type="TypeParamètrePositif " use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="BeforeEndOf">
          <xs:complexType>
            <xs:attribute name="Dmin" type="TypeParamètrePositif " use="required"/>
            <xs:attribute name="Dmax" type="TypeParamètrePositif " use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="CoEnd">
  <xs:complexType>
    <xs:attribute name="Dmin" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="Dmax" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="While">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Delayed">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="StartIn">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="EndIn">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Cross">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="Overlaps">
  <xs:complexType>
    <xs:attribute name="D1min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D1max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D2max" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D3min" type="TypeParamètrePositif " use="required"/>
    <xs:attribute name="D3max" type="TypeParamètrePositif " use="required"/>
  </xs:complexType>
</xs:element>
</xs:choice>
<xs:element name="ObjetRéférence" type="xs:string"/>
</xs:sequence>
</xs:complexType>

```

```

</xs:element>
<!--=====-->
<!--===== OBJET =====>
<!--=====-->
<xs:element name="OBJET">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="RelationTemporelle" maxOccurs="unbounded"/>
      <xs:element ref="RelationSpatiale" maxOccurs="unbounded"/>
      <xs:element name="Destination" type="TypeDestination" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="NomObjet" type="xs:string" use="required"/>
    <xs:attribute name="AdresseObjet" type="xs:anyURI" use="required"/>
    <xs:attribute name="TypeObjet" type="ObjectType" use="required"/>
    <xs:attribute name="Droite" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="Gauche" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="Sup" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="Inf" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="VMilieu" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="HMilieu" type="xs:nonNegativeInteger" use="required"/>
    <xs:attribute name="Début" type="TypeParamètrePositif" use="required"/>
    <xs:attribute name="Fin" type="TypeParamètrePositif" use="required"/>
    <xs:attribute name="Durée" type="TypeParamètrePositif" use="required"/>
    <xs:attribute name="DuréePref" type="TypeParamètrePositif" use="required"/>
    <xs:attribute name="DuréeMax" type="TypeParamètrePositif" use="required"/>
    <xs:attribute name="DuréeMin" type="TypeParamètrePositif" use="optional" default="0"/>
    <xs:attribute name="Cliquable" type="xs:boolean" use="optional" default="false"/>
    <xs:attribute name="Interactionnel" type="xs:boolean" use="optional" default="false"/>
  </xs:complexType>
</xs:element>
<!--=====-->
<!--===== SCENE =====>
<!--=====-->
<xs:element name="SCENE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="OBJET" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--=====-->
<!--===== DOCUMENT DERIVE =====>
<!--=====-->
<xs:element name="DOCUMENT_DERIVE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SCENE" maxOccurs="unbounded"/>
      <xs:element ref="PARCOURS" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--=====-->
<!--===== DOCUMENT PRIMAIRE =====>
<!--=====-->
<xs:element name="DOCUMENT_PRIMAIRE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="OBJET" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```

</xs:element>
<xs:element name="SCENE_PARCOURS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SCENE"/>
    </xs:sequence>
    <xs:attribute name="Ordre" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
<!--=====-->
<!--=====PARCOURS=====-->
<!--=====-->
<xs:element name="PARCOURS">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="SCENE_PARCOURS" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```