INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : « Informatique : système et communication »

préparée au laboratoire de l'Institut National de Recherche en Informatique et en Automatique dans le cadre de **l'Ecole Doctorale « Mathématiques, Sciences et technologies de l'Information »**

Présentée et soutenue publiquement

par

Tien TRAN THUONG

le 03/02/2003

Titre :

Modélisation et traitement du contenu des médias pour l'édition et la présentation de documents multimédias

Directeur de thèse : Mme. Cécile ROISIN

JURY

M. Roger MOHR	, Président
M. Liming CHEN	, Rapporteur
M. Jean-Claude DUFOURD	, Rapporteur
Mme. Cécile ROISIN	, Directeur de thèse
M. Yves CHIARAMELLA	, Examinateur
Mme. Christine VANOIRBEEK	, Examinateur

Remerciements

Pour commencer, je tiens à remercier Cécile Roisin et Vincent Quint pour m'avoir accueilli dans le projet Opéra voici quelques années. Je remercie également Roger Mohr qui a été le premier à m'accueillir à l'INRIA et Augustin Lux qui a accepté de m'initier à la recherche pendant le DEA.

Je remercie tout particulièrement Cécile Roisin pour m'avoir guidé et supporté durant cette thèse et pour ses nombreuses relectures du mémoire.

Je tiens à remercier les membres du jury :

Jean-Claude Dufourd directeur d'étude à l'ENST et Liming Chen professeur à l'ECL pour avoir jugé mon travail,

Roger Mohr professeur à l'INPG, Yves Chiaramella professeur à l'UJF, directeur de l'IMAG et Christine Vanoirbeek chercheur à l'EPFL de m'avoir fait l'honneur de participer à mon jury de thèse.

Je remercie Muriel Jourdan, Jean-Yves Vion Dury, Vincent Kober et Frédéric Bes qui ont accepté la tâche de la lecture des premières versions du manuscrit et m'ont permis de l'améliorer.

Je tiens à remercier également tous mes collègues passés et présents du projet Opéra par qui leurs conseils, leurs encouragements et leur aide ont contribué à l'aboutissement de cette thèse, à savoir Lionel Villard contributeur important de Madeus 2.0 avec qui j'ai fait mes tous premiers pas dans Madeus, Nabil Layaïda pour m'avoir consacré du temps en répondant à mes nombreuses questions, Muriel Jourdan, Frederic Bes, Tayeb Lemlouma, Laurent Carcone, Laurent Garçon, Laurent Tardif, Vincent Kober, Julien Guyard, Loay Sabry, Irène Vatton et tous les autres.

Enfin et surtout, je remercie toute ma famille, mes parents sans qui je ne serais pas là et ma sœur. Je remercie également mes professeurs, mes amis au VietNam et tous mes amis vietnamiens à Grenoble.

Pour finir, je remercie tout particulièrement ma femme, pour m'avoir accompagné et soutenu tout au long de mes études, pour m'avoir offert un petit ange qui m'a beaucoup encouragé.

RESUME en français

Les travaux de cette thèse proposent une nouvelle voie qui permet d'éditer/présenter plus facilement des documents multimédias sophistiqués. Elle consiste à modéliser le contenu des médias complexes (vidéo, audio) en sous-éléments (objets en mouvement, plans, scènes). Ainsi, à ces objets internes à un média peuvent être associés des comportements (hyperliens) ou des relations spatiales ou temporelles avec d'autres objets du document de façon à obtenir des présentations multimédias plus riches. Outre l'objectif de couvrir les besoins de description des auteurs pour réaliser des synchronisations fines entre médias, la difficulté majeure de ce travail a consisté à assurer que ce modèle reste cohérent par rapport au modèle de composition de documents. L'approche choisie consiste à utiliser les outils de description de MPEG-7 pour décrire les médias et à intégrer ces descriptions au sein d'une extension du modèle de document à base de contraintes Madeus.

TITRE en anglais

Media content modelling and processing for authoring and presenting multimedia documents

RESUME en anglais

This work proposes a new way to edit/present easily multimedia documents. It consists in modelling the contents of complex media (video, audio) as a structure of sub-elements (moving objects, shots, scenes). These internal media fragments can be associated with behaviors (hyperlinks) or spatial/temporal relations with other objects of the document. This enables richer multimedia presentations thanks to a finer synchronization between media. The difficulty of this work is to insure that this model remains consistent with the composition model of multimedia documents and that it covers the needs of the authors for multimedia fine-grained synchronization. The approach chosen consists in using description tools from MPEG-7 to describe media contents and in integrating these descriptions into an extension of the Madeus constraint-based composition model.

DISCIPLINE - SPECIALITE DOCTORALE

Informatique : Système et Communication

MOTS-CLES

document multimédia, synchronisation fine, document structuré, description du contenu, édition de document, animation abstraite, MPEG-7, SMIL.

Unité de Recherche INRIA Rhône-Alpes : Zirst - 655 avenue de l'Europe - Montbonnot - 38334 Saint Ismier Cedex – France

Table des matières

Table des figures

Chapitre I. Introduction

I.1 Motivations

Deux grands moyens de communication d'information sont entrés dans une phase de convergence résultant de leur capacité à transmettre un même type d'information : le *document multimédia*. En effet, le réseau Internet, qui fournit à travers le Web un accès à l'information hypertexte, offre maintenant la possibilité de synchroniser les objets médias dans les documents hypertextes. De son côté, le système de communication télévisuelle, qui fournissait jusqu'à présent des informations audiovisuelles passives, commence à offrir des moyens d'interaction et d'enrichissement de l'information diffusée. Les deux standards qui illustrent ces nouveaux services sont SMIL (synchronisation pour le Web) et MPEG-4 (diffusion de médias enrichis). Cette évolution fait exploser les besoins en documents multimédias et, par conséquence, pousse le développement d'outils de création, de production et de diffusion de ces types de documents. Pour les concepteurs de sites ou les producteurs de programmes audiovisuels, la création des documents multimédias devient donc de plus en plus importante.

La réalisation d'un document multimédia est la mise en synchronisation de l'ensemble des médias (textes, images, vidéos, sons, etc.) en différentes dimensions : logique, temporelle et spatiale. Il existe déjà plusieurs types de *modélisation* pour cette mise en forme. Ces modèles sont proposés dans des standards comme HyTime, MHEG et SMIL ou ont fait l'objet de projets de recherche comme CMIF, ZYX, Madeus, etc. Cependant, ces modèles considèrent les objets médias comme des boîtes noires. En conséquence, la synchronisation (temporelle et spatiale) entre les médias est limitée par cette granularité du niveau média et il est difficile d'exprimer des synchronisations plus fines, entre deux fragments de média par exemple. Or il est clair que la plupart des médias ont un contenu riche comme l'image avec ses objets au premier plan et son deuxième plan ; la vidéo avec sa structure encore plus complexe comprenant des scènes, des plans, des objets, des événements, etc. ; le média textuel lui-même a une structure qui pourrait être exploitée lors de la composition multimédia : caractères, mots, phrases, etc. ; certains médias ont déjà une structure codée comme HTML, SVG, MathML qui peut être utile de la synchronisation fine. Les exemples de scénarios qui utilisent cette information interne aux médias sont nombreux ; par exemple : la présentation d'un commentaire textuel lorsque de l'apparition d'un personnage

dans une vidéo ; la sélection d'un mot dans une phrase de texte synchronisée avec une portion de flux audio ; un hyperlien sur un objet vidéo ou sur une région particulière d'une image ; *etc.* Ces scénarios sont très difficiles à réaliser avec des synchronisations à gros grain. Dans SMIL par exemple, ils sont spécifiés de façon absolue par des valeurs fixes (ex. begin="3s") identifiant le début du fragment synchronisé. Un si bas niveau sémantique de spécification empêche les outils d'analyse et de recherche de traiter le contenu de tels documents multimédias.

En parallèle à ces nouveaux besoins de création, un besoin de gestion de bases de données multimédia émerge également : les bases deviennent énormes, notamment parce que les médias comme la vidéo et le son nécessitent de très grandes capacités de stockage. Pour assurer le traitement, la recherche et l'accès à ces contenus, des modèles de description de média ont été définis. Beaucoup d'efforts de recherche ont été consacrés à ces modèles pour standardiser la description du contenu multimédia. Parmi eux, le plus important est la norme MPEG-7, connue comme 'l'Interface de Description du Contenu de Multimédia", qui vise à fournir des technologies fondamentales standardisées permettant la description de contenu de données audiovisuelles dans les environnements multimédias. L'utilisation de la description du contenu des médias pour les applications de l'indexation et la recherche multimédia est déjà évidente. Par contre, l'utilisation de ce type de description dans des applications d'édition/présentation de documents multimédias est encore très rare et reste même encore un objectif. Par exemple [Rutledge et al. 01b] envisage d'utiliser la description MPEG-7 pour intégrer plus finement des fragments de média dans le document Web. Pourtant, si le contenu des médias est décrit à un plus haut niveau sémantique, cette information sera disponible pour le processus de composition de document multimédia et permettra d'élaborer les scénarios plus riches comme ceux cités ci-dessus. Aussi, si la description du contenu de média est si utile pour éditer et présenter le document multimédia, pourquoi n'est elle pas encore utilisée largement ? Même dans le standard récent SMIL 2.0 ce type de spécification n'est pas pris en compte. L'étude pour obtenir la possibilité d'utiliser la description dans la composition de documents multimédias est donc nécessaire.

La réalisation de documents multimédias peut être effectuée facilement en utilisant des outils d'édition multimédia actuels (ex. DoCoMo01, LimSee, Director 8.5, GRiNS 2.0). Ces outils proposent des interfaces graphiques sur lesquelles l'auteur peut manipuler directement des objets médias pour les synchroniser entre eux. Cependant les interfaces sont encore insuffisantes pour les scénarios que nous visons car la visualisation à gros grain des objets médias sur les interfaces graphiques ne permet pas à l'auteur de spécifier une portion de média à synchroniser. En conséquence, les scénarios mentionnés ci-dessus sont très difficiles à éditer. Pendant l'édition des synchronisations fines, il est nécessaire d'offrir le moyen de visualiser la structure du contenu des médias et de sélectionner facilement tous les éléments dans cette structure pour les synchroniser avec d'autres objets médias.

De plus, les systèmes multimédia existants sont souvent indépendants des outils de recherches et de traitement des médias. Avant de commencer à éditer son scénario, l'auteur doit dépenser beaucoup d'efforts à chercher des médias puis à les traiter pour avoir enfin une bonne collection des médias à intégrer dans le

document multimédia. Existera-t-il dans l'avenir un système multimédia qui permette à l'auteur de décrire simplement une présentation multimédia souhaitée, et avec lequel les médias appropriés à la présentation seront assemblés et mis en forme automatiquement ? Pour cela, l'une des premières étapes est d'avoir un modèle de description du contenu des médias adapté à l'édition de document multimédia et une bonne intégration de ce modèle avec les modèles de document multimédia.

Le travail de cette thèse a pour but de contribuer aux thèmes de l'édition et de la présentation de documents multimédias. L'édition doit permettre d'utiliser la description du contenu des médias pour intégrer plus finement ces fragments de médias dans le scénario et avoir ainsi des documents plus sophistiqués tout en gardant la structure logique de ces médias lors de leur synchronisation dans le document.

I.2 Objectifs de la thèse

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

- ♦ Définir ou choisir un modèle de description du contenu des médias qui soit approprié à l'édition et la présentation de documents multimédias ;
- ♦ Définir ou choisir un modèle de document multimédia qui permette d'utiliser des descriptions du contenu de médias ci-dessus ;
- ♦ Définir ou choisir un modèle d'animation abstraite qui permette de réutiliser des définitions d'animations dans les documents multimédia ;
- ♦ Définir une nouvelle architecture pour un environnement auteur qui puisse offrir le moyen de générer et d'éditer les descriptions du contenu des médias, puis de les utiliser lors de l'édition de documents multimédias et enfin présenter le document résultant. Cette architecture doit bénéficier au maximum des modèles mentionnés ci-dessus.

Les objectifs présentés ci-dessus se situent clairement dans un contexte applicatif. Par conséquent, toute proposition pour répondre à ces objectifs doit être validée à travers la réalisation de prototypes. En particulier, une première validation des modèles et outils proposés pourra être effectuée à travers la réalisation des scénarios multimédias comportant des synchronisations fines comme décrit en I.1.

I.3 Contexte de la thèse

Pendant mon stage de DEA effectué dans les projets Movi et Opéra de l'INRIA Rhône-Alpes, j'ai initié un premier travail dans le cadre d'une collaboration entre le projet d'analyse de vidéo (Vidéoprep) de Movi et le projet d'environnement auteur multimédia (Madeus) fait dans Opéra. La collaboration porte sur la traduction des structures du média vidéo issues de Vidéoprep en format XML. L'objectif était de s'affranchir du codage spécifique utilisé dans Vidéoprep pour permettre d'utiliser ses descriptions dans une application d'édition et de présentation de documents multimédias comme Madeus.

Cette thèse, qui s'est déroulée au sein du projet Opéra, est une continuation de ce premier travail avec un objectif plus général que le traitement des structures des fichiers d'analyse de Vidéoprep.

Le projet Opéra s'intéresse aux documents électroniques : documents techniques, hypertextes, multimédias, etc. Opéra étudie des modèles de documents qui rendent compte à la fois de leur organisation logique, de leur présentation graphique, de leur enchaînement temporel et des contenus multimédias. Il met également au point des techniques d'édition et de présentation qui s'appuient sur ces modèles.

Le projet Opéra a abordé les recherches sur le multimédia depuis 1994 selon trois directions complémentaires : la modélisation de l'information temporelle [Layaïda 97], la conception d'environnements auteur [Jourdan et al. 98a] et [Jourdan et al. 98b] et la conception de systèmes de présentation multimédia performants [Sabry 99]. Ces travaux sont caractérisés par une approche de spécification à base de contraintes car le modèle est en effet fondé sur les relations temporelles d'Allen [Allen 83], dont une partie a été adaptée au placement spatial. Le format source des documents Madeus est spécifié sous forme d'une DTD XML.

Madeus est un système d'édition et de présentation de documents structurés multimédia, dont la première version a été développée dans le cadre des travaux de [Layaïda 97] et [Sabry 99]. Ce travail posait les bases d'une édition à base de relations spatiales et temporelles. Il fut étendu par [Tardif 00], qui intégra des résolveurs de contraintes pour renforcer les capacités d'édition et de formatage du système. Une vraie refonte de Madeus a été menée dans le travail de [Villard 02] qui a rendu le modèle plus souple pour l'édition et l'adaptation de présentations multimédias. Cette refonte comprend aussi la modélisation de l'abstraction, de la structuration du contenu des médias et de la synchronisation fine effectuée dans le travail présenté ici pour l'édition et la présentation de documents multimédias complexes.

I.4 Plan de la thèse

La suite de ce mémoire est organisée en deux parties. La première partie présente un état de l'art (les chapitres II, III et IV) et la seconde partie décrit à la fois les aspects de modélisation de notre contribution (le chapitre V) et les implantations qui en découlent (le chapitre VI).

Chapitre II. Analyses des besoins d'un nouveau système multimédia. Ce chapitre introduit une évolution de trois générations de documents multimédias vers un multimédia sémantique dans lequel la production de documents multimédias est de plus en plus automatisée. Mais il n'existe pas encore de solutions concrètes pour y aboutir, à cause des complexités du document multimédia. Ce chapitre montre ces complexités à travers l'étude des concepts de base des documents multimédias. Puis, en étudiant la production de document multimédia dans les trois générations identifiées, les limitations des systèmes actuels sont identifiées. Les solutions et les systèmes idéaux sont aussi proposés pour chaque génération. De ce fait, les besoins de modélisation du contenu des médias pour les applications d'édition et de présentation de documents multimédias sont montrés.

Chapitre III. Modélisation de multimédia. Ce chapitre fait une étude complète de la modélisation multimédia en trois niveaux : analyse, description et intégration. L'étude de l'analyse du contenu des médias permet d'identifier les caractéristiques

de chaque type de média (vidéo, audio, image et texte) et de donner une vision des possibilités d'analyse dans ce domaine. L'étude de la description de média, qui est effectuée dans deux contextes : les standards et les travaux de recherche, permet de spécifier un modèle non seulement complet par assemblage des cas spécifiques, mais aussi générique en se basant sur les standards, parmi lesquels la norme MPEG-7 semble la plus importante. Enfin, l'étude de la modélisation des documents multimédias effectuée selon les besoins d'intégration fine entre segments de média montre les limitations des modèles actuels, caractérisés par un accès à gros grain et absolu dans le contenu des médias. Une intégration du modèle de description de média et du modèle d'intégration de média est donc proposée.

Chapitre IV. Applications multimédias. Ce chapitre cherche un environnement auteur idéal pour les applications multimédias actuelles. Cette recherche classe ces applications en trois catégories : indexation, production et intégration multimédia. Dans chaque catégorie, on peut trouver une partie des caractéristiques adaptées à l'environnement auteur idéal. On propose donc un enchaînement souple de ces trois catégories pour une architecture d'environnement auteur plus confortable. La souplesse de cet enchaînement peut être donnée par un modèle multimédia d'intégration mentionné dans le chapitre III. et implémenté dans le chapitre V.

Chapitre V. Modèles de description du contenu des médias et de leur intégration dans les documents. Ce chapitre présente le modèle d'intégration de la description de média et de l'intégration de média. Un modèle de description du contenu pour l'édition et la présentation de documents multimédias est donc proposé. Ce modèle prend en compte plusieurs approches étudiées dans l'état de l'art, de manière à s'adapter à la composition multimédia. Une implémentation de ce modèle se basant sur les schémas standard de la norme MPEG-7 est ensuite décrite. Dans un deuxième temps on décrit un modèle de sous-éléments qui permet d'intégrer le modèle de média dans le modèle Madeus. Enfin un modèle d'animation abstrait est décrit et ensuite déployé dans le modèle Madeus, grâce au modèle des sous-éléments.

Chapitre VI. Mdéfi : Un environnement auteur pour l'intégration fine de média. Ce chapitre décrit l'implémentation de l'environnement auteur *Mdéfi* qui s'appuie sur l'architecture mentionnée dans le chapitre IV et sur le modèle d'intégration proposé dans le chapitre V. Il fournit des outils pour analyser le contenu de la vidéo, générer/modifier des descriptions du contenu et les exploiter dans la composition du document multimédia.

Chapitre VII. Conclusion. Dans ce dernier chapitre, nous effectuons un résumé sur l'apport essentiel de cette thèse. Nous tirons aussi le bilan des réalisations ainsi que les perspectives de recherche suggérées par ce travail.

Chapitre II. Analyses des besoins d'un nouveau système multimédia

II.1 Vers le multimédia sémantique

Le "document électronique" est l'un des principaux composants qui concrétise l'ère de l'information. En fait, une part de plus en plus importante de l'information de notre monde, comme des ouvrages anciens, des oeuvres d'art ou des images des premiers symboles gravés sur des roches datant d'avant Jésus-Christ, sont aujourd'hui numérisés pour faciliter leur stockage, leur conservation, leur traitement ainsi que leur accès. De plus, grâce au Web et à l'Internet, le document électronique peut être largement distribué dans le monde. Cet ensemble d'information constitue non seulement la plus grande base de connaissance jamais vue, mais aussi le moyen de transport et d'échange d'information le plus rapide qui soit.

Parce qu'il joue un rôle si important, le domaine du document électronique suscite beaucoup de travaux de recherche et en conséquence a une vitesse d'évolution très rapide. Nous pouvons analyser cette évolution à travers la décomposition des systèmes de documents en trois générations [Ossenbruggen et al. 01] [Decker et al. 00] [Jourdan et al. 01] :

- *Première génération* : les documents sont édités "à la main" à l'aide d'éditeurs de documents comme *Word*, *FrameMaker*, *Amaya*, dans lesquels le modèle de document intègre de façon plus ou moins forte les données de présentation et le style ou même la logique de présentation. Le processus de production de document est constitué du formatage qui peut être intégré ou non dans l'éditeur (éditeur de texte avec formateur *LaTex* …) ;
- *Deuxième génération* : c'est une étape intermédiaire vers la production automatique de documents dans laquelle le modèle de génération du document est constitué de trois composants : le modèle métier, le modèle de présentation et la feuille de style qui contient un ensemble de règles pour la transformation des documents de type métier en documents de type présentation [Jourdan et al. 01] [Ossenbruggen et al. 01] [Villard 02] ;
- *Troisième génération* : c'est l'évolution vers le document sémantique, où la machine non seulement présente des informations brutes mais surtout est capable d'interpréter la sémantique de ces informations. Un exemple de document dans cette génération est bien connu comme l'avenir du Web, c'est le *Web sémantique* [W3C SW01].

Il est à noter qu'une génération de documents électroniques ne remplace pas la génération précédente. Elles se recouvrent l'une et l'autre. En effet, chaque génération développe un aspect nouveau du domaine du document électronique. Par exemple, la deuxième génération utilise des modèles de documents de la première génération comme modèles cibles de la transformation pour générer des présentations dynamiques au lieu de les spécifier à la main. La troisième génération vise à apporter un contenu plus significatif à la structure du document : pour cela, des modèles de métadonnées sont utilisés côte à côte avec des modèles de la structure de document de deuxième et première générations.

Les évolutions des trois générations décrites ci-dessus sont loin d'être achevées. En fait, il n'y a que pour le document statique, dont le contenu est principalement basé sur des textes, que la deuxième génération est réalisée et que la troisième génération commence à être mise en oeuvre. Par contre, le document multimédia, dont le contenu résulte de l'intégration de différents média comme *texte*, *image*, *audio*, *vidéo*, est seulement en train de prendre en compte les techniques de la deuxième génération.

La raison de ce retard est que le traitement du document multimédia est plus complexe que celui du document textuel. La présentation d'un document multimédia est orchestrée dans le temps et la mise en page de ce type de document est plus riche que celle d'un flot de caractères comme dans les documents textuels. D'autre part, les systèmes multimédias actuels n'implantent pas de politique avancée pour traiter des composants complexes comme la vidéo, l'audio et l'image. Ces médias sont en effet tous considérés comme des boîtes noires, ce qui ne permet pas de réaliser des compositions fines au sein des documents multimédias. Par exemple, des données textuelles sont traitées très finement dans les processeurs textuels (titre, paragraphe, phrase, liste, etc.) ; des liens hypertexte, des images ou des vidéos peuvent être associés à n'importe quelle partie du document textuel. Malheureusement, dans les systèmes multimédias les éléments textuels sont gérés à gros grain, comme des éléments basiques qui peuvent uniquement être affichés. Par exemple les documents HTML intégrés dans les présentations SMIL. Comme on le verra dans la section II.2 consacrée à l'étude de modèles multimédias, il n'existe pas encore de modèle assez flexible qui permette de composer parfaitement à grain fin plusieurs types de média dans un document selon les besoins des utilisateurs.

Il y a déjà des solutions partielles pour franchir ces limites. Elles sont envisagées dans les sections suivantes : dans la section II.2 ci-dessous, nous donnons les définitions nécessaires à cette étude, puis nous décrivons comment est réalisé le processus de production de documents multimédias en considérant les trois générations ci-dessus.

II.2 Concepts du document multimédia

Au sens générique, le terme "multimédia" se rapporte à une communication à travers plusieurs types de média. Un ensemble bien ordonné de médias permet une communication plus intéressante et plus dynamique qui peut mieux capter l'attention des interlocuteurs et l'information peut ainsi atteindre les destinataires plus efficacement. Parce qu'il a une telle efficacité, le multimédia est beaucoup utilisé dans notre vie. En effet, nous pouvons trouver de l'information multimédia partout dans les sites Web, dans des outils de formation assistée par ordinateur,

dans des kiosques interactifs, dans des systèmes d'accès à des manuels techniques, dans les jeux vidéo, dans l'annonce d'un nouveau produit, sur un cdrom ou un DVD de logiciel, etc. Ce déploiement des applications de nature multimédia est rendu possible par les évolutions technologiques comme la capacité et la puissance des ordinateurs, les possibilités de communication rapides (supports rapides, protocoles de bas et de haut niveaux comme ceux de l'Internet), les périphériques adaptés au multimédia (la carte vidéo, la carte son, le cdrom, et le DVD). Les couches logicielles elles aussi évoluent pour faciliter le déploiement du multimédia, que ce soit dans les domaines du système d'exploitation (Linux, Windows NT/2000/XP), des langages de programmation (C/C++, Java, Python, etc.) et des langages de structuration de l'information (XML). Mais qu'est donc au juste le multimédia ? La suite de cette section présente plus en détail ce qu'est un système multimédia.

II.2.1 Définition et composants du multimédia

Comme on l'a vu, le multimédia est une combinaison des présentations de plusieurs types de média selon une organisation structurale dans le temps et dans l'espace, et qui sont accédés de façon interactive par les utilisateurs. Plus précisément, le terme multimédia se retrouve à tous les niveaux qui composent un système informatique :

- ♦ l'information multimédia basique ou structurée,
- ♦ les programmes d'application,
- ♦ l'infrastructure logicielle et matérielle.

Nous étudions ces trois niveaux ci dessous.

L'information multimédia ou "document multimédia" peut être considérée comme un média composite où les éléments sont des médias qui forment ce média composite. En général, tous les médias numérisés peuvent être les éléments d'un document multimédia. On y trouve des médias traditionnels comme les médias statiques (texte, graphique), les médias continus (animation, audio et vidéo), les médias structurés (HTML, SMIL, SVG), ou même des programmes comme des *applets* ou *scripts*. Les médias composites sont créés de façon récursive à partir d'éléments multimédias. Une bonne combinaison des présentations de plusieurs éléments multimédias peut donner un meilleur résultat de présentation des informations. Par exemple, une présentation intégrant en même temps une vidéo avec une audio et des textes de commentaires est souvent plus efficace que les trois présentations indépendantes.

La composition d'éléments multimédias est définie selon plusieurs dimensions :

a) La structure de style permet de décorer ou paramétrer la présentation des éléments de multimédia. Par exemple, la taille et la police du texte, la vitesse d'affichage des images de la vidéo, le volume de l'audio, etc.
b) La structure spatiale permet de présenter graphiquement ou mettre en page des éléments de multimédia.
c) La structure temporelle est la dimension spécifique du multimédia. Elle permet de définir l'évolution du contenu du multimédia dans le temps, par opposition aux présentations fixes des documents statiques traditionnels.

d) L'interactivité est aussi un caractère intéressant du multimédia. Elle donne à l'utilisateur la possibilité d'interagir avec une présentation du multimédia. Cette dimension est aussi considérée comme une dimension sémantique du multimédia qui permet d'organiser le contenu d'une présentation de manière non linéaire. L'utilisateur peut donc suivre des liens sémantiques dans la présentation pour accéder aux parties qui l'intéressent.
e) L'ajout d'effets d'animation permet de dynamiser la présentation d'un élément ou d'un groupe d'éléments multimédias. Elle a un rôle important dans la présentation multimédia. Sans animation, une présentation multimédia peut devenir terne même si elle est bien définie dans les autres dimensions. Aujourd'hui, quand il manque de vrais systèmes multimédia, l'animation est utilisée pour augmenter la dynamique du document. On peut constater par ailleurs que de nombreux langages/outils non multimédias proposent des moyens pour introduire des effets d'animation : la qualité des présentations *PowerPoint* a été améliorée depuis qu'il supporte des effets d'animation ; HTML utilise des *scripts* et des *applets* pour animer des éléments de HTML, ce qui est connu sous le nom de *HTML dynamique* ; *Flash* avec son format binaire de l'animation (*swf*) est également utilisé sur le Web. De ce fait les présentations obtenues sont plus vivantes et plus attractives.

Le programme d'application multimédia : un programme informatique peut intégrer un ensemble d'éléments multimédias dans une présentation multimédia selon les dimensions ci-dessus. C'est alors une application multimédia. Une telle application multimédia peut être un produit multimédia comme les jeux sur cédérom, tel le *Strongth Old* (jeu de stratégie), et des cédéroms éducatifs tels que la série *Adibou* (histoires pour enfants). Ce peut être aussi un outil dédié à créer des présentations multimédia comme *Flash* de *MacroMedia*, *RealPresenter* et *RealSlideshow* de *RealNetworks*, *Adobe Premiere de Adobe*, ou *PowerPoint* de *Microsoft*. Un dernier type d'application multimédia est constitué des navigateurs de documents multimédias comme des outils d'édition et présentation des documents de type SMIL tels que *GRiNS* de *Oratrix*, *RealOne* et *SMIL Gens* de *RealNetworks*, et *Limsee* du projet de recherche *Opéra*. Bref les applications multimédias sont variées et utilisent des approches différentes, qui sont détaillées dans la section qui suit.

L'infrastructure logicielle et matérielle : le multimédia demande aussi des matériels et des logiciels de base adaptés : un ordinateur assez puissant ayant des périphériques spécifiques (carte graphique, carte son, haut-parleur, etc.) et des encodeurs/décodeurs de médias pour pouvoir prendre en charge les éléments de multimédia tels que la vidéo et l'audio. De plus, certains périphériques comme le clavier, la souris ou l'écran tactile, permettant à l'utilisateur d'interagir avec la présentation multimédia, sont également nécessaires. Enfin, une connexion Internet haut débit fournit l'accès à des systèmes multimédias disponibles à distance.

Un système multimédia complet est donc très complexe. Nous avons présenté ci-dessus des caractères généraux du système y compris ceux liés aux aspects logiciels et matériels. Dans la suite de cette thèse nous nous intéressons uniquement à la partie logicielle du multimédia. Nous étudions d'une part quels modèles de composition multimédia permettent d'exprimer des scénarios d'une présentation

multimédia (chapitre III et chapitre V) et d'autre part comment construire une application multimédia pour réaliser non seulement un programme de présentation multimédia, mais aussi un outil d'édition plus confortable pour les rédacteurs multimédia (chapitre IV et chapitre VI).

II.2.2 Applications multimédia

Une application multimédia peut être réalisée selon trois approches classées selon le format des produits multimédias : format *programme*, format de données *binaires* et format de données *textuelles*. Dans la suite de cette section, nous synthétisons un bref historique des applications multimédias à travers ces trois approches.

Dans un premier temps, grâce au développement des langages de programmation et des matériels multimédias, les concepteurs ont créé des programmes multimédias comme *Encarta, Atlas mondial, Myst*, *Microsoft gold* (jeu), ou *A.D.A.M* (formation à l'anatomie), *from Alice to ocean* (voyage en Australie). A cette époque la réalisation d'une application multimédia était coûteuse par les compétences des développeurs qu'elle demandait et par le temps de mise au point. C'est pourquoi ces développements sont réalisés au sein d'entreprises spécialisées et non par des concepteurs individuels. Selon cette approche, pour créer un produit multimédia, un projet de développement doit être établi, le projet doit être évalué selon sa durée, son coût de développement, la vie du produit, etc.

L'étape suivante a consisté à séparer les données des traitements pour :

- ♦ permettre la réutilisation,
- ♦ faciliter la réalisation d'applications multimédias,
- ♦ contrôler les coûts.

Cette évolution a permis le développement d'applications multimédias "ordinaires" comme la publicité sur le Web, les *SlideShows* assistant les présentations, ou les albums de photos familiales. L'application multimédia est devenue non seulement la machine de présentation multimédia, mais aussi un outil assistant les utilisateurs pour éditer des présentations multimédias. La tâche de création multimédia est donc plus facile avec cette approche de l'application multimédia comme *Flash* de *Macromédia*, *Adobe Illustrator*, *Realplayer*, *Picture It ! Photo*, *Microsoft Multimedia Player*, *PowerPoint*, *QuickTime*, etc. Cependant, le coût de la création multimédia est encore important, parce que les outils sont chers. De plus l'auteur a besoin d'une collection d'outils pour pré-traiter divers types de média avant de les composer, comme *Adobe Illustrator* et *Corel Draw* pour traiter les textes et les graphiques ; *PaintShop* et *PhotoShop* pour les images ; *Sound Forge* pour l'audio ; *Flash* de *Macromédia* pour les animations ; *Adobe Première* pour réaliser des montages de clips vidéo. Par ailleurs, les produits de ces outils sont en format binaire et propriétaire, ce qui rend difficile leur modification et leur échange entre auteurs. Ces formats favorisent également la sensibilité aux virus informatiques. Le format binaire n'est pas adapté à d'autres types d'utilisation, il rend difficile l'accès au contenu des médias, et toutes les informations concernant la structure sont perdues. Par exemple, une animation

Flash dans une page Web qui peut fonctionner de façon satisfaisante sur un PC, peut-elle encore s'exécuter correctement sur un PDA, ou sur des clients de faible puissance ? Comment associer à ces médias animés comme Flash des informations plus sémantiques, qui sont nécessaires aux outils de recherche ou qui permettent de rendre interopérables les applications multimédias ? Les produits en format binaire sont des boîtes noires desquelles il est très difficile d'extraire des fragments et des informations sémantiques. De plus dans le contexte de recherche vers le Web sémantique, un tel format binaire empêche la mise en oeuvre d'agents intelligents qui se chargeraient de traitements automatiques.

C'est pourquoi la plupart des standards multimédias sont textuels : HyTime [HyTime:ISO 97], SMIL [SMIL2.0 01], SVG [SVG1.1 03], MPEG-7 [MPEG-7]. Même MPEG-4 [MPEG-4] qui utilise un format binaire BIFS pour encoder les scènes multimédias a évolué vers un format textuel XMT [Kim et al. 00]. Ces standards sont tous basés sur les technologies SGML/XML. En effet ces technologies sont adaptées pour décrire de manière textuelle la structure et le scénario de documents multimédias. Ils permettent d'annoter le contenu par des informations sémantiques nécessaires aux outils de recherche. Le format textuel permet non seulement la séparation des données et des traitements, mais surtout leur indépendance et une accessibilité plus grande de l'information. Ainsi la réutilisation au niveau des fragments multimédias et l'interopérabilité parmi des applications multimédias différentes sont possibles. Enfin l'approche déclarative fournit beaucoup plus de flexibilité que les deux approches précédentes : cette flexibilité est en partie due à la séparation données/traitements évoquée précédemment, mais elle dépend aussi du modèle de spécification des scénarios multimédias qui intègre ou non des abstractions adaptées. Nous décrivons dans la sous-section II.2.3 ci-dessous ce que permettent de définir les modèles de document multimédia.

Ainsi, l'approche déclarative a tendance à faciliter la rédaction et l'utilisation multimédia, et à propager plus largement le multimédia dans tous nos moyens d'informations ordinaires, tel que le Web. Avec cette approche, le domaine des applications multimédias rejoint celui des documents électroniques où la présentation multimédia est issue d'un document multimédia, et la rédaction du document multimédia a besoin d'un modèle de document multimédia. En effet le document électronique aujourd'hui est le moyen le plus efficace et universel pour présenter, traiter et transporter des informations. Cependant, le mode de présentation des informations est encore limité, car la présentation est souvent statique et est principalement basée sur le texte. Le document électronique a besoin d'importer des technologies du multimédia pour traiter les divers types d'information, dynamiser et synchroniser les informations présentées. SMIL est un bon exemple de ce sens de l'évolution du document électronique.

Un autre exemple typique de cette tendance est l'évolution des travaux du groupe MPEG. Ses premiers travaux avec MPEG-1 et MPEG-2 permettent d'encoder des informations audiovisuelles, puis MPEG-4 et MPEG-7 permettent de décrire textuellement et d'annoter des scènes complexes d'une présentation multimédia. Enfin les travaux en cours de ce groupe visent à définir MPEG-21 [MPEG-21], un modèle de métadonnées du multimédia pour supporter la propriété, le droit de l'utilisation et l'interopérabilité parmi des applications multimédias.

II.2.3 Modèle de document multimédia

Selon l'approche déclarative du multimédia, un document multimédia décrit une composition d'une présentation multimédia. La composition doit être spécifiée selon une logique de composition, qui est appelée *modèle de document*. Le type de document (DTD) et le schéma (*XML Schema*) sont des bons exemples d'outils de spécification de modèle de document. On peut dire également que le modèle de document est la logique abstraite de la structure, qui identifie les caractéristiques communes d'une classe de documents. Les besoins de composition des documents multimédias ont fait l'objectif de nombreux travaux qui aboutissent tous à l'identification des quatre axes de composition (appelés aussi dimensions) tels que décrits en II.2.1 : *logique*, *temporel*, *spatial*, *hypermédia* [André et al. 89] [Hardman et al. 93] [Layaïda 97].

Par extension des travaux sur les documents structurés statiques, les modèles de composition logique, temporel et spatial utilisent une approche hiérarchique (organisation arborescente) tandis que la composition hypermédia conduit à une structure d'hypergraphe.

- La dimension logique permet d'organiser hiérarchiquement des informations multimédias selon un ordre logique de présentation, par exemple des *chapitres*, *sections*, *paragraphes*, etc.
- La dimension spatiale concerne la structure hiérarchique de l'organisation spatiale des éléments multimédias pendant la présentation.
- La dimension temporelle concerne l'arrangement hiérarchique des éléments multimédias dans le temps.
- La dimension hypermédia concerne des liens entre des portions de document. Les liens peuvent être intra ou extra document. Deux portions liées par un lien ont souvent des relations sémantiques. Dans ce cas, le lien hypermédia permet à l'utilisateur de naviguer dans l'espace sémantique de la présentation. L'utilisateur peut suivre des liens pour accéder aux contenus que il/elle veut consulter.

À ces quatre dimensions de base, il est nécessaire d'ajouter d'autres modèles de composition pour couvrir les besoins de spécification multimédia. En particulier, pour permettre de définir des comportements dynamiques sur des objets médias (ou des groupes d'objets), un modèle d'animation doit être intégré, le module d'animation de SMIL [Schmitz et al. 01] est un bon exemple de modèle répondant à ce type de besoin.

Des moyens de définition de synchronisations fines entre les éléments multimédias doivent également être offerts pour obtenir des présentations multimédias sophistiquées. Un document de type Karaoke, dans lequel des animations sur des fragments de texte (animation de colorisation) accompagnent les fragments d'audio auxquels correspondent les rythmes d'audio, ou la synchronisation dans des documents techniques entre l'audio, la vidéo et la description textuelle sont des applications typiques demandant une synchronisation fine. Non seulement un modèle permettant de mettre en relation fine des médias est nécessaire, mais aussi un modèle pour décrire explicitement les structures internes des médias. Ces besoins de composition impliquent donc l'existence de modèles intra média qui permettent de décrire les structures internes des médias. Les

travaux les plus représentatifs dans ce domaine sont ceux de MPEG-7. Enfin la mise en correspondance des deux modèles interne et externe doit être effectuée [Rutledge et al. 01b] [Tran_Thuong et al. 02a].

Parmi les quatre dimensions de base identifiées précédemment seulement les trois dimensions temporelle, spatiale et hypermédia sont modélisées concrètement. La spécification d'un modèle logique reste encore floue dans les modèles de document multimédia. En effet, la logique de présentation est très variée, elle dépend de chaque auteur, chaque domaine ou encore du but spécifique de la présentation. Tous les modèles qui supportent la logique de présentation sont toujours dédiés à une application spécifique. Par exemple, RealSlideShow utilise SMIL et un modèle avec une logique de présentation dédiée à présenter un ensemble d'images successives en parallèle avec une source sonore. Ou bien dans des systèmes d'adaptation de documents multimédias, le modèle de document du système est divisé en un modèle métier et un modèle de présentation [Villard 02] : le modèle métier permet d'exprimer une organisation logique structurée des sources d'information à transformer. Cependant le modèle métier est souvent dédié à une classe d'applications. C'est pourquoi tous les systèmes d'adaptation sont aussi limités aux modèles métier qu'ils supportent. La recherche d'un modèle commun à toutes les présentations multimédias et les domaines d'application est une difficulté pour aller vers un multimédia sémantique [Ossenbruggen et al. 01] dans lequel les systèmes doivent avoir la capacité d'interopérabilité.

Un axe important dans la recherche actuelle vers le multimédia sémantique concerne la notion de méta-modèle. L'intégration d'un méta-modèle dans le modèle de document multimédia permet à l'auteur de définir de l'information sémantique au contenu du document multimédia. Cela facilite beaucoup les tâches d'analyse, d'indexation, d'archivage ou de recherche des informations. Actuellement le modèle de document multimédia est défini localement, cela permet d'isoler les systèmes multimédias et leurs produits. Grâce à un méta-modèle qui permet d'ajouter des informations sémantiques à la fois dans les documents et dans les modèles du document, les systèmes multimédias différents peuvent se comprendre et présenter un document multimédia de n'importe quel autre système. Un exemple typique de cet axe est celui effectué dans [Hunter 01]. Elle consiste à annoter le modèle standard de description des informations audiovisuelles (MPEG-7) et à proposer une architecture permettant l'interopérabilité entre les différents standards de métadonnées comme MPEG-7, Dublin Core, INDECS et CIDOC. Cette architecture d'interopérabilité permet aux diverses applications multimédias de communiquer entre elles. On peut aussi citer le travail d'[Allsopp et al. 01] qui sous l'auspice de CoABS (Control of Agent-Based Systems de DARPA) a construit une infrastructure supportant la communication entre des agents différents. Le travail est basé sur le modèle générique de métadonnées (RDF) pour échanger des informations sémantiques entre des agents. L'utilisation récente du concept d'ontologie dans le domaine du Web sémantique peut permettre d'attacher des ressources multimédias aux connaissances connues. Il permet aussi de décrire la signification propre d'une ressource et ses relations avec les autres ressources. Une collection de ressources appropriées à un document multimédia peut être ainsi facilement ou même automatiquement obtenue pour assembler le document multimédia. En bref, l'utilisation des métadonnées dans le document multimédia ou

même au niveau plus générique pour le modèle de document multimédia permet de construire des systèmes multimédias plus intelligents, qui permettront de générer des présentations multimédias selon la demande de l'utilisateur. Ce type de système peut être connu sous le nom de multimédia sémantique.

II.2.4 Synthèse

Nous avons présenté dans cette partie des concepts généraux du multimédia et une vue globale des systèmes multimédias. Nous avons aussi évoqué rapidement l'existence de trois approches principales pour concevoir une application multimédia : l'approche de type *programmation*, l'approche par *génération binaire* et enfin l'approche *déclarative*. Parmi ces trois approches, l'approche déclarative a été mentionnée comme étant la plus flexible et celle qui fournit des facilités en création et en consultation des présentations multimédias. Ces travaux on été récemment concrétisés par la définition de standards multimédias comme SMIL, SVG, MPEG-4, MPEG-7. Cependant ces résultats sont seulement une première étape, ils ont encore des lacunes vis-à-vis de systèmes multimédias actuels ou à venir. Les travaux présentés dans cette thèse ont pour objectifs d'observer, d'analyser et de contribuer aux évolutions futures de cette approche.

Les deux concepts de l'approche déclarative, qui sont la notion de document multimédia et de modèle de document multimédia, ont été introduits comme les briques indispensables du domaine. Ils nous servent de connaissances de base pour aller plus loin dans ce domaine. Nous les développerons dans les chapitres III et V.

II.3 Le processus de production de document multimédia

Nous avons mentionné dans la section ci-dessus l'évolution rapide des technologies liées aux documents multimédias. Dans la section I.1 nous avons aussi présenté la tendance d'évolution des documents multimédias vers un système plus sémantique. En effet, cette évolution est fortement liée à la façon de produire des documents multimédias. C'est pourquoi dans cette section nous allons étudier ce processus de production. Nous allons commencer par les étapes standard, puis nous allons analyser comment est effectuée la composition de documents multimédias dans des systèmes multimédias différents, classifiés selon les trois générations d'évolution de documents multimédias décrits précédemment. Cette analyse nous permettra de caractériser ensuite les trois générations de document multimédia et de poser des problématiques pour chaque génération. Ce sont sur ces problématiques que les travaux présentés dans cette thèse s'appuient.

II.3.1 Les étapes du processus de production de document multimédia

Puisque le document multimédia est largement étudié et exploité, des études de processus de production de document multimédia sont aussi largement décrites dans la littérature de recherche [Bailey et al. 01a] [Bailey et al. 01b] ou même appliquées dans le monde industriel[1]. Parce que la compréhension claire des diverses étapes dans le processus de production du document multimédia peut directement influer sur le succès du produit multimédia développé, elle est pour un

[1] http://www.edb.utexas.edu/mmresearch/Students97/Rutledge/html/interviews.html

auteur/producteur plus importante et plus prioritaire que les habiletés de conception ou de programmation. Les chercheurs doivent eux aussi déterminer clairement les étapes de ce processus afin de positionner et valoriser leurs travaux de recherche [Tardif 00] [Bailey et al. 01a] [Bailey et al. 01b]. Le processus de production multimédia suit et adapte celui des projets de développement industriel. De ce fait, dans son déroulement le plus complet, les étapes suivantes ont été identifiées :

- *Arrivée d'une idée* : le concepteur a une entrevue avec le client pour déterminer ses besoins et les caractéristiques du produit futur. Le concepteur doit être clair et noter soigneusement cette entrevue.
- *Analyse de l'idée* : le designer analyse l'idée initiale de l'entrevue, essaye de restructurer cette idée initiale dans un langage plus formel, identifie des composants ainsi que des relations entre eux. En bref, la logique du produit doit être identifiée dans cette étape.
- *Confrontation des idées (Brainstorming)* : accueille des idées de construction du produit selon des idées générales des deux premières étapes concernant le contenu, l'apparition et l'organisation du produit, au début le concepteur accueille largement les idées de construction, puis avec le client le concepteur restreint ces idées et établit un cadre pour le produit pour éviter les évolutions possibles dans la phase de production.
- *Structuration (Outlining)* : une fois que le produit futur est cadré, le concepteur décrit la structure générale du produit. Les descriptions de la structure peuvent être dessinées sur des grands papiers et affichées dans un endroit commun que toute l'équipe de production peut consulter.
- *Scénarimage* (story-board) : représentation grossière de manière visuelle des écrans graphiques attendus du produit. Quelquefois des clips sonores sont aussi déterminés. Les médias utilisés dans chaque écran ainsi que les interactivités à attacher à ces médias sont identifiés. Une sous-étape de scénarimage est la *conception de l'interface*. Parfois, il existe diverses solutions d'interfaces pour un écran. Des modèles de ces interfaces peuvent être dessinés de façon plus détaillée pour mieux les percevoir et les comparer. Il faut tester ces modèles d'interfaces avec des utilisateurs potentiels, puis présenter les modèles et les résultats de tests au client pour avoir une approbation finale.
- *Modélisation et évaluation* : à travers l'exploitation des résultats des étapes précédentes, le concepteur essaye de construire le modèle du produit. Il est important que le modèle du produit soit le plus complet possible et le plus proche du produit final. Le concepteur peut utiliser n'importe quel outil de représentation ou de simulation pour construire le modèle. Un tel modèle peut permettre au concepteur de facilement évaluer et puis recommander des changements de produit. Le modèle peut donc être reconstruit plusieurs fois selon ses recommandations, afin qu'il soit de plus en plus proche du produit final. Cette étape permet de tester la réalité du produit pour diminuer ou éviter au minimum des changements de produit dans l'étape de production. On peut remarquer que jusqu'ici le processus de production s'effectue dans les étapes de conception qui permet de retourner d'une étape à une autre étape précédente pour modifier et améliorer l'image du produit. A partir de

l'étape suivante, s'il y a des retours arrière dans le processus de production, le coût sera très cher ou même impossible.

- *Production* du document : une fois qu'un modèle final du produit est disponible, il ne reste plus qu'à produire le document. Des médias déterminés pour composer le document seront créés ou collectés par des spécialistes de média comme des graphistes, des créateurs de vidéo, d'audio et d'animations, des concepteurs d'interface ou même des programmeurs. Puis ces médias sont assemblés selon le scénario donné dans les étapes de conception pour constituer le document.
- *Diffusion* : le document produit peut être enregistré sur CD-ROM/DVD pour le distribuer à l'utilisateur, ou dans des fichiers qui peuvent être publiés sur l'Internet.

Nous nous intéressons au cours de cette thèse à un environnement d'édition et présentation de documents multimédias. Cet environnement concerne les deux dernières étapes du processus de conception. Il aide l'auteur à mettre en scène un scénario issu des étapes de conception et donc à intégrer les médias élémentaires dans un document. Nous allons considérer dans les sections qui suivent quelles méthodes sont utilisées pour réaliser cette intégration de média. Cette étude reprend les 3 générations identifiées en section I.1 et présente pour chacune d'elles :

- Le principe de production,
- Les modèles de document utilisés,
- Des exemples de systèmes.

II.3.2 Première génération du système d'intégration multimédia

Dans cette première génération identifiée sous le nom de "production manuelle de document", le document multimédia est composé par le rédacteur à partir de l'ensemble de médias bruts. Le scénario peut être édité directement via un éditeur textuel, ou peut être généré par des environnements d'édition et présentation. Ces derniers fournissent plus de confort (par exemple, par la visualisation du scénario en cours de construction) comme *Macromedia Director*, *GRiNS* de *Oratrix* ou des prototypes comme *Limsee* et *Madeus* du projet *Opéra*.

Figure 1. Première génération du système d'intégration multimédia.

Il reste cependant des limitations dans le processus de production associé à cette génération d'outils. Premièrement, l'auteur doit chercher manuellement des médias adéquats à intégrer. Ce travail est très fastidieux, surtout qu'aujourd'hui

l'auteur dispose d'une importante ressource d'information sur Internet. Le problème principal pour faciliter cette recherche est qu'on n'a pas encore une façon standard et assez riche de décrire le contenu des médias. De plus, excepté pour le texte, les outils de recherche actuels n'ont pas encore la capacité d'effectuer correctement des recherches se basant sur le contenu. Par conséquent, bien que plus en plus de médias soient créés, peu sont réutilisés. De ce fait, les utilisateurs sont obligés de créer eux-mêmes des médias qui correspondent à leurs besoins. Cette solution est connue pour être non seulement très difficile mais aussi très coûteuse (voir la section II.2.2). Enfin, les produits créés sont difficiles à réutiliser pour réaliser d'autres produits.

Deuxièmement, les médias collectés sont des médias bruts qui sont très difficiles à intégrer pour créer un scénario sophistiqué (voir les exemples de *Karaoke* et de document technique dans la section II.2.3). De plus, les modèles déclaratifs d'intégration actuels comme SMIL permettent difficilement de réaliser de tels scénarios.

Pour surmonter ces limitations, nous proposons (voir la Figure 2) d'ajouter dans le processus un groupe d'opérations qui permettent d'analyser/de générer/d'éditer des descriptions du contenu des médias. Cela permet de créer une base de médias sémantiques à partir des médias bruts auxquels sont associés des métadonnées pour décrire la structure et la sémantique du contenu du média de la façon le plus standard possible pour une exploitation très large. L'utilisation de médias sémantiques au lieu de médias bruts permet de réaliser plus facilement des scénarios complexes dans lesquels des compositions fines entre des fragments de médias sont demandées.

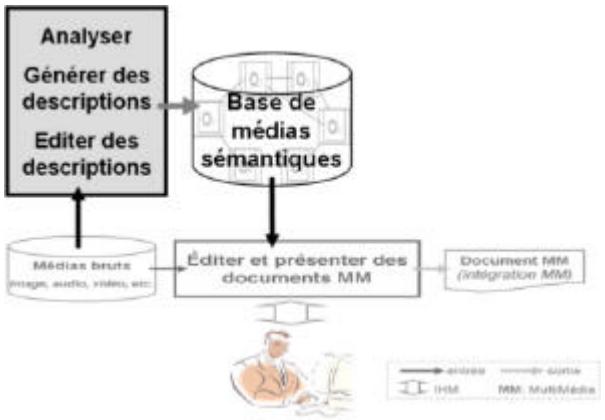

Figure 2. L'utilisation des médias sémantiques dans le processus de production de document multimédias de première génération.

Il y a encore une autre limitation importante des outils de cette génération : le document multimédia obtenu n'est pas capable d'adaptation. Il est souvent approprié à un seul type de client comme un processeur puissant, une bonne résolution de l'écran, un réseau haut débit, un langage, etc. La cause de cette limitation est le mélange des ressources d'information, structure de présentation et structure logique dans un même modèle. Cependant, certains travaux visent à améliorer cet aspect : intégration d'un opérateur *Switch* dans SMIL qui permet de choisir un sous-scénario en fonction du contexte de présentation ; ou encore le même principe est proposé mais de façon plus améliorée dans le modèle *ZYX* dans

lequel un média ou un groupe de médias peut être choisi dynamiquement pour adapter le contexte de présentation au lieu d'être prédéfini comme statiquement dans le modèle de SMIL. Néanmoins, ces améliorations ne permettent pas de répondre à tous les besoins liés à la problématique d'adaptation des documents multimédias [Jourdan et al. 01] [Ossenbruggen et al. 01] [Villard 02].

En effet, la variété des terminaux de présentation des documents multimédias est telle : ordinateurs de puissance différente, écrans de taille et de résolution différentes, connexions plus ou moins performantes, utilisateurs de langues différentes, etc., qu'il faut des opérateurs de plus haut niveau pour prendre en compte toutes les adaptations possibles. De plus, l'adaptation de cette variété devient de plus en plus critique avec la sortie des nouveaux matériels comme les PDA (*Personnel Digital Assistant*) ou le téléphone mobile de la troisième génération des communications mobiles (UMTS).

Enfin, la croissance de la production de média a pour conséquence le développement de bases de données multimédias énormes et suscite de nombreux travaux de recherche pour l'organisation, l'indexation et l'accès à de telles quantités de données multimédias.

Les outils de production de document multimédia doivent permettre de réutiliser au maximum les ressources existantes dans les bases de données multimédias et les documents créés doivent supporter divers types d'environnement et de ressource de l'utilisateur final. Il faut éviter de sauvegarder différentes versions d'un média (par exemple, un journal de la télévision avec des divers formats : *.avi*, *.mpeg*, *.mov* pour divers lecteurs chez l'utilisateur final; ou diverses résolutions pour différentes connexions) pour différents contextes d'utilisation, et éviter de créer manuellement un document spécifique pour chaque mode d'utilisation spécifique.

Dans le contexte ci-dessus, les systèmes de production multimédia de la première génération ne sont pas suffisamment capables d'adaptation. Il faut donc fournir des systèmes plus flexibles pour s'adapter aux nouveaux contextes et pour simplifier la tâche des concepteurs.

II.3.3 Deuxième génération du système d'intégration multimédia

Outre ce besoin d'adaptation au contexte que le processus de production unitaire de document multimédia n'assure pas, un autre besoin mal couvert par la première génération de systèmes concerne l'homogénéité des présentations pour des documents de même classe. La deuxième génération de système de production multimédia permet de répondre à ces besoins grâce à l'émergence des technologies de traitement de document du Web comme *XML* et les langages associés *XSL*, *XSLT*, *XPath*, *XQuery*, etc. (voir la Figure 3). La caractéristique principale de cette génération est sa capacité de production de *classes de document multimédia* (document abstrait) qui peuvent être utilisées pour générer au vol des présentations multimédias adaptées au contexte d'utilisation.

En effet, la production de document multimédia avec des outils de deuxième génération ne s'intéresse pas uniquement à la présentation du document multimédia, mais aussi à adapter largement le document à différentes sortes d'applications. Ils sont fondés sur la technique de transformation de document qui

sépare complètement l'information source de la présentation (voir la Figure 3). L'information ainsi séparée peut être organisée dans une structure logique plus riche, abstraite et neutre par rapport à la présentation. Cette structure est appelée le document abstrait ou le document métier. Les modèles *RST* (*Rhetorical Structure Theory*) [William et al. 89], Docbook [Docbook 01] et ATA [ATA 00] sont les types de documents abstraits. La présentation de documents est aussi plus souple grâce à cette technique, car elle est attachée aux types d'information au lieu de l'information elle-même. Cet attachement s'effectue par l'intermédiaire de feuilles de présentation. Ces dernières contiennent un ensemble de règles et de contraintes qu'un processeur de transformation peut utiliser pour transformer le document abstrait en document dont la présentation s'adapte au contexte de présentation spécifique. Les langages CSS, DSSSL [DSSSL:ISO] et XSLT [XSLT:W3C] sont des exemples de format de spécification de feuilles de présentation.

La Figure 3 présente un modèle idéal de production de documents multimédias de deuxième génération. Le processus est plus évolué que celui de la première génération. Le rédacteur édite d'abord au niveau abstrait des présentations logiques (organisations logiques des ressources) et présentations abstraites (feuilles de présentation) à partir des médias bruts dans une base. Les documents abstraits et présentations abstraites sont stockés dans une base qui supporte un moteur de recherche. Le moteur peut prendre des profils et des demandes d'un client dans un environnement hétérogène pour choisir à la fois un document et une présentation abstraite pertinente à la demande. Ensuite un processeur peut exécuter la transformation à partir de ce couple concernant un document et une présentation abstraite pour dynamiquement générer une présentation multimédia finale appropriée aux demandes du client. Le processeur de transformation/génération peut fournir une interface pour éditer les documents et les présentations abstraites. A ce niveau les éditions sont effectuées sur des présentations finales. Cela permet une meilleure perception que l'édition au niveau abstrait (voir la section **La transformation incrémentale**).

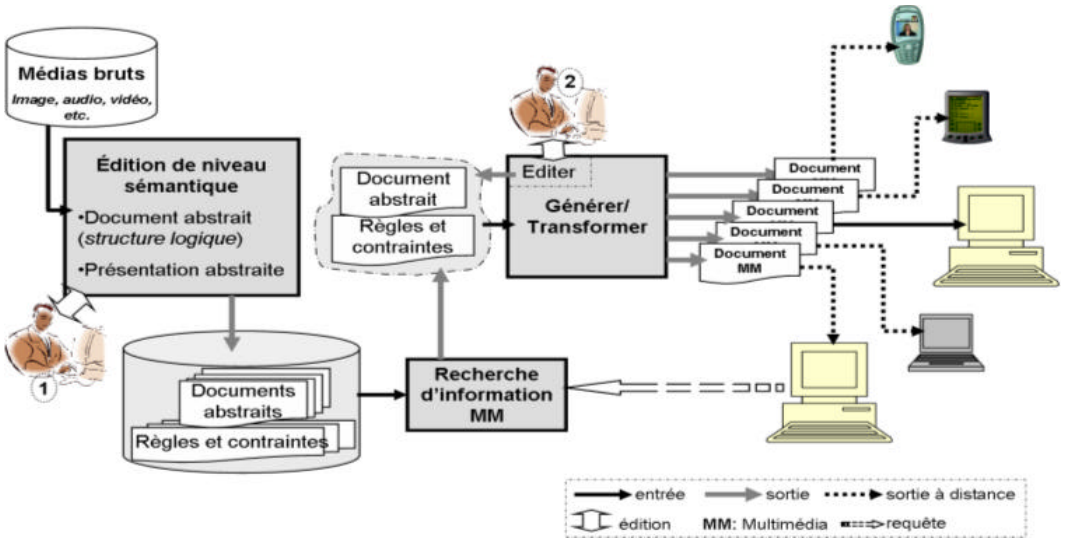

Figure 3. Modèle idéal de production multimédia de deuxième génération

La technique de transformation dans cette architecture idéale est héritée du succès de la transformation de documents textuels. Mais l'évolution de ces

techniques aux documents multimédias n'est pas si simple. Le travail expérimental de J. Ossenbruggen et al. [Ossenbruggen et al. 01] qui a développé un prototype d'un environnement de transformation du document multimédia, Cuypers, a déterminé que la transformation de document multimédia est beaucoup plus complexe que la transformation de documents textuels. En particulier, la transformation multimédia utilise différents médias et différentes présentations abstraites [Villard et al. 00], ses règles de formatage sont organisées de manière plus complètes (cinq niveaux de transformation dans Cuypers et trois niveaux dans l'architecture de [Jourdan et al. 01]), elle a besoin d'échange d'information entre les niveaux de formatage et enfin il est difficile de décrire les feuilles de transformation nécessaires avec les langages de style courants (CSS et XSLT) qui sont principalement dédiées pour la transformation de documents statiques et textuels [Ossenbruggen et al. 01]. Nous allons plus précisément considérer ci-dessous l'état existant dans cette génération de systèmes multimédias.

II.3.3.1 Systèmes existants

La deuxième génération de documents multimédias se focalise essentiellement sur les systèmes d'adaptation dont un état de l'art peut être trouvé dans [Jourdan et al. 01] et [Villard 02]. Nous présentons dans les sous-sections suivantes quelques systèmes des deux techniques d'adaptation les plus importantes du domaine : celle basée sur la notion d'*alternative* puis celle basée sur la notion de *transformation* [Villard 02]. Celles-ci correspondent aux deux niveaux différents d'adaptation : celui des *médias* et celui de *structure* globale de la présentation. Puis nous terminerons par un système qui combine ces deux techniques.

II.3.3.1.1 Adaptation à base d'alternatives

Cette technique s'appuie directement sur le modèle de présentation d'un document. Le modèle de présentation est amélioré par des attributs ou éléments alternatifs qui permettent de déterminer un média ou une partie du document selon le contexte d'utilisateur. Par exemple, l'attribut *alt* dans HTML permet de spécifier un média alternatif pour un élément HTML. L'élément *switch* dans le langage SMIL 2.0 permet quant à lui de spécifier des alternatives de fragments de document multimédia. Cependant l'alternative reste statique car elle est spécifiée au moment de l'édition du document, ce qui signifie que tous les contextes de présentation doivent avoir été prévus.

Dans [Boll et al. 99a] une évolution essentielle et importante dans cette technique est présentée : à chaque élément (atomique ou complexe) est associé un ensemble de métadonnées qui est exploité de manière dynamique par la stratégie d'adaptation dynamique. De plus, ce travail propose un modèle d'édition et publication des présentations multimédias adaptable dans laquelle l'édition des médias alternatifs est contrôlée de façon à assurer l'équivalence sémantique entre les médias. Alors, l'adaptation dynamique pendant la présentation est effectuée tout en offrant une garantie de cohérence sémantique du flot d'information. Ajouter suffisamment de métadonnées dans un élément média peut en plus permettre à ce média de s'auto adapter. Par exemple, dans [Vetro et al. 01], le système utilise des métadonnées associées à une vidéo pour ne transmettre que les objets intéressants. Ce système repose sur les schémas de description des informations audiovisuelles

définis dans MPEG-7. En particulier, l'outil de génération/transformation pourrait exploiter de telles métadonnées pour mieux choisir des contenus et ne pas être restreint aux traitements prévus.

On peur noter que cette technique est limitée au niveau de médias [Villard 02]. Une adaptation de plus haut niveau, au niveau de la structure, doit être ajoutée et ne peut être réalisée qu'avec la technique de transformation.

II.3.3.1.2 Adaptation à base de transformations

Les techniques de transformation sont utilisées dans de nombreux traitements de documents comme la production et l'exploitation de documents statiques. De plus, depuis le déploiement de XML, de nouvelles perspectives sont explorées avec des langages de transformation comme XSLT (*XSL Transformation*). En effet, de nombreuses informations ne sont plus attachées directement à la structure de présentation, mais plutôt à la structure logique (typiquement sous forme XML) qui est neutre par rapport à la présentation. A partir de ces formats neutres, la transformation est toujours appliquée pour traiter les informations. Par exemple, pour réutiliser des informations préexistantes il faut d'abord les transformer en format du contexte de l'utilisation (XML vers HTML, SVG ou SMIL); l'exportation d'un document vers un autre format a aussi besoin de transformation ; ou de même la recherche d'informations a besoin de transformation pour structurer des résultats de la recherche (XML vers HTML, XML vers SVG ou bien XML vers SMIL). Dans les sous-sections suivantes nous présentons deux systèmes d'adaptation de documents multimédias qui sont basés sur des techniques de transformation.

Le système Cuypers

[Ossenbruggen et al. 01] a positionné le développement du multimédia sur le Web par rapport aux trois générations du Web. En fait, seuls les documents textuels de type HTML sont en train de rapidement évoluer vers la troisième génération du Web ; alors que les documents multimédias sont encore du niveau de la première génération et récemment commencent à évoluer vers la deuxième génération. La cause de ce déphasage est la différence fondamentale entre le contenu multimédia et le contenu purement textuel qui implique des différences dans la modélisation, le formatage et l'expression des transformations. Ainsi, si les besoins sont les mêmes, les techniques pour y répondre sont différentes.

Le prototype de production de présentations multimédias Cuypers répond à ces besoins sous forme d'une interface entre une base de données multimédias semi structurées et le serveur Web (Figure 4). Cuypers prend en compte les expériences des premiers prototypes (par exemple, le travail de Bailey et al. [Rutledge et al. 00]), qui ont montré que la transformation directe d'un couple (structure logique, présentation abstraite) vers une présentation multimédia concrète est très difficile, car la différence entre ces deux niveaux est trop grande. Au lieu de cela, Cuypers adopte une approche incrémentale, qui décompose la transformation totale en cinq étapes plus petites, chacune correspondant à un niveau d'abstraction différent : du niveau sémantique (relations abstraites entre les éléments dépendant de l'application, par exemple "éléments en séquence") au niveau physique (exemple HTML, SMIL). Les niveaux intermédiaires transforment les relations abstraites en

structures abstraites de présentations spatiales, temporelles et liens, puis en contraintes de présentations qualitatives et finalement en contraintes quantitatives conduisant au formatage final.

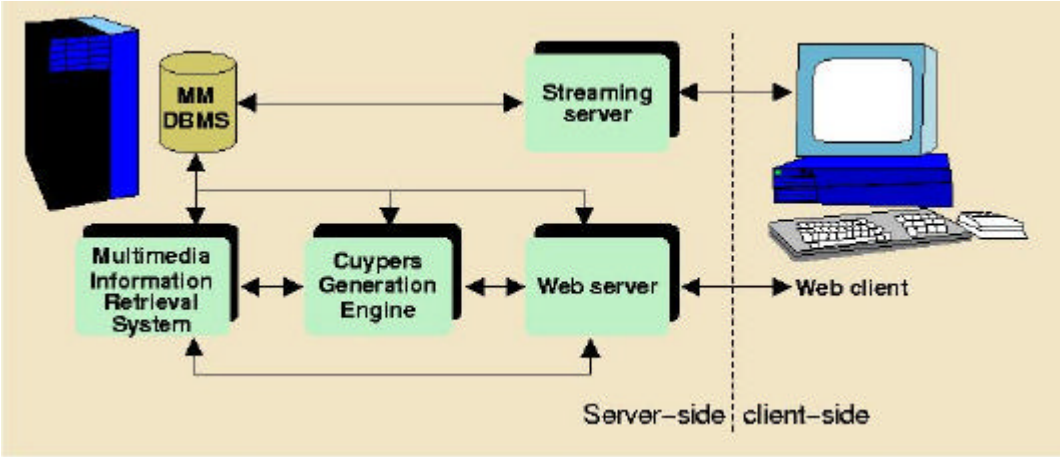

Figure 4. L'environnement du moteur de génération *Cuypers*

En résumé, Cuypers a prouvé qu'il était possible de générer dynamiquement des présentations multimédias adaptables à la fois à divers types de clients Web mais aussi à divers formats de présentation finale. En plus, son approche par transformation successive est très intéressante. Cependant, la première version du prototype est limitée à un sous ensemble restreint de la structure de relation sémantique (*Rhetorical Structure Theory - RST*). De plus, la génération des présentations RST n'est pas dynamique. Les présentations RST sont simplement classifiées dans la base (MM DBMS, voir la Figure 4) de façon à les récupérer facilement par le serveur de recherche. Il est à noter que certaines de ces limites font l'objet de propositions de solutions dans [Ossenbruggen et al. 02].

Une architecture générique pour construction automatique de présentation de multimédia

La même architecture de base a été proposée dans [Jourdan et al. 01] par la génération automatique de présentations multimédias. Mais, plutôt que d'enchaîner des transformations successives, une étape d'analyse des paramètres dynamiques sélectionne les feuilles de transformation (TS) et les contraintes (CS), puis une étape de sélection de contenu permet de produire des fragments XML de présentation (par exemple des nœuds SMIL). Enfin l'étape de transformation proprement dite s'applique sur les fragments en appliquant les feuilles de style sélectionnées à la première étape (voir la Figure 5).

L'originalité de l'approche réside dans l'utilisation d'un résolveur de contraintes pour la phase de sélection de contenu, ces contraintes étant dynamiquement sélectionnées dans la phase d'analyse de paramètres.

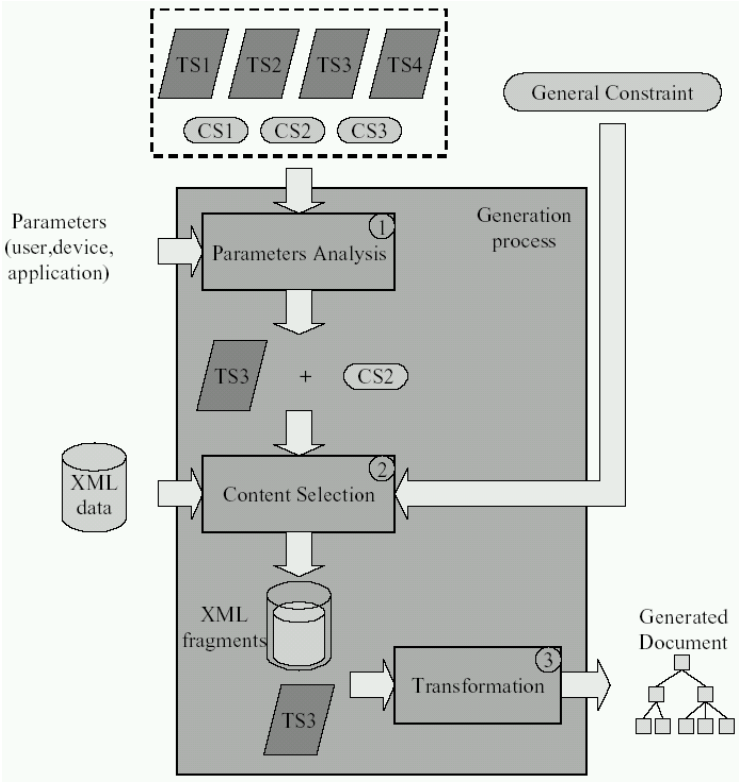

Figure 5. Architecture du système [Jourdan et al. 01].

L'intérêt majeur est de non seulement pouvoir s'adapter à la capacité d'affichage de l'appareil de l'utilisateur, mais aussi à la durée préférée de l'utilisateur. Mais le système reste encore très spécifique. L'utilisateur est limité au un ensemble de présentations abstraites (TS) prédéfinis ce qui sont des feuilles de transformation. On ne peut pas décrire un scénario complexe par des règles dans les feuilles de style. De plus l'approche par feuilles de transformation est aussi très sensible au domaine d'application. Par conséquent, un modèle abstrait pour la présentation multimédia reste une tâche très difficile à la base encore de nombreux travaux de recherche du domaine multimédia.

La transformation incrémentale

Les deux travaux présentés ci-dessus visent seulement l'étape de génération. L'auteur se situe quant à lui dans une étape en amont dans la chaîne du système (Figure 3) : C'est l'étape d'édition sémantique de la Figure 3 avec l'auteur numéroté 1. Concernant cette étape, le travail de [Villard 02] a proposé un système d'édition de transformation et de présentation des documents multimédias basé sur une édition directe, interactive et incrémentale. Ce système a donc non seulement la capacité de transformer/générer automatiquement des présentations multimédias adaptables mais aussi permet à l'auteur de participer à la tâche de transformation/génération (cf. la Figure 6). L'auteur se situe alors à la position numérotée 2 dans la chaîne générale de la deuxième génération du système multimédia (Figure 3). L'auteur peut éditer la présentation abstraite via une présentation cible, ce qui est plus visible et alors plus facile que l'édition directe au niveau abstrait (comme les travaux présentés au-dessus). Dans le sens inverse, une modification dans la présentation abstraite est toute suite affichée dans la

présentation cible. L'auteur peut donc voir les résultats de ses éditions de façon instantanée. Dans les deux cas, une feuille de la transformation est générée ou mise à jour automatiquement à chaque modification que ce soit du côté de la présentation cible ou du côté de la présentation abstraite.

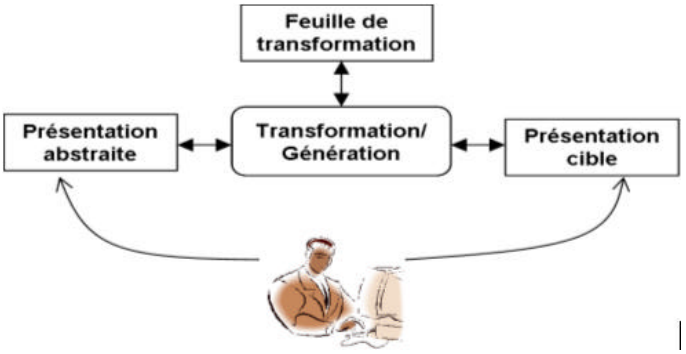

Figure 6. Architecture générale du système d'édition dans [Villard 02].

Le système est basé sur un processeur de transformation incrémentale dont l'objectif est de mettre à jour le document résultat de la transformation à une modification soit du document source, soit de la spécification de la transformation. Pour que cette mise à jour soit efficace, le processeur incrémental ne doit réexécuter que les fragments de la transformation qui produisent un résultat différent [Villard 02].

Modèle d'adaptation à travers de multiples plates-formes

Le degré d'adaptation dynamique est encore plus important dans les cas où les paramètres de la plate-forme ne sont connus qu'au moment même de la présentation, par exemple, lorsque la disponibilité de débit du réseau ou de certaines ressources de l'utilisateur varie à chaque moment d'utilisation.

[Ossenbruggen et al. 99] a proposé une solution pour répondre à ce problème. Il a choisi alors une approche hybride de présentation adaptable et adaptative. L'approche adaptable repose sur une présentation abstraite qui peut être transformée/générée vers différentes présentations cibles sur de multiples plates-formes potentielles. L'approche adaptative est basée sur des contenus alternatifs qui contiennent suffisamment d'information ainsi qu'un noyau d'exécution d'adaptation pour permettre une adaptation dynamique. Cependant, la capacité du système est encore limitée, parce que l'alternative reste statique : un média choisi au début de la présentation doit être joué dans toute la présentation. Si la ressource ou le débit du réseau sont diminués, il ne peut pas changer automatiquement et dynamiquement un autre média pour préserver la qualité de la présentation. De plus, il ne fournit pas l'assurance d'une équivalence sémantique entre les contenus alternatifs.

II.3.3.2 Synthèse

Les travaux rapportés ci-dessus montrent la faisabilité de la production de document multimédia adaptable par génération. Cependant la qualité des présentations multimédias obtenues est encore insuffisante car les scénarios générés sont de simples intégrations des médias, présentés en séquence ou en

parallèle. Ces présentations ne peuvent se comparer avec la sophistication des programmes multimédias de première génération. C'est pourquoi la proposition de [Boll et al. 99a] doit être considérée avec intérêt, car elle ouvre la voie à une réelle adaptation dynamique grâce à l'utilisation de métadonnées.

Nous proposons donc d'intégrer dans la chaîne de la Figure 3 un maillon de description des éléments média. Ce maillon va se charger de produire une base des médias sémantiques qui seront utilisés pour réaliser des transformations plus sophistiquées et donc des adaptations plus dynamiques et plus cohérentes.

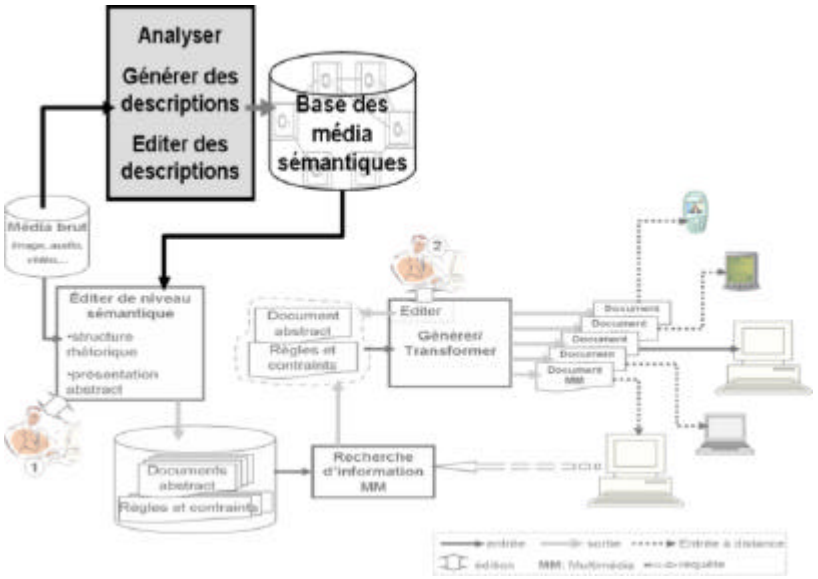

Figure 7. Maillon de description des éléments media dans la deuxième génération du multimédia.

Cependant, La sémantique de la base de médias sémantiques est encore locale, c'est-à-dire elle ne peut être comprise que par le système dans lequel elle est créée. Pour offrir plus d'interopérabilité, il est nécessaire de disposer d'une modélisation de plus haut niveau comme on va le voir dans la section suivante.

II.3.4 Troisième génération de systèmes d'intégration multimédia

Nous venons de présenter les deux premières générations de systèmes d'intégration multimédia. Nous allons maintenant présenter la plus novatrice : le multimédia sémantique. Les recherches liées à cette génération visent à utiliser des métadonnées, qui reflètent plus directement la sémantique des contenus d'information. De cette manière, les programmes informatiques pourront traiter automatiquement des ressources d'information. En bref, l'idée est de faire davantage de travail sur les ressources pour permettre ensuite aux fonctions de traitement de réaliser le plus possible de tâches. C'est l'idée principale de cette génération du multimédia connue sous le nom de Web sémantique [W3C SW01] [Berners-Lee et al. 01].

II.3.4.1 Nouvelle génération de système multimédia

L'édition multimédia devient de plus en plus familière à l'utilisateur, il est reconnu que, malgré l'existence d'outils d'édition avancés comme GRiNS et Director, et le

pouvoir d'expression du format SMIL, l'édition et la présentation de documents multimédias souffrent encore de certaines limites [Rutledge et al. 01a] :

- ♦ Les outils existants offrent un support limité pour intégrer des fragments de média : les médias sont considérés comme des éléments de base sans possibilité d'un accès plus fin.
- ♦ L'auteur est peu aidé dans sa recherche des médias à intégrer et cette phase reste encore très coûteuse.
- ♦ Au final, il reste toujours à savoir comment le processus d'édition multimédia peut être automatisé pour minimiser plus possible les tâches de l'auteur.

Un nouveau cycle de recherche a donc commencé pour une nouvelle génération de l'édition et présentation du document multimédia dans lequel le processus d'édition multimédia est le plus possible automatisé pour diminuer des efforts de l'auteur. L'idée est que l'auteur puisse entrer des requêtes, puis recevoir des médias appropriés à l'intégration. Un processus encore plus avancé serait celui où l'auteur peut décrire une présentation multimédia que il souhaite obtenir, puis non seulement tous les medias pertinents sont automatiquement amenés, mais ils sont aussi automatiquement intégrés et structurés dans, par exemple, une présentation SMIL.

II.3.4.2 Le Web sémantique

La consultation des informations sur le Web est encore très limitée. L'utilisateur doit prendre en charge beaucoup de traitements qui devraient être effectués par l'ordinateur. Les outils actuels du Web ne s'occupent presque qu'uniquement d'afficher des informations. Ceci est confirmé par les propos de B. Gates qui dans un message envoyé aux développeurs et professionnels [Gates 01] a précisé « … le navigateur joue le rôle de terminal muet … . Pire, les pages du Web sont simplement une "image" des données, pas les données elles-mêmes … ». A l'opposé «le Web Sémantique est une extension du Web actuel dans lequel on donne à l'information une signification bien définie permettant aux ordinateurs et aux personnes de mieux travailler en coopération» [Berners-Lee et al. 01]. En fait, dans cet article, Berners-Lee et al. ont imaginé une application du futur Web sémantique où l'utilisateur peut prendre un rendez-vous avec un médecin de façon très rapide, facile, précise et performante. Le choix du rendez-vous est fait totalement automatiquement selon les demandes de l'utilisateur. A partir de la demande de rendez-vous de l'utilisateur, l'outil va chercher dans toutes les pages Web des cabinets des médecins dans les régions préférées de l'utilisateur, puis filtre les cabinets selon qu'ils acceptent ou non l'assurance de l'utilisateur avant de comparer les agendas de ces cabinets avec celui de l'utilisateur pour trouver des dates disponibles. Il peut consulter ensuite des informations sur la page Web du trafic routier pour à la fois trouver les accès les plus rapides au cabinet et éviter les routes embouteillées, etc. L'utilisateur peut alors consulter les propositions pour en choisir une ou relancer une autre recherche.

Cet exemple illustre bien les possibilités du Web sémantique. Cependant, il reste limité à des applications basées sur un contenu textuel. Or les documents multimédia qui intègrent le texte, l'image, l'audio et la vidéo dans des documents structurés complexes, dans lesquels des relations temporelles, spatiales, structurales

et sémantiques existent entre des composants posent des problèmes d'indexation, d'archivage et, de recherche infiniment plus complexes que la découverte de ressources des documents textuels. Les sections suivantes envisagent plus en détail les solutions existantes.

II.3.4.3 Les solutions existantes

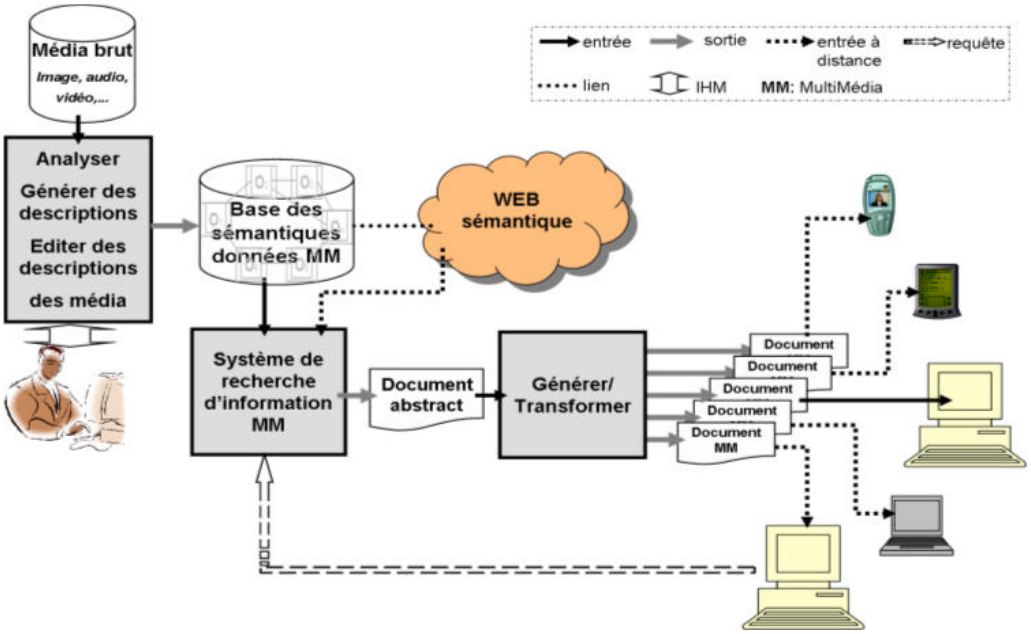

Figure 8. Troisième génération du multimédia.

Nous nous intéressons à présenter au schéma général de la troisième génération de multimédia comme décrit Figure 8. Dans cette architecture l'utilisateur peut simplement envoyer une demande pour une présentation multimédia. Un composant du système reçoit cette demande, les décompose et l'analyse pour trouver des contenus pertinents et des relations entre eux. Ces informations permettront à l'outil de former une requête suffisamment détaillée, c'est-à-dire, qui contienne non seulement des informations pour trouver chaque média élémentaire adéquat, mais aussi des contraintes qui permettent à ces médias élémentaires d'être composés de manière satisfaisante. Ensuite, au niveau de la base de données sémantique, un outil de gestion de la base va donner une présentation abstraite correspondant aux demandes de l'utilisateur. A partir de la présentation abstraite trouvée, le système peut transformer et générer la présentation finale adaptée à l'environnement de l'utilisateur comme c'est le cas pour les systèmes de deuxième génération.

Le cœur de l'intelligence du système est la sémantisation des informations dans le système. Malheureusement, la plupart des médias créés et distribués actuellement ne fournissent aucune sémantique au système, c'est en particulier le cas des pages Web ou des documents basés sur technologie XML. Le système doit alors ajouter un outil d'analyse et de génération charger de la description sémantique du contenu de ces médias bruts.

Le schéma fourni dans la Figure 8 décrit seulement le cas idéal d'un système de la troisième génération. Il n'existe pas actuellement un tel système complet. Mais,

des recherches en cours peuvent fournir des solutions partielles. Nous donnons dans cette section une vision très globale de ces travaux. L'analyse détaillée de chaque travail sera donnée dans le chapitre suivant.

II.3.4.3.1 Description sémantique et média structuré

De nombreux standards permettent maintenant de décrire le contenu des médias. D'abord les standards de description du contenu de média comme Dublin Core, qui fournit un ensemble de treize éléments standard extensibles pour décrire le contenu de média ; RDF (*Resource Description Framework*) est un standard de W3C pour description des ressources ; MPEG7 est un standard de MPEG pour décrire des ressources audiovisuelles ; et encore beaucoup d'autres solutions moins standard comme INDECS [Rust et al. 00], IMS [IMS], VRA Core [VRA].

En parallèle, l'augmentation des formats multimédias structurés constitue aussi un avantage important pour la nouvelle génération. Les média structurés permettent d'identifier et de localiser à chaque fragment élémentaire, ainsi d'appliquer l'annotation sémantique au contenu de média plus profond jusqu'à ces fragments élémentaires. Cela permet de concevoir des outils beaucoup plus intelligents qui peuvent trouver des fragments de média appropriés à l'intégration. Alors le système peut automatiquement et indépendamment utiliser des fragments de média lors de la réalisation d'une présentation. Les travaux typiques dans ce sens sont les nouveaux standards du W3C : SMIL pour l'intégration multimédia sur le Web, et SVG (Scalable Vector Graphics) pour encoder des graphiques en XML ; et le standard du groupe MPEG : MPEG-7, qui fournit une façon standardisée pour indexer le contenu des média basiques comme l'image, l'audio et la vidéo. Une évolution importante dans le codage des audio et des vidéos numériques est le standard MPEG-4 basé sur une technologie orientée objet. Elle permet de mieux adapter des flux média aux conditions de transport, et d'intégrer plusieurs types de média et des interactions dans une scène (structurarion d'une scène). Le couplage de MPEG-4 et MPEG-7 va former un format de codage d'audio et vidéo. Ce type de format est d'aussi haut niveau que l'approche déclarative (HTML, SMIL, SVG, etc.) qui supporte facilement des outils de recherche. Mais en plus il conserve au final un codage binaire qui est beaucoup plus performant.

II.3.4.3.2 Modèles interopérables

Comme nous avons pu le voir dans la section ci-dessus, bien qu'il y ait beaucoup de solutions pour ajouter de la sémantique aux ressources, cela reste encore un défi actuel. En fait, un système qui peut traiter tous les modèles de sémantisation est complexe, voire impossible. En plus, la sémantique est très vaste, il y a beaucoup de différences entre les domaines, les connaissances, les cultures, etc. Tout cela crée des complexités qu'aucun système global ne peut résoudre. Heureusement, des recherches en cours pour des modèles d'interopérabilité peuvent nous fournir la solution. Ils permettent aux systèmes de s'appuyer non seulement sur une base de média locale, mais aussi sur un réseau interopérable de ressources comme le Web sémantique (Figure 8).

L'interopérabilité est un protocole commun qui aide des systèmes différents à se reconnaître et qui gère par les systèmes les problèmes de format, de connaissance, ou de procédure inconnus. Par exemple, actuellement la technologie multimédia dans le monde industriel utilise principalement des formats de médias propriétaires. *RealPlayer* de *RealNetwork*, *Windows media player* de *Microsoft* et *Quick time* de *Apple*, chacun occupant une part importante du marché multimédia [Mariano 01] : 28.8 millions de personnes à la maison et 15.5 millions au bureau utilisent RealPlayer ; Windows Media a de 13 millions de consommateurs à la maison et 8.8 millions au bureau ; QuickTime a attiré 8.2 millions de consommateurs à la maison et 5.3 millions au bureau. Dans ce contexte, si des fournisseurs de contenu multimédia veulent distribuer largement leurs produits, ils doivent donc les diffuser sous au moins ces trois formats les plus populaires. Du côté des consommateurs, cela signifie qu'ils doivent être capables de lire ces 3 formats s'ils veulent voir la plupart des média sur le Web. A l'opposé, le groupe ISMA (*Internet Streaming Media Alliance*) fournit la première version de son standard ouvert pour diffuser la vidéo (ISMA 1.0) basé sur MPEG-4. Cette spécification propose un système interopérable qui permet à l'utilisateur d'installer une seule fois un lecteur multimédia, et aux fournisseurs d'enregistrer une seule fois leurs contenus média [Mariano 01].

Dans le même sens, le projet MÆNAD (*Multimedia Access for Enterprises across Network And Domains*) a démarré à partir du constat qu'il y a plusieurs développements de standards de métadonnées comme Dublin Core, MPEG-7, INDECS, IMS, VRA Core, etc. Chacun de ces standards est dédié soit à un type de média : Dublin Core fournit de l'interopérabilité sémantique pour les documents textuels, MPEG-7 fournit la même chose pour les documents audiovisuels ; soit à un aspect spécial comme INDECS est un standard de métadonnées pour décrire des informations légales dans le commerce électronique, ou IMS pour des ressources d'apprentissage en ligne. Les objectifs du projet MÆNAD sont de développer des outils pouvant fournir des solutions aux problèmes de la recherche de ressources multimédias. Les outils vont améliorer la capacité de l'interopérabilité parmi les standards de tous les deux niveaux : la sémantique métadonnées et le format de média.

Les exemples considérés au-dessus décrivent seulement des cas simples de la création et de la consommation de média dans le domaine multimédia. En fait, la chaîne de la création du contenu, puis la livraison du contenu et enfin la consommation du contenu est beaucoup plus complexe et diversifiée. Elle dépend non seulement du format encodé de média et le lecteur multimédia mais aussi plusieurs autres caractéristiques liées au chaque environnement comme le modèle, des règles, des procédures, des buts, le réseau de connexion et la capacité de calcul de l'environnement. Pour un tel niveau global d'interopérabilité, une nouvelle recherche du groupe MPEG appelée MPEG-21 (Multimedia Framework) a démarré en 2001 avec pour objectif « l'accès universel au multimédia». MPEG-21 promet un modèle multimédia commun qui permettra une coopération facile entre des infrastructures différentes. On peut noter que non seulement la différence entre les systèmes empêche l'interopérabilité mais aussi la protection des ressources (problème de la propriété intellectuelle) produit actuellement aussi des obstacles. Les parties : IPMP (Intellectual Property Management and Protection), RDD

(Rights Data Dictionary) et REL (Rights Expression Language) intégrées dans le travail de MPEG-21 sont dédiées à ce problème [Koemen 01].

On peut trouver le même contexte mais dans le domaine de la télévision, le groupe TV-Anytime Forum formé en 1999, qui est en train de développer une spécification ouverte pour un système interopérable et intégré qui permettra aux diffuseurs et autres producteurs de ressources électroniques d'utiliser des ressources d'autres origines.

Enfin, RDF (*Resource Description Framework*) est un modèle très abstrait et neutre pour décrire des ressources d'information de manière sémantique que des outils peuvent non seulement lire, mais aussi comprendre. Ce travail constitue le fondement du Web sémantique, de nombreux standards sont construits, le plus important entre eux étant DAML+OIL. RDF a une capacité de déduction comparable avec les systèmes IA (*Intelligence Artificiel*), mais sa puissance n'est pas limitée à une base locale de connaissance, mais elle peut être élargie infiniment grâce à sa capacité d'interopérabilité entre des ressources [Berners-Lee et al. 01].

Un fait avéré est que tous ces travaux utilisent la technologie XML pour encoder leur modèle. En fait, XML avec son caractère neutre vis-à-vis des plates-formes est la base de tous les systèmes d'interopérabilité.

II.3.4.4 Synthèse

En résumé, les recherches actuelles visent à développer une infrastructure sur laquelle une nouvelle génération de système multimédia pourra être construite. Leurs fondements sont des standards de métadonnées pour décrire des ressources et des métamodèles qui ouvrent la voie à l'interopérabilité entre des bases de données multimédias. Dans ces nouveaux systèmes, le producteur doit créer des ressources et leurs relations décrites au niveau sémantique au lieu de créer directement des présentations finales. L'idée est donc que l'auteur ne spécifie plus en détail la présentation qu'il veut avoir, afin que celle-ci puisse être automatiquement générée par outil.

Cependant, les applications existantes sont encore trop parcellaires et indépendantes, par exemple :

- SMIL permet seulement d'intégrer des médias dans une présentation multimédia synchronisée.
- MÆNAD permet seulement de découvrir des ressources multimédia. Bien qu'il permette d'entrer des requêtes compliquées comme «*trouvez l'extrait vidéo du troisième plan de la cinquième scène dans le quel apparaît la vase rouge en haut et à gauche*», les réponses sont simplement des médias ou des fragments de média.
- ISMA basé sur MPEG-4 résout simplement le problème de l'intégration des formats différents de la vidéo.

En fait, le processus de production de document multimédia a besoin d'enchaîner des travaux de divers aspects de traitements de document pour avoir une chaîne plus complète à partir de la collecte de médias puis la structuration des informations et leur composition et finalement l'adaptation de la présentation résultant au contexte utilisateur. Par exemple, ISMA peut être intégré dans les

applications de MÆNAD et de SMIL pour assurer la transparence de ces applications vis-à-vis des différents formats de média ; les réponses de MÆNAD seront un ensemble de médias qui soit s'intégrer directement en format SMIL ou soit être dans un format intermédiaire qui conserve les relations entre média de façon à permettre leur transformation en SMIL (par exemple par une transformation XSLT).

II.4 Synthèse

Nous avons présenté dans ce chapitre une vue globale du concept de document multimédia dans laquelle nous avons également précisé que l'approche qui répond le mieux aux attentes de cette thèse.

Nous avons vu par ailleurs que le concept de production de document multimédia a évolué très rapidement. À partir du modèle d'intégration des médias qui ne permet de produire que des présentations finales, des extensions ont été définies pour produire des présentations multimédias plus abstraites qui permettent de générer au vol des présentations finales appropriées à chaque contexte d'utilisation. Enfin le multimédia sémantique, qui est en cours de définition, promet encore plus de confort dans le processus de production et de consultation de documents multimédias. Ces évolutions se limitent encore bien souvent à considérer les médias comme des objets atomiques. De ce fait, la production de documents sophistiqués qui demande des capacités de composition fine à l'intérieur des médias n'est pas possible. De plus, l'explosion des bases de données multimédias fait apparaître des nouveaux besoins en annotation et en indexation de ressources pour pouvoir gérer celles-ci au mieux. Dans ce contexte, les objets médias ne doivent plus être considérés comme atomiques, mais au contraire comme des objets structurés contenant même des métadonnées sémantiques. Grâce à cela, un modèle de composition de document multimédia pourra raffiner le processus de la composition.

Les problèmes qui restent ouverts sont liés à la représentation de la structure interne des médias pour pouvoir l'utiliser après dans la production de document multimédia.

Nous consacrons les deux chapitres suivants à une analyse plus fine d'une part des modèles de description multimédia, et d'autre part des applications multimédias. A partir de cet état de l'art, nous serons en mesure de présenter notre proposition de modélisation et de réalisation logicielle qui contribue à l'émergence de cette troisième génération de système de production multimédia.

Chapitre III. Modélisation de multimédia

Ce chapitre est consacré à l'étude des modèles multimédias selon trois niveaux, à partir de modèles du contenu de média individuel jusqu'aux modèles de l'intégration et de la synchronisation multimédia. Nous étudions à chaque niveau, l'état actuel des modèles, leurs capacités et leurs insuffisances en vue de constituer un modèle global d'intégration de tous les niveaux.

III.1 Introduction

Le besoin d'un accès plus précis à la structure d'un objet multimédia nécessite l'intégration d'une chaîne plus complète de traitement du contenu dans le processus de production d'un document multimédia. Généralement, une chaîne complète d'une application multimédia est divisée en trois étapes (voir la Figure 9). En entrée de cette chaîne, les médias sont analysés pour pouvoir extraire automatiquement et/ou manuellement des informations pertinentes, puis ces informations sont représentées sous un format prédéfini pour pouvoir être largement et efficacement utilisées dans des applications et traitements de média.

Figure 9. Chaîne d'application multimédia

L'analyse et la description du contenu d'un média sont maintenant utilisées dans des applications de gestion et d'indexation d'informations, par exemple, des bases de données multimédias. Cela signifie que des médias et même des fragments de médias peuvent être trouvés facilement et efficacement dans la base de données multimédias à partir de requêtes portant sur le contenu des médias eux-mêmes. Bien qu'ayant un intérêt général, l'analyse et la description du contenu multimédia sont jusqu'à présent principalement employées dans le domaine de la recherche d'information. Pourtant, la composition de document multimédia à partir de médias élémentaires peut être aussi considérée comme une application multimédia dans laquelle des médias ou des fragments de média appropriés ont besoin d'être récupérés pour composer une présentation multimédia. Néanmoins, l'utilisation d'application d'analyse et de description du contenu est encore très limitée dans le domaine de la production de document multimédia. Ce type d'applications traite les médias comme des boîtes noires ce qui rend très difficile des compositions fines. Il faut noter cependant que l'évolution récente de la technologie de

l'indexation basée sur le contenu rend aujourd'hui disponibles des médias avec des descriptions de leur contenu. Ceci permet plus facilement d'avoir accès à certaines portions d'un objet multimédia. Cette évolution va donc le sens des besoins des applications de document multimédia mais n'est pas encore suffisante. En particulier, la granularité d'accès aux médias n'est pas satisfaisante. D'autre part, les descriptions de contenu utilisées pour l'indexation ne couvrent pas les besoins de la composition multimédia comme la structure logique ou la structure temporelle. La connaissance de ces structures est importante pour éviter de manipuler les portions de contenu de façon absolue, par exemple, au lieu d'identifier simplement un extrait de vidéo par ses numéros d'image de début et de fin, il serait plus intéressant d'en connaître sa position dans la structure temporelle : le quatrième plan de la deuxième scène. La description du contenu des médias doit donc non seulement fournir des facilités pour la composition fine de document multimédia, mais aussi permettre de produire des documents plus structurés que ceux composés par spécification absolue.

Un simple regroupement des modules existants d'analyse, de description et de traitement multimédia permet-il de répondre à ces besoins ? Quels éléments sont manquants pour l'intégration d'une chaîne complète de traitement des médias dans le processus de production de document multimédia ? Nous allons envisager ces questions dans ce chapitre. L'étude est effectuée selon les trois maillons de la chaîne qui correspondent aux trois niveaux du domaine multimédia : l'analyse, la description et la composition.

III.2 Étude de l'analyse du contenu de média

L'analyse du contenu des médias est un domaine très développé dans les sciences informatiques. On la retrouve donc beaucoup dans la littérature de recherche, ainsi que dans des applications concrètes. En effet, l'analyse du contenu d'une information est une tâche indispensable avant son traitement pour effectuer et améliorer les performances d'un processus de traitement des informations (voir la Figure 9). Dans cette thèse, nous nous focalisons sur les travaux d'analyse de média de nature visuelle ou sonore, même si quelques travaux existent déjà sur d'autres médias liés aux autres sens (toucher, odorat, etc.).

On peut identifier dans un média plusieurs niveaux de structure. Actuellement la structure la plus accessible d'un média correspond au niveau de manipulation des outils de capture ou de production, qui exploitent des caractéristiques bas niveau du type : largeur, hauteur, pixels, couleur d'une image, ou bien une suite d'image d'une vidéo, etc.

D'autres types de structures peuvent être identifiés comme ceux nécessaires pour la restructuration de média et qui sont encodés dans les formats comme MPEG1/2/4, JPEG 2000, etc. (restructuration progressive, adaptation de la qualité et fonction des ressources, etc.).

Enfin les structures plus abstraites, liées à la sémantique portée par le média, sont encore peu spécifiées comme on l'a vu dans le chapitre précédent, ce sont celles qui nous intéressent car elles sont nécessaires aux applications multimédias. Les sections qui suivent décrivent les travaux existants qui permettent, par

différentes techniques d'analyse, d'extraire ces informations tout d'abord pour les médias visuels, puis pour les médias sonores.

III.2.1 Analyse des informations visuelles

Les informations visuelles agissent sur la perception visuelle avec leurs caractéristiques de *couleur*, *luminosité*, *texture*, de *forme* et de *position*. La base de l'analyse des informations visuelles est donc le traitement de ces différents éléments et dépend du type de contenu et de sa nature dynamique ou statique.

III.2.1.1 Médias graphiques et textuels

Parmi les médias visuels il est possible de distinguer les *graphiques* et les *textes*.

Les graphiques comme les photos, les images animées et les vidéos sont des organisations d'éléments spatiaux basiques comme les points et les régions. Chaque élément possède un ensemble d'informations bas niveau qui le caractérisent comme ses coordonnées, sa forme, sa couleur et sa texture. L'élément peut contenir des attributs sémantiques qui spécifient le sens de l'élément. Actuellement, les formats de base (MPEG, JPEG, GIF, etc.) utilisés pour coder ces médias ne permettent pas de coder ces éléments et ces attributs. De ce fait les applications multimédias doivent faire une analyse pour identifier ces éléments dans le contenu de chaque média. Une analyse consiste donc à extraire les informations caractéristiques des éléments (voir la Figure 10). Actuellement, dans des cas spécifiques ou lorsque les médias sont bien codés, cette extraction sur les médias graphiques peut donner des résultats précis. Cependant, cette extraction est encore limitée aux informations spatiales physiques de bas niveau comme la couleur, la texture et la forme. L'extraction des informations sémantiques reste encore un obstacle à franchir dans le domaine de l'analyse d'images graphiques. Par exemple, la Figure 10 représente le résultat de l'analyse d'une image effectué semi-automatiquement. Elle a permis d'identifier différentes régions et même certaines relations spatiales de base (à travers une décomposition hiérarchique), mais elle ne peut pas déduire la signification sémantique des régions et des relations extraites de l'image. Les informations sémantiques codées sous forme de texte et associées aux fragments de la structure extraite doivent donc être ajoutées manuellement comme proposé dans SIGMA [Matsuyama et al. 90], un système de segmentation d'images aériennes qui utilise une base de connaissance codée manuellement. L'extraction sémantique peut être automatisée par l'utilisation des techniques de reconnaissance de forme comme [Mikolajczyk et al. 01] qui permet de détecter automatiquement des visages dans une vidéo ; par une technique d'apprentissage pour classifier des images [Image-Indexer] ou même par interaction avec l'utilisateur comme dans [Dillon et al. 98] qui propose un système d'annotation et de segmentation incrémentale d'images en référençant des informations entrées par les utilisateurs. Les résultats de ces techniques sont encore spécifiques, limités à un ensemble d'objets prédéfinis ou demander souvent l'interaction humaine pour extraire des informations de haut niveau sémantique.

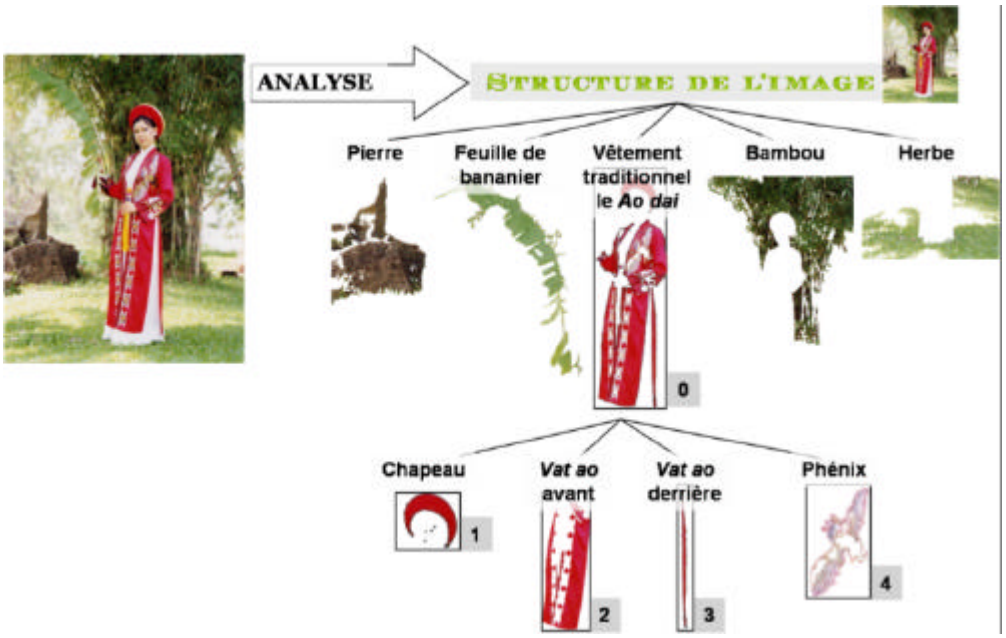

Figure 10. L'analyse d'une image en régions

Étant donné que les textes (comme une page HTML) sont des flux de caractères hiérarchisés en mots, expressions, phrases, paragraphes, sections et chapitres, la segmentation des médias textuels est plus facile que celle des médias graphiques. De plus, l'avènement et le déploiement des technologies XML permettent de créer des médias textuels de plus en plus exploitables par la machine, par exemple [Kunze et al. 01] propose une approche pour exploiter la connaissance de document Web basée sur l'intégration les technologies XML et le traitement du langage naturel. Cependant, il existe aussi des limitations au niveau de l'analyse sémantique à cause de la richesse et de l'ambiguïté de la langue naturelle, d'où le besoin d'associer des informations complémentaires sous forme de métadonnées comme avec l'infrastructure *Annotea* [Kahan et al. 01] qui fournit un système d'annotation de document Web basé sur RDF ou plus générale le sémantique Web.

III.2.1.2 Médias statiques vs continus

Dans la liste des médias visuels nous pouvons identifier deux types de média : les médias *statiques* et les médias *continus*.

Un média statique comme le texte et l'image présente toujours le même contenu pendant toute sa durée de présentation. Il représente donc un seul état d'une chose ou d'un fait ou bien d'un processus. Les informations que ce média transmet aux lecteurs sont des informations spatiales comme les couleurs, les textures, les formes et les dispositions des objets spatiaux les uns par rapport aux autres. L'analyse d'un média statique consiste donc à réaliser l'extraction de ces informations spatiales.

En revanche, un média continu comme la vidéo ou l'image animée présente un contenu dynamique qui évolue pendant la présentation. Il représente donc une évolution des choses, des faits ou d'un processus. Son contenu informationnel est non seulement constitué des objets spatiaux qui le composent mais aussi de leur

enchaînement dans le temps qui forme précisément le "récit" transmis au lecteur. Il intègre donc une notion temporelle dans sa structure de présentation.

On peut aussi considérer que le média continu est une séquence successive de médias statiques (voir la Figure 11). Plus précisément, cela signifie que des informations purement spatiales évoluent dans le temps et deviennent des objets mobiles. Par conséquent, l'analyse d'un média continu doit extraire non seulement des informations spatiales, mais aussi les trajectoires de ces informations dans le temps [Lin et al. 97] [Dubuisson et al. 01] (cf. la Figure 12). En d'autres termes, dans un média continu, la structure de présentation ne s'arrête pas à la structure spatiale comme dans un média statique, elle est aussi organisée dans le temps. De ce fait, la présentation d'un média continu peut être décomposée hiérarchiquement en des présentations unitaires (voir la Figure 11).

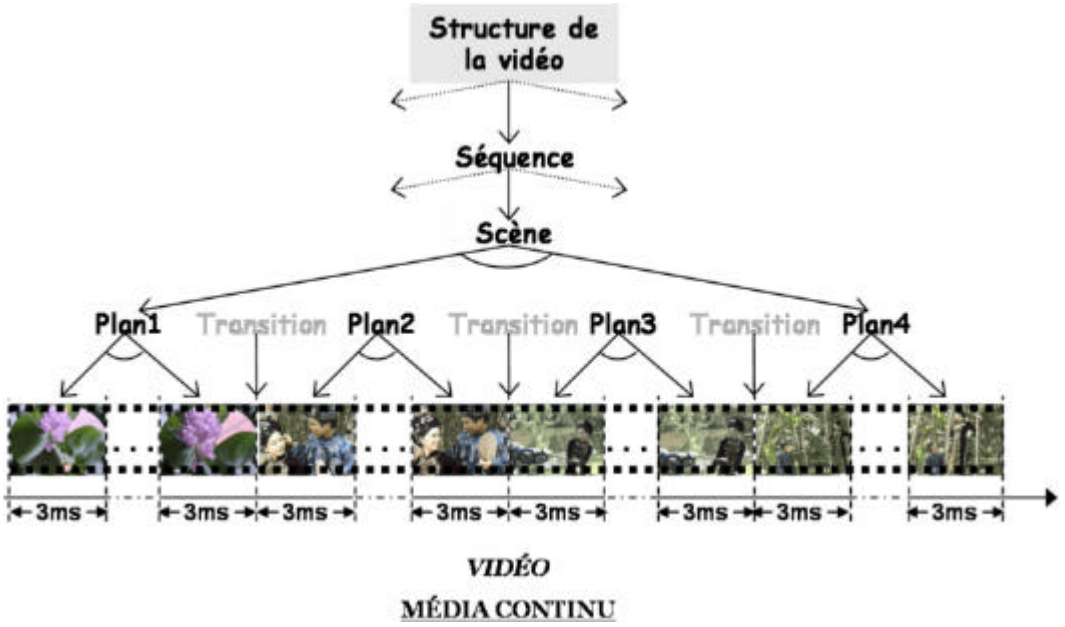

Figure 11. Une décomposition temporelle de média continu

Figure 12. L'extraction de l'objet *Fleur*, une région mobile, dans le *Plan1* d'un média continu

La décomposition de la présentation d'un média continu est basée sur des changements globaux d'informations comme la couleur, la texture ou même des indices ou caractéristiques plus consistants extraits des images entre des plans successifs d'une vidéo [Ardebilian 00]. Ces changements peuvent durer un court temps comme les types de transitions *fade-in*, *fade-out*, *dissolve*, etc. entre deux

plans de vidéo média, ou voir même être instantané comme dans le cas d'une transition *cut*. Ce type de changement est réalisé par le mouvement rapide de la caméra ou le traitement des images entre des clips de vidéo lors du montage d'une vidéo complète.

L'autre type de décomposition peut être l'identification de périodes significatives de la présentation du média comme la scène et la séquence ou même un petit événement (*event*) significatif dans le média [Hammoud et al. 98] [Wang et al. 01].

En comparaison de l'analyse de média statique, la décomposition dans le temps d'un média continu peut être aussi divisée en deux types de décomposition : la décomposition physique et la décomposition sémantique. La décomposition physique comme le découpage en plans et la détection d'objets mobiles est une évolution issue des méthodes d'analyse et d'extraction de régions dans un ensemble de graphiques successifs. Elle peut être effectuée automatiquement par des outils de découpage ou de détection d'objets mobiles [Lin et al. 97] [Dubuisson et al. 01]. Par contre, il n'existe pas encore actuellement d'outils automatiques et précis de décomposition au niveau sémantique. Il existe tout de même des travaux qui ont essayé de détecter des changements de scène dans la vidéo comme [Hammoud et al. 98] dans lequel la segmentation sémantique du film est basée sur les modèles de relation temporelle (Allen) entre des segments ; ou encore le travail plus perfectionné de [Wang et al. 01] basé sur le modèle cinématographique et les paramètres de la caméra. Toutefois, les résultats de ces travaux restent encore au niveau d'un regroupement d'un ensemble de plans qui bien souvent ne représentent pas correctement des entités sémantiques comme la scène ou la séquence de la vidéo média. Il est donc toujours nécessaire qu'une interaction humaine ait lieu pour compléter les résultats ainsi obtenus.

En conclusion, les médias continus sont plus attractifs par leur côté dynamique que les médias statiques pour transmettre une idée, mais ils nécessitent de transmettre des volumes d'information beaucoup plus importants et demandent des techniques d'analyse plus complexes.

III.2.2 Analyse des informations sonores

Les médias sonores sont aussi un type de média continu qui permet de transmettre des informations évoluant dans le temps. L'analyse de base des médias sonores consiste à extraire de manière régulière (en général, toutes les 20ms), une dizaine de coefficients spectraux ainsi que l'énergie du signal. Ces informations de bas niveau sont utilisées pour décomposer le média sonore en composants plus sémantiques comme les types de son "*musique*, *parole* et *bruit*" qui se retrouvent souvent synthétisés au sein d'une même source sonore. Cette décomposition peut se poursuivre pour chacun de ces composants, par exemple des mots dans la parole ou des séquences de notes en musique. [André-Obrecht et al. 02] propose une discussion plus approfondie du domaine d'analyse et d'indexation de document sonore.

III.2.3 Synthèse de l'étude de l'analyse du contenu multimédia

En résumé, l'analyse du contenu des médias est une première étape très importante dans toutes les applications multimédias pour augmenter la performance de traitement du contenu des médias. Cependant, pour des multiples raisons comme le mauvais codage des informations ou l'ambiguïté dans la représentation des couleurs, des formes, etc.) le résultat de l'analyse automatique est souvent incomplet ou erroné. Par conséquent, l'intervention humaine pour compléter et corriger le résultat de ces analyses est le plus souvent nécessaire, en particulier dans le cas de l'extraction d'informations sémantiques.

En effet, à cause de la difficulté et de la complexité de l'extraction sémantique, la plupart des outils d'analyse ne fournissent que l'extraction de caractéristiques physiques comme la couleur, les coordonnées d'une région ou d'un point intérêt. Cependant, l'application a souvent besoin de plus de caractéristiques sémantiques sur le contenu pour traiter plus efficacement les informations. Par exemple, la segmentation de la couleur rouge dans une image est très ambiguë pour des applications comme l'indexation et la recherche sauf si elle est attachée à une sémantique comme la couleur d'une fleur. Un autre exemple est donné par l'édition de document multimédia, qui a besoin d'informations plus typées pour pouvoir effectuer des compositions abstraites au lieu de compositions très spécifiques. Il faut donc avoir un module capable d'inférer dans le processus d'analyse pour grouper des informations brutes à un niveau plus sémantique. Cette tâche difficile est un des défis du domaine de l'analyse de haut niveau du contenu multimédia.

Comme mentionné ci-dessus, l'analyse fournit des résultats bruts qui peuvent contenir des ambiguïtés et des redondances. Ces résultats sont ensuite codés selon différents formats de représentation standard ou non (*MPEG-7*, *Vidéoprep*, etc.) de contenu qui sont étudiés en section III.3.

Enfin, l'étude ci-dessus n'a pas eu pour but d'envisager les méthodes et les algorithmes d'analyse du contenu multimédia, ainsi que leurs performances. Nous avons voulu identifier la capacité actuelle et future de ce domaine de recherche, et en particulier quelles caractéristiques du contenu des médias peuvent actuellement être extraites automatiquement par l'analyse. C'est en effet avec ces informations que nous pourrons construire notre modèle et notre outil de description du contenu multimédia, et déployer la composition des documents multimédias. Le fonctionnement harmonieux entre les trois phases de l'application multimédia (cf. la Figure 9) sera alors possible pour le type d'application que nous visons.

III.3 Description du contenu multimédia

Dans le contexte de l'explosion de la production du contenu multimédia et de la diffusion de documents électroniques, la description du contenu multimédia fournit une solution pour mieux gérer et déployer des ressources électroniques. Par exemple, l'indexation basée sur la description du contenu multimédia permet d'organiser et de gérer plus finement des bases de données multimédias ; La recherche d'informations multimédias peut donc répondre à des questions plus fines basées non seulement sur les caractéristiques de bas niveau du contenu, mais aussi sur les caractéristiques sémantiques ; de même la création d'un document multimédia a besoin d'accès plus profonds dans la structure du contenu de médias

pour intégrer plus finement des médias dans un document (par exemple, la synthèse une nouvelle vidéo à partir d'un ensemble de vidéos). La présentation d'un document multimédia ainsi créé sera beaucoup plus attractive, et l'archivage puis la recherche de ce type de documents sera plus efficace grâce à une structure explicite des synchronisations fines entre les médias.

Pour répondre aux besoins des applications, la description de contenu multimédia doit prendre en compte les caractéristiques générales suivantes :

- le modèle de description doit permettre premièrement de décrire des structures de base du contenu multimédia. Il doit contenir, par exemple, des informations sur la *couleur*, la *texture*, la *forme*, *etc.* pour des informations visuelles ; le *point* et la *région* pour les médias graphiques ; le *caractère*, le *mot*, la *phrase*, *etc.* pour le texte média ; la *région mobile* et le *point mobile*, le *plan* de média continu pour la vidéo, *etc.*
- la structure sémantique de la présentation du contenu multimédia doit être aussi prise en compte pour déployer plus intelligemment le contenu de média. Bien que, comme on l'a vu, les informations sémantiques ne puissent pas actuellement être extraites automatiquement, des descripteurs sémantiques peuvent déjà être ajoutés manuellement et de toutes façons, on peut espérer dans un proche avenir, disposer d'outils d'extraction automatique de ces informations.
- de plus, les modèles de description doivent être ouverts et extensibles. Ils doivent être le plus général possible pour s'adapter aux divers domaines ou même à chaque application spécifique.

La suite de cette section se consacre à étudier des modèles de description existants selon les critères ci-dessus. Cette étude sépare les modèles en deux parties : les *standards* et les *travaux spécifiques*.

III.3.1 Les standards généraux

L'utilisation large des métadonnées pour décrire des ressources électroniques demande à ces métadonnées d'être standardisées pour que les descriptions soient uniformes et interopérables. Aujourd'hui, un certain nombre de travaux cherchent à appliquer les standards comme le DC (*Dublin Core*), le RDF (*Resource Description Framework*) et le MPEG 7 (*Multimedia Content Description Interface*) à la description du contenu multimédia. Les standards fournissent des solutions générales disponibles pour que le plus largement possible des applications puissent les adopter et les utiliser facilement. Ils ont aussi l'avantage d'être largement acceptés parce que leur mode d'élaboration fait appel à des experts de différents domaines. Nous allons les présenter dans les sous-sections qui suivent.

III.3.1.1 Métadonnées de Dublin Core (DC)

DCMES (*Dublin Core Metadata Element Set*) est un ensemble d'éléments de métadonnées destiné à décrire les ressources électroniques. Cet ensemble consiste en 15 éléments regroupés en trois groupes de métadonnées (cf. le tableau ci-dessous) :

- *Content*, groupe relatif au contenu,

- *Intellectual Property*, groupe relatif aux informations sur la propriété intellectuelle,
- et enfin, *Instantiation*, groupe relatif aux informations sur le média lui-même.

Content	Intellectual Property	Instantiation
Coverage	Contributor	Date
Description	Creator	Format
Type	Publisher	Identifier
Relation	Rights	Language
Source		
Subject		
Title		

Ces 15 éléments de base de DC peuvent être qualifiés et raffinés (les *qualificateurs* permettent d'enrichir les éléments pour les adapter à des applications spécifiques ; tandis que les *éléments raffinés* limitent les portées de la signification des éléments) pour avoir des métadonnées plus riches encore. Les métadonnées du DC sont en relation avec des ressources qu'elles décrivent. Si le format du contenu de la ressource le permet, les métadonnées du DC peuvent être incorporées dans le contenu de média. C'est le cas des documents au format déclaratif comme HTML/XHTML [XHTML 00], ainsi que de tous les autres formats fondés sur XML comme les standards du multimédia SVG et SMIL [Kunze 99]. DC est donc plus pertinent pour les ressources textuelles dans lesquelles DC peut être utilisé directement dans leur corps.

DCMES peut aussi être utilisé dans d'autres modèles de métadonnées pour enrichir leur capacité de description. Par exemple l'utilisation conjointe avec RDF fournit un standard expressif de métadonnées.

DCMES est très général et peut donc s'appliquer en particulier à décrire des ressources audiovisuelles. [Hunter 99] a fait une proposition de l'utilisation additionnelle de DCMES avec MPEG-7 dans une structure de document vidéo. Une telle application de DC fournit une haute interopérabilité car DCMES est général, concis, interdisciplinaire, non-spécialiste et largement utilisé. Les quinze éléments basiques de Dublin Core sont utilisés pour décrire des informations de nature bibliographique à propos du document (par exemple, *Title*, *Author*, *Contributor*, *Date*, etc.). Pour aller plus loin dans la structure hiérarchique des documents vidéo, l'extension par qualification ou par raffinement des quatre éléments (*Type*, *Description Relation*, *Coverage*) permet de décrire des informations de plus bas niveau (*sequence*, *scene*, *shot*, *frame*). La Figure 13 (extraite de [Hunter 99]) présente la structure logique d'un document multimédia qui contient un document vidéo dont la structure est annotée par les éléments de Dublin Core et par des descriptions MPEG-7 (cf. III.3.1.3).

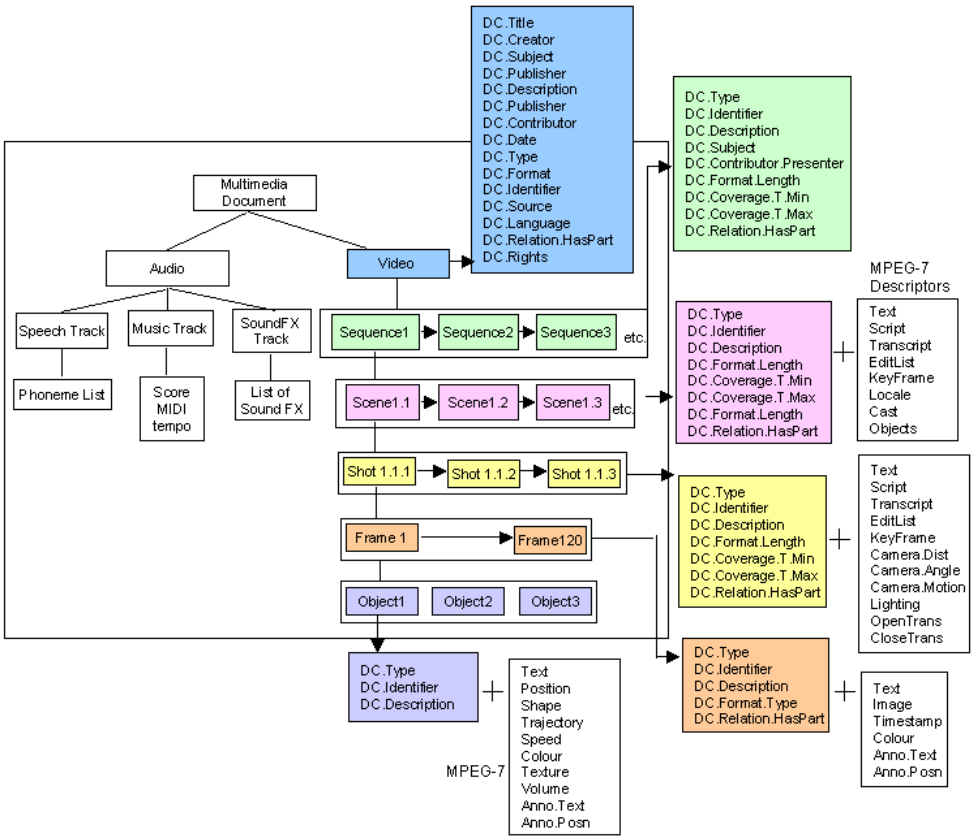

Figure 13. La structure hiérarchique et les attributs d'un document vidéo exprimé avec DC.

Nous étudions dans la suite de cette section comment décrire en DC les informations de structure de la vidéo à travers les utilisations de trois éléments : *Type*, *Relation* et *Coverage*.

III.3.1.1.1 DC.Type

Cet élément définit une catégorie de ressources. Par exemple, dans [Hunter 98] les catégories de ressources multimédias pour le contenu de l'élément *DC.Type* sont classées selon la hiérarchie ci-dessous :

?Image
- ?Moving
  - ?Animation
  - ?Film
    - Animation
    - Documentary
      - +*Sequence*
        - +*Scene*
          - +*Shot*
            - +*Frame*
              - +*Object*
    - etc.
  - ?TV
    - Documentary
    - News
    - Comedy

```
        etc.
    ?Photograph
    ?Graphic
```

Pour accéder plus finement aux composants d'une ressource, cette hiérarchie peut être développée pour identifier les éléments structuraux de celle-ci. Ainsi, on peut voir ci-dessus la décomposition d'un film en *séquences*, *scènes*, *plans* (*shots*), *images* (*frames*) et *objets*, ces derniers permettent d'identifier des personnages ou des objets dans les régions de suite d'images. Par exemple, les deux *DC.Types* présentés ci-dessous décrivent une scène dans une séquence d'un document et une image d'un plan d'une scène :

```
DC.Type = "Image.Moving.Film.Documentary.sequence.scene"
DC.Type = "Image.Moving.TV.News.sequence.scene.shot.frame"
```

III.3.1.1.2 DC.Relation

L'élément *Relation* permet de décrire des références entre ressources. Un cas particulier de relation est la relation hiérarchique qui définit l'inclusion des structures. Ainsi, l'élément relation peut être qualifié pour décrire des relations structurales par l'utilisation des sous-éléments *HasPart* et *IsPartOf* qui sont paramétrés par un attribut *Content* pour spécifier les descendants et le parent d'un composant. Par exemple, pour décrire la scène *scene3.2* qui est descendant de la séquence *sequence3* et qui est constituée des trois plans *shot3.2.1*, *shot3.2.2* et *shot3.2.3*, on peut utiliser les deux relations suivantes :

```
DC.Relation.HasPart Content= shot3.2.1, shot3.2.2, shot3.2.3
DC.Relation.IsPartOf Content= sequence3
```

Cependant, il est nécessaire de disposer de beaucoup d'éléments qualifiés pour l'élément *Relation* pour pouvoir décrire un ensemble riche de relations comme les relations temporelles *meets*, *co-starts*, *co-finishes*, *before*, *after*, etc. ; les relations spatiales comme *top-align*, *bottom-align*, *right-align*, etc. ; ou bien des relations structurales plus complexes comme les relations conceptuelles comme *exemple de*, *démonstration*, *prononcé par*, etc. Dans ce contexte, DCMES n'est évidemment pas adapté et devient un langage trop lourd.

III.3.1.1.3 DC.Coverage

L'élément *Coverage* décrit la portée du contenu de la ressource. Cet élément peut être utilisé pour décrire la localisation temporelle des composants de ressources comme : *clip*, *scène*, *plan*, *etc* dans une vidéo. Le format de la valeur du temps peut être une durée ou un temps absolu à partir de début. Par exemple, les moments où une ressource est déclenchée et arrêtée peuvent être décrits par les deux éléments qualifiés : *min* et *max* de l'élément *Coverage.t* de la façon suivante :

```
(SMPTE est un format de codage du temps)
DC.Coverage.t.min scheme=SMPTE content="09:45:23.14"
DC.Coverage.t.max scheme=SMPTE content="09:45:32.1"
```

De plus, les sous-éléments qualifiés de *Coverage* comme : *Coverage.x*, *Coverage.y*, *Coverage.z*, *Coverage.line*, *Coverage.polygon* et *Coverage.3D* peuvent être utilisés pour décrire des localisations spatiales et des formes pour les objets/personnages.

L'utilisation conjointe de **x**, **y**, **z**, **line**, **polygon**, **3D** avec **t** permet de décrire des informations spatio-temporelles comme le mouvement d'objet. Cependant dans ce domaine, l'ensemble des éléments de DC devient un outil peu approprié, particulièrement, dans le secteur des informations géographiques qui nécessite une grande quantité de mesures d'informations spatiales et temporelles[2].

III.3.1.1.4 Conclusion

DCMES est un standard de métadonnées intéressant ayant pour caractéristiques principales :

- ♦ il permet de définir des informations générales sur les ressources décrites,
- ♦ il s'applique à n'importe quel domaine,
- ♦ ses descriptions sont concises,
- ♦ sa syntaxe est simple et donc il est accessible à des non spécialistes.

Il permet d'annoter des bibliothèques de ressources informatiques de façon simple et interopérable sous forme de métadonnées. De plus, avec l'extension par qualification des éléments, le pouvoir d'expression est augmenté de façon significative. Cependant, la pertinence de DCMES est limitée au niveau de la gestion de ressources générales comme le titre, le créateur, l'éditeur, etc. A un niveau plus fin de description de la structure des ressources, seuls les éléments comme *DC.Type*, *DC.Identifier*, *DC.Description*, etc. peuvent être utilisés comme des métadonnées additionnelles. En effet, bien que DC fournisse l'élément *DC.Coverage* pour décrire la structure du contenu, cet outil est encore trop simple pour satisfaire des applications complexes comme la recherche basée sur le contenu, l'adaptation du contenu multimédia ou bien la composition de document multimédia. Il faut aussi prendre en compte la faiblesse de DC dans la description des relations spatiales, temporelles, conceptuelles, etc. entre des composants dans la structure du contenu des ressources.

III.3.1.2 RDF

RDF est un standard de description de métadonnées, conçu par le W3C. Son but est de fournir un mécanisme général approprié pour décrire de l'information sur n'importe quel domaine de telle façon qu'elle puisse être échangée entre des applications sans perte de signification. À titre d'illustration de l'utilisation de RDF on peut citer [Allsopp et al. 01] qui présente une construction d'une infrastructure pour augmenter l'interopérabilité entre des systèmes hétérogènes, grâce à l'utilisation de RDF pour la communication entre des systèmes. Par exemple une requête complexe d'un agent *A* peut être raffinée en plusieurs fragments au format RDF général (le triplet : <*sujet*, *prédicat* et *objet*>) avant de la transférer à un autre agent *B* qui est capable d'interpréter le modèle général de RDF.

---

[2] SCHEMAS - Metadata Watch Report #2 : 3.4 Geographical information sector, http://www.schemas-forum.org/metadata-watch/second/section3.4.html

RDF est fondé sur un modèle de triplet : *sujet*, *prédicat*, *objet*. Ce modèle permet de spécifier une description simple d'une ressource. Par exemple, cette thèse qui se décrit par le triplet : *l'auteur de cette thèse est Tien TRAN THUONG*, peut être décomposée en une déclaration RDF comme suit :

La ressource (*sujet*) : *cette thèse*
La propriété (*prédicat*) : *auteur*
La valeur (*objet*) : *Tien TRAN THUONG*

Le modèle RDF correspond à un graphe composé de nœuds et d'arcs qui permet aux applications de traiter les descriptions par des parcours de graphes (voir la Figure 14). Il permet de décrire les propriétés des ressources et les relations entre ces ressources. La représentation graphique de la déclaration RDF ci-dessus est donnée dans la Figure 14a : la ressource et sa valeur sont représentées par des nœuds tandis que la propriété représentant la relation entre la ressource et la valeur est représentée par un arc. La valeur peut être une simple chaîne de caractères ou une autre ressource. Si c'est une ressource, une autre déclaration RDF représente la ressource. Alors la propriété représente la relation entre des ressources et le graphe RDF est agrandi en conséquence (voir la Figure 14b).

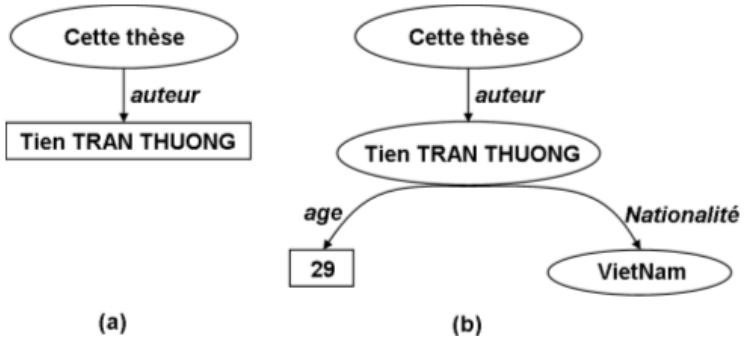

Figure 14. Deux graphes de RDF

Les descriptions ci-dessus ne sont pas directement utilisables par l'ordinateur. Pour les rendre manipulables par un ordinateur le modèle de RDF doit assurer que tous les composants (*sujet*, *prédicat* et *objet*) d'une description ont une identification unique, et que la description RDF est représentée sous un format accessible par la machine. L'architecture du Web aujourd'hui permet de fournir des solutions à ce problème de désignation.

L'identification des composants de description RDF doit être unique pour éviter le conflit entre des concepts. RDF utilise donc le modèle des URIs[3] (*Uniform Resource Identifier*) pour identifier les composants dans la description. En utilisant ce formalisme, l'exemple précédent (la Figure 14b) de la description RDF ci-dessus doit être représenté par les 3-triplets comme suivant :

[3] Uniform Resource Identifiers (URI): Generic Syntax, http://www.isi.edu/in-notes/rfc2396.txt

La ressource : «http://opera.inrialpes.fr/people/Tien.Tran-Thuong/These.html» (cette thèse)
La propriété : «http://opera.inrialpes.fr/example/terms#editor» (auteur)
La valeur : «http://www.inrialpes.fr/people/Tien.Tran-Thuong» (Tien TRAN THUONG)

La ressource : «http://www.inrialpes.fr/people/Tien.Tran-Thuong» (Tien TRAN THUONG)
La propriété : «http://opera.inrialpes.fr/example/terms#Age» (age)
La valeur : 29

La ressource : «http://www.inrialpes.fr/people/Tien.Tran-Thuong» (Tien TRAN THUONG)
La propriété : «http://opera.inrialpes.fr/example/terms#Nationality» (Nationalité)
La valeur : «http://www.vietnam.net/Welcome.html» (Viet Nam)

XML (*Extensible Markup Language*) est un format de représentation des données structurées qui est flexible, extensible et indépendant des applications. Son utilisation pour représenter et échanger les descriptions RDF est donc tout à fait adaptée. De plus XML fournit la technique des espaces de nom (Namespaces[4]) qui permet d'abréger un URI en un préfixe suivi d'un nom local. Par exemple `dc:label` est une abréviation de l'URI `http://purl.org/dc/elements/1.1/label` qui identifie l'élément `label` dans l'ensemble des quinze éléments de Dublin Core.

La description RDF ci-dessus peut ainsi être représentée en XML de la façon suivante :

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns# "
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:my="http://opera.inrialpes.fr/example/terms#">
    <rdf:Description rdf:about="http://opera.inrialpes.fr/opera/people/Tien.Tran-Thuong/These.html">
       <dc:label>Thèse de Tien TRAN THUONG</dc:label>
       <dc:title>Description de la structure des media pour l'environnement d'édition et de présentation de
documents multimédia</dc:title>
       <my:editor>
          <rdf:Description rdf:about="http://opera.inrialpes.fr/people/Tien.Tran-Thuong">
             <my:Age>29</my:Age>
             <my:Nationality rdf:resource="http://www.vietnam.net/Welcome.html"/>
          </rdf:Description>
       </my:editor>
    </rdf:Description>
</rdf:RDF>
```

La Figure 15 présente un graphique RDF de la description RDF ci-dessus généré automatiquement par le validateur RDF en ligne[5] du W3C.

[4] Namespaces in XML, http://www.w3.org/TR/REC-xml-names/

[5] http://www.w3.org/RDF/Validator/

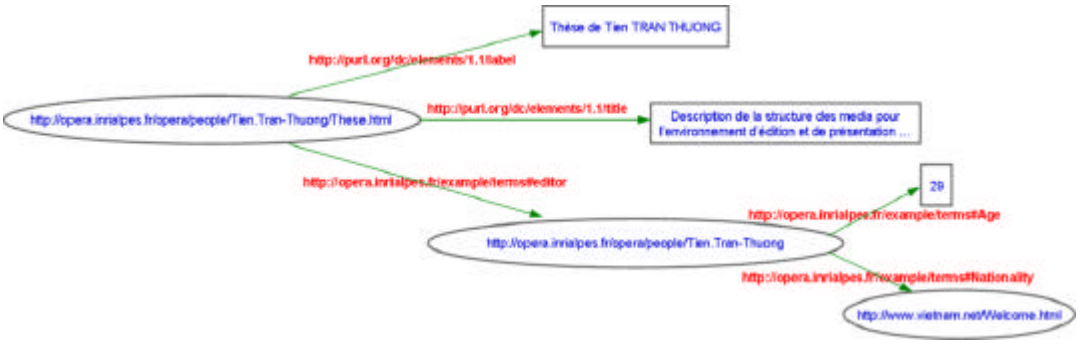

Figure 15. Le graphe RDF généré automatiquement par l'outil de validation du W3C

RDF fournit un modèle simple pour décrire de la même façon toutes les ressources. Cependant dans les applications il est souvent nécessaire d'utiliser des descriptions spécifiques. Par exemple, il serait intéressant de pouvoir classer la ressource identifiée par l'URL http://opera.inrialpes.fr/opera/people/Tien.Tran-Thuong/These.html dans le catalogue des thèses au lieu d'en avoir une description de manière trop générale comme ci-dessus. Pour répondre à ce besoin, RDF fournit un outil qui permet de faire évoluer la capacité de descriptions en type ou catalogue adaptées à des applications spécifiques. Cet outil est RDF Schema (RDFs). RDFs est un langage orienté-objet qui permet de définir des classes de ressources ou également des sous-classes qui héritent de classes existantes. Une classe RDF représente une collection ou un catalogue de ressources. L'extrait ci-dessous présente la création d'une classe *thèse* qui est utilisée pour typer la ressource décrite dans l'exemple précédent :

```
<rdf:Description rdf:ID="These">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdf:Description>
   ...
<rdf:Description rdf:about="http://opera.inrialpes.fr/opera/people/Tien.Tran-Thuong/These.html">
   <rdf:type rdf:resource="#These"/>
    <dc:label>Thèse de Tien TRAN THUONG</dc:label>
    ...
</rdf:Description>
```

Comme présenté dans [RDF 99], RDF est créé pour traiter automatiquement des ressources du Web, RDF peut être employé dans une variété de secteurs d'application comme la découverte de ressources, la définition de catalogues de bibliothèques et de répertoires mondiaux sur la syndication et le classement des nouvelles, l'organisation des collections personnelles de musique, de photos, etc.

Ainsi RDF fournit un modèle de métadonnées simples, qui peut être utilisé pour les descriptions sémantiques du contenu multimédia proposé. Par exemple, J. Saarela, dans [Saarela 98], a présenté un modèle de description du contenu de la vidéo basé sur RDF. Avec ce modèle, le contenu d'une vidéo peut être simplement annoté de la façon suivante :

```
...
<rdf:Description about="http://foo.bar/video.mpg">
   <v:Structure>
      <rdf:Seq ID="scenes">
         <rdf:li ID="Scene 1">
            <v:hasPerson>Janne Saarela</v:hasPerson>
         </rdf:li>
         <rdf:li ID="Scene 2">
            <v:hasPerson>Jay Leno</v:hasPerson>
         </rdf:li>
      </rdf:Seq>
   </v:Structure>
</rdf:Description>
...
```

Cette approche a été reprise par J. Hunter et L. Armstrong [Hunter et al. 99] mais de façon plus complète puisque leur schéma basé sur RDF permet de décrire non seulement de la structure sémantique (*scènes* et *personnes*), mais aussi la structure de composition de média (*séquence*, *scène*, *plan*, *image*, *région*, etc.). Ce schéma a été une des propositions initiales de langage de définition de description pour MPEG-7. En fait, il est facile de définir un schéma avec de nouveaux descripteurs pour définir la structure du contenu d'un média. L'exemple ci-dessous présente un schéma simple de la structure générale d'une vidéo (*Sequence*, *Scene*, *Shot*, *Frame*, *Object*, etc.) :

```
...
<rdf:Description rdf:ID="VideoContentDescription">
  <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  <rdfs:comment>Classe qui représente unecollection de la description du contenu de la video
</rdfs:comment>
</rdf:Description>

<rdf:Description rdf:ID="Sequence">
   <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:comment>Classe qui représente des sequ ences d'un document video </rdfs:comment>
   <rdfs:subClassOf  rdf:resource="#VideoContentDescription"/>
</rdf:Description>

<rdf:Description rdf:ID="Scene">
   <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:comment>Classe qui représente des scènes</rdfs:comment>
   <rdfs:subClassOf  rdf:resource="#VideoContentDescription"/>
</rdf:Description>

<rdf:Description rdf:ID="Shot">
   <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:comment> Classe qui représente des plans</rdfs:comment>
   <rdfs:subClassOf  rdf:resource="#VideoContentDescription"/>
</rdf:Description>

<rdf:Description rdf:ID="Frame">
   <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:comment>Représenter des images</rdfs:comment>
   <rdfs:subClassOf rdf:resource="#VideoContentDescription"/>
</rdf:Description>

<rdf:Description rdf:ID="Object">
   <rdf:Type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:comment> Représenter des objets dans des images</rdfs:comment>
   <rdfs:subClassOf rdf:resource="#VideoContentDescription"/>
</rdf:Description>
```

...

Pour décrire l'exemple ci-dessus par un schéma de description, nous définissons des propriétés (*contains_sequences, contains_scenes, contains_shots, contains_frames, contains_objects*) qui mettent en relation cet ensemble de vocabulaire, selon une structure générale de vidéo :

```
...
<rdf:Description rdf:ID="contains_sequences">
   <rdf:Type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
   <rdfs:domain rdf:resource="#VideoContentDescription"/>
   <rdfs:range rdf:resource="#Sequence"/>
</rdf:Description>

<rdf:Description rdf:ID="contains_scenes">
   <rdf:Type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
   <rdfs:domain rdf:resource="#Sequence"/>
   <rdfs:range rdf:resource="#Scene"/>
</rdf:Description>

<rdf:Description rdf:ID="contains_shots">
   <rdf:Type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property">
   <rdfs:domain rdf:resource="#Scene"/>
   <rdfs:range rdf:resource="#Shot"/>
</rdf:Description>

<rdf:Description rdf:ID="contains_frames">
   <rdf:Type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:domain rdf:resource="#Shot"/>
   <rdfs:range rdf:resource="#Frame"/>
</rdf:Description>

<rdf:Description rdf:ID="contains_objects">
   <rdfs:Type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
   <rdfs:domain rdf:resource="#Frame"/>
   <rdfs:range rdf:resource="#Object"/>
</rdf:Description>
...
```

Cette description illustre la capacité de RDF schéma à décrire une structure de contenu d'un média de type vidéo. Toutefois, RDF présente encore des limitations pour le domaine qui nous intéresse [Hunter et al. 99] :

RDF est rudimentaire pour la description de propriétés physiques comme la forme, le contour, la couleur, les histogrammes, la trace, etc. Ces caractéristiques ont besoin de types de données de base (comme les entiers, réels, vecteurs, matrices, probabilités, etc) qui ne sont pas directement accessibles dans RDF. De plus, les techniques de restriction ne sont pas supportées comme les intervalles de domaines de valeur (*min* et *max*) ou encore les cardinalités *min* et *max* d'occurrence d'éléments.

La description RDF est centrée sur les propriétés, ce qui rend la définition d'un schéma souvent plate, et du même coup peu lisible, en particulier dans des cas de structures à multi niveaux.

De plus la description des relations structurales de RDF est encore limitée. Elles sont basées principalement sur trois conteneurs Seq, Bag et Alt, qui ne fournissent pas assez de sémantique pour décrire les relations temporelles, spatiales ou bien spatio-temporelles.

Enfin RDF est trop général, il ne fournit pas un ensemble de vocabulaires standardisés pour décrire le contenu des ressources audiovisuelles, l'utilisateur doit donc utiliser des descripteurs existants ou en les créer. C'est un travail difficile qui n'est accessible que pour des experts.

En conclusion, RDF/RDFs fournissent une façon simple, générale, extensible et puissante pour décrire non seulement toutes les ressources qui sont identifiées par une URI, mais aussi des relations entre ces ressources. RDF ne vise pas à remplacer les autres standards de descriptions de ressources, en revanche, il permet d'utiliser des vocabulaires issus de différents schémas dans une description en utilisant la technique des espaces de noms. Ces qualités expliquent pourquoi RDF joue le rôle de liens entre divers schémas de différentes applications. Il joue donc un rôle majeur dans l'évolution vers une plus grande interopérabilité entre applications. Malgré cela, RDF n'est pas un outil adéquat ou assez sophistiqué pour la description du contenu des média à cause des limitations décrites ci-dessus. Mais il est une bonne solution pour organiser une base de données sémantiques dans laquelle des relations entre des médias sont décrites de façon exploitable par la machine et interopérable avec des autres bases. RDF fournit le moyen de non seulement trouver automatiquement des médias mais aussi de les intégrer automatiquement dans une présentation multimédia.

III.3.1.3 MPEG-7

Apres avoir présenté les standards d'encodage des informations audiovisuelles (MPEG-1, MEPG-2 et MPEG-4 - le format d'encodage basé sur des objets), le groupe MPEG (*Moving Picture, Expert Group*) a commencé depuis octobre 1996 à travailler à l'élaboration d'un nouveau standard de description du contenu audiovisuel et multimédia, MPEG-7 (*Multimedia Content Description Interface*). Si les précédents standards (MPEG-1, MEPG-2 et MPEG-4) ont contribué au déploiement des médias, MPEG-7 vise à les compléter pour diffuser ces ressources de façon plus intelligente. Pour situer le positionnement de MPEG-7 par rapport aux autres standards du groupe MPEG, nous donnons un exemple d'un flux vidéo auquel sont associées des métadonnées MPEG-7 (cf. la Figure 16).

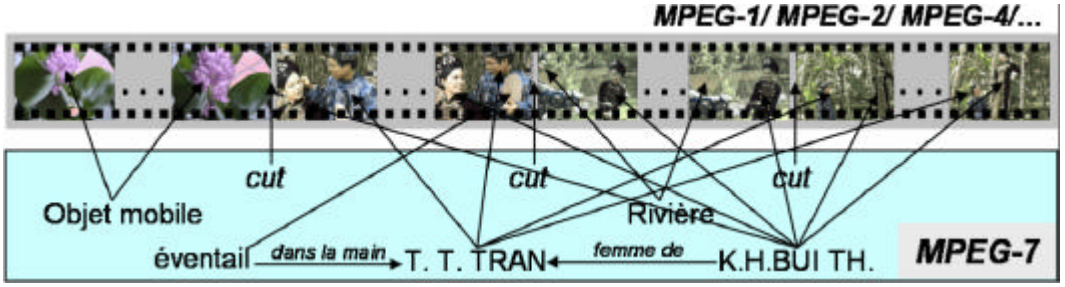

Figure 16. Les métadonnées MPEG-7 associé au flux de la vidéo

III.3.1.3.1 Corps de MPEG-7

MPEG-7 s'intéresse seulement à la description standardisée des informations audiovisuelles et multimédias (voir la Figure 17 - le corps de MPEG-7). Ces descriptions doivent permettre en particulier la recherche et le filtrage de données audiovisuelles. Par exemple, une image avec des descriptions permet d'identifier des objets dans l'image et d'afficher leurs formes, leurs mesures et leurs

caractéristiques ; ou bien un film émis avec des descriptions sur le contenu permet à un récepteur d'en enregistrer le contenu à l'exclusion par exemple des scènes violentes.

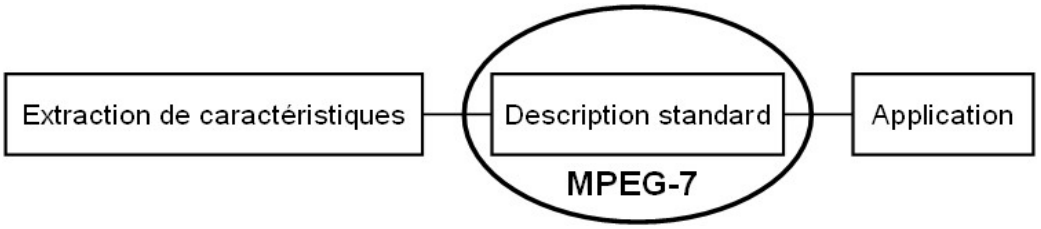

Figure 17. Le corps du MPEG-7.

MPEG-7 est aussi un cadre de travail de définition de métadonnées, mais il est différent des standards présentés précédemment (comme DC et RDF) qui supportent principalement les descriptions de haut niveau sémantique (Content management). En fait, MPEG-7 s'intéresse à un large éventail de niveaux de description [Salembier et al. 01] :

- les caractéristiques de bas niveau concernant le contenu comme la couleur, la forme, la texture, le mouvement ;
- la structure et la sémantique du contenu comme une scène contient un plan dans lequel il y a d'une jeune fille avec une petite chienne dans ses bras ;
- les collections et des classifications du contenu, en particulier, la définition de caractéristiques génériques de contenus.

Il fournit également un moyen simple et général pour l'échange et la réutilisation des descriptions du contenu de médias. De plus, des descriptions MPEG-7 peuvent être représentées en deux formats (au lieu d'un seul format textuel comme DC et RDF) : le format textuel XML pour supporter la recherche, l'édition, le filtrage et l'interopérabilité ; et le format binaire pour le stockage, le transport et la distribution continue [Seyrat 01].

III.3.1.3.2 Applications potentielles de MPEG-7

MPEG-7 est une norme générique et ouverte de la description du contenu multimédia supportant un grand nombre d'applications (*MPEG-7 Requirements*[6]). Les applications de Mpeg-7 sont divisées en 3 classes :

- Les applications "*pull*", comme la consultation de bases vidéo,
- Les applications "*push*", comme la télévision personnalisée, pour lesquelles il faudra permettre des filtrages. Cette classe inclut aussi la présentation d'information multimédia qui demande d'être capable d'organiser et de restituer de façon intelligente et automatique un ensemble de documents.
- Les applications spécialisées comme le téléshopping, l'édition multimédia, la capture de commande référencée, etc.

III.3.1.3.3 Ensemble d'outils de description MPEG-7

MPEG-7 fournit un ensemble riche d'outils de description qui sont suffisamment puissants pour décrire complètement le contenu multimédia. Pour cela, il supporte

[6] MPEG-7 Requirements http://mpeg.telecomitalialab.com/working_documents.htm

un large spectre de caractéristiques liées à la description d'un contenu multimédia et qui doit être considéré pour couvrir toutes les applications. Chaque application spécifique peut donc utiliser un sous-ensemble de ces descripteurs. De plus, des descripteurs peuvent être automatiquement extraits par des outils d'analyse et des traitements particuliers ou être spécifiés à la main par les utilisateurs. MPEG-7 standardise seulement les descripteurs et les schémas de description sans fixer comment ils seront extraits et utilisés. Ceci explique pourquoi MPEG-7 peut être utilisé par des applications comportant des techniques existantes d'analyse et des traitements spécifiques, ainsi il devrait pouvoir s'adapter aux évolutions futures de ce domaine.

Pour atteindre ces objectifs, MPEG-7 propose un ensemble de standards de descripteurs (Ds), de schémas de description (DSs), et un langage de description des définitions (DDL) :

- Des *descripteurs* (Ds) qui présentent les parties distinctives ou caractéristiques des données qui sont significatives (ex : un histogramme d'intensité lumineuse, la texture et la forme d'un objet, le texte d'un titre, l'auteur d'une vidéo, etc.),
- Des *schémas de description* (DSs) qui comportent en particulier des structures et des relations sémantiques entre descripteurs ou même entre des schémas de description,
- Un *langage de définition de description* (DDL) qui doit permettre en particulier la création de nouveaux schémas de description et de descripteurs. Il aussi doit permettre la modification et l'extension des schémas de description et des descripteurs existants. Dans un souci d'interopérabilité, MPEG-7 DDL est basé sur XML *Schema*.

Pour mieux comprendre les principes de conception présentés ci-dessus et leurs relations, la Figure 18 (issue du document MPEG-7 Requirements) présente les relations entre Ds, DSs et DDL sous forme d'un schéma UML. Des données audiovisuelles à partir des sources matérielles sont spécifiées sous forme de caractéristiques par le système d'observation ou l'utilisateur. Ces caractéristiques sont regroupées en descripteurs qui sont utilisés pour créer des schémas de description. Un descripteur peut appartenir à plusieurs schémas. Un schéma peut aussi être défini à partir d'autres schémas. Finalement un schéma est défini par un langage de définition de descriptions (DDL).

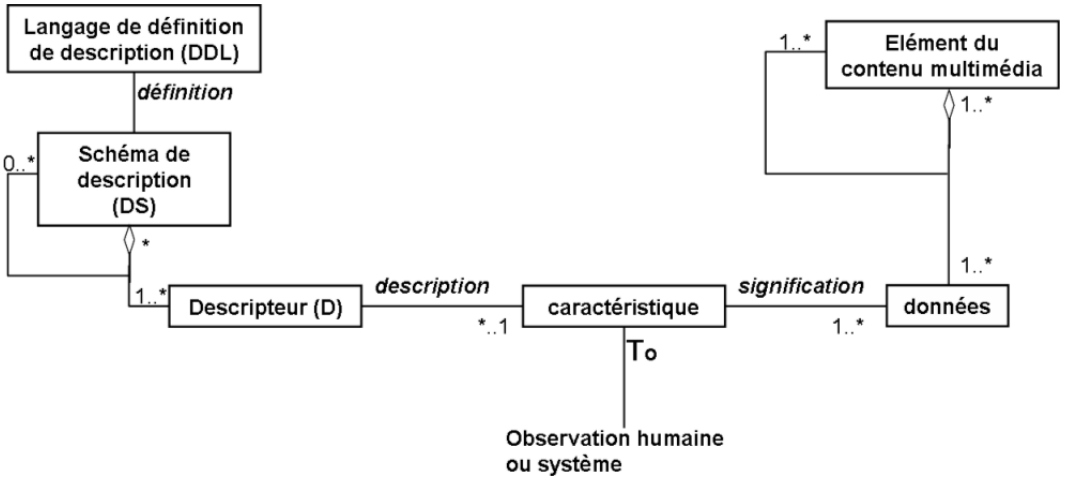

Figure 18. Présentation des relations entre Ds, DSs et DDL

III.3.1.3.4 Intégration de MPEG-7

Les métadonnées MPEG-7 peuvent aussi utiliser d'autres descripteurs issus d'autres schémas de description multimédia. Dans [Hunter et al. 00] Hunter et al. propose une méthode pour harmoniser MPEG-7 avec *Dublin Core*. L'intérêt est d'augmenter la capacité de description et d'interopérabilité de MPEG-7. D'ailleurs une description MPEG-7 peut être utilisée dans n'importe quel document XML par exemple des documents SMIL ou SVG.

III.3.1.3.5 Applications

La définition de MPEG-7 joue un rôle principal dans l'évolution d'une meilleure gestion du contenu multimédia. En fait, même s'il est encore en cours de construction, il y a déjà de nombreuses applications multimédias qui cherchent à utiliser ce futur standard. Un nombre important d'applications de MPEG-7 peuvent être consultées dans le rapport d'activité de MPEG-7 (le document *MPEG-7 Application*[7]). Nous citons quelques applications pertinentes pour notre travail. Ainsi, dans [Rutledge et al. 01b], L. Rutledge et P. Schmitz ont montré le besoin de médias au format MPEG-7 pour améliorer l'intégration de fragments de médias dans les documents du Web. Par exemple, une localisation URI d'un fragment peut être exprimée soit selon une désignation par nom (1) soit selon une désignation avec la structure MPEG-7 (2) :

(1)
http://www.examples.org/romeo.mpg#mpeg7(annotFile="http://www.examples.org/romeo.mpg7", clip="act2scene2line3")

(2)
http://www.examples.org/romeo.mpg#mpeg7(annotFile="http://www.examples.org/romeo.mpg7", act="2", scene="2", line="3")

Notons que les fragments de médias que l'on peut intégrer dans un document Web ne peuvent actuellement être issus que de documents structurés XML ou format comme HTML, SVG ou SMIL.

[7] MPEG-7 Applications http://mpeg.telecomitalialab.com/working_documents.htm

Par ailleurs le consortium TV-AnyTime dont l'objectif est l'utilisation de la télévision numérique pour fournir des services interactifs à valeur ajoutée, a aussi dit que la collection MPEG-7 de descripteurs et des schémas de description pour le contenu multimédia est capable de répondre aux besoins de métadonnées pour ce type d'applications [Pfeiffer et al. 00]. Beaucoup d'autres projets ont choisi MPEG-7 pour réaliser les systèmes qui permettent aux utilisateurs de chercher, naviguer et récupérer l'information audiovisuelle beaucoup plus efficacement qu'ils ne pourraient le faire aujourd'hui, car les outils actuels sont des moteurs de recherche principalement à base de texte [Day 01].

III.3.1.3.6 Conclusion

MPEG-7 offre le moyen d'obtenir des descriptions standardisées des divers types d'information multimédia. Cette description est associée au contenu des médias pour permettre aux matériels de traiter rapidement et efficacement l'information demandée par les utilisateurs. MPEG-7 se trouve au coeur de la plupart des travaux actuels pour la représentation et les applications de données audiovisuelles.

Cependant, parce que l'objectif de MPEG-7 est trop large (*un standard de description des **informations audiovisuelles** qui peut adapter à **toutes les applications***), l'ensemble des outils de MPEG-7 devient trop gros, tandis que pour des applications spécifiques il est trop général. Chaque classe d'application doit réaliser le sous-ensemble qui lui est adapté. C'est ce que nous proposons de faire dans cette thèse pour le domaine de la composition de document multimédia (voir le chapitre V).

III.3.1.4 Synthèse des travaux sur la description standardisée

Les standards fournissent un cadre général, partagé et bien adapté au développement d'un large éventail d'applications. Actuellement, les besoins d'utilisation et de gestion plus efficaces et intelligentes des ressources informatiques font naître un grand nombre d'applications de métadonnées. Les standards de description gardent alors un rôle de fédération pour ces applications qui peuvent communiquer grâce aux métadonnées facilement échangées et réutilisées.

Pour ces raisons, la plupart des standards ont aussi besoin d'un format servant de support à leur définition. XML, qui est considéré comme une évolution de l'ASCII, est le meilleur candidat à ce support. En fait, les standards de description de données fournissent toute leur puissance, s'ils sont représentés en format XML. Le schéma de la Figure 19 présente la position des représentations XML des standards de métadonnées sur lesquels des applications de description du contenu multimédia peuvent être construites.

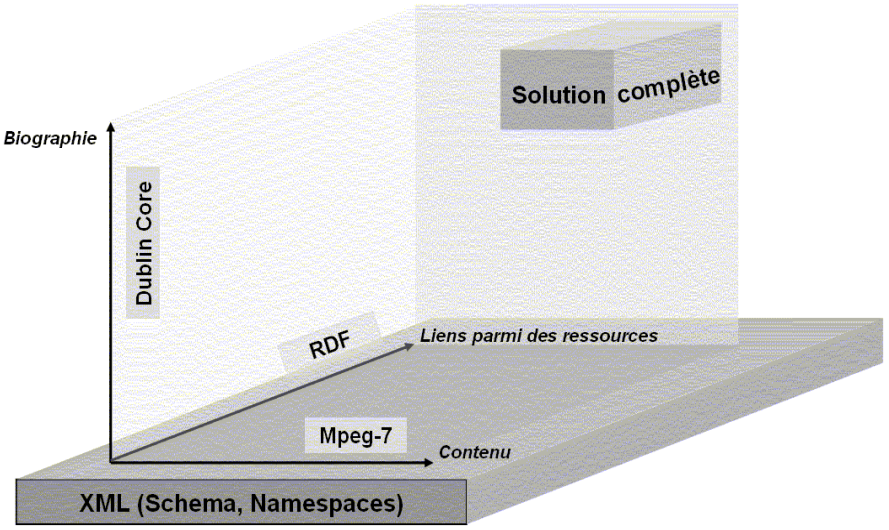

Figure 19. L'architecture des applications basées sur XML

Un standard est générique, mais son vrai potentiel est souvent localisé dans un niveau particulier. Par exemple, DC est spécialisé dans la description biographique ; RDF est une façon simple, mais puissante pour décrire des propriétés entre ressources ; et MPEG-7 est un ensemble riche d'outils dédiés à la description intra-média. L'utilisation harmonieuse de ces standards fournira la solution la plus complète. Par exemple, DC est souvent utilisé dans des descriptions RDF et MPEG-7 pour décrire des propriétés biographiques ; des propositions de construire MPEG-7 basé sur RDF ont été considérées [Saarela 98] [Hunter et al. 99] ; on peut même imaginer tirer parti des deux standards MPEG-7 et RDF pour générer des présentations multimédias à partir d'une base de données multimédias dans laquelle non seulement les médias et les descriptions MPEG-7 sont stockés mais aussi les liens sémantiques RDF entre les médias.

Il faut enfin noter que les applications spécifiques ont toujours des besoins particuliers qui ne seront jamais complètement pris en compte par les standards. Elles doivent donc dans ce cas là construire des outils adaptés à leurs propres besoins. C'est pourquoi il existe des travaux spécifiques que l'on va considérer dans la partie suivante.

III.3.2 Modèles de description spécifiques

En plus des travaux standard présentés ci-dessus, il existe de nombreux autres travaux concernant ce sujet. Certains sont relativement anciens ; tandis que d'autres ont débuté à partir d'appels à proposition pour la normalisation internationale comme c'est le cas de MPEG-7. D'autre part, certains de ces travaux sont basés sur les standards alors qu'au contraire d'autres sont totalement spécifiques à des applications particulières. Nous essayons de les classifier dans trois groupes :

- Les *prédécesseurs* : ce sont des travaux précurseurs du domaine.
- Les *contributeurs* : ces travaux visent à contribuer à la construction de standards.
- Les *développeurs* : ce sont des applications spécifiques qui n'utilisent pas ou n'appliquent pas de façon intensive les standards.

III.3.2.1 Les prédécesseurs

Les prédécesseurs appartiennent le plus souvent à l'une des deux approches suivantes : soit ils s'intéressent aux caractéristiques de bas niveau du contenu ; soit au contraire à la sémantique du contenu.

III.3.2.1.1 Méthodes utilisant les caractéristiques de bas niveau

L'idée basique de cette première l'approche est que l'utilisateur fournit des descriptions de quelques caractéristiques des médias qui sont utilisées par le système pour chercher dans une base multimédia des médias correspondants. Le système peut également fonctionner selon le principe de similarité entre médias : dans ce cas, l'utilisateur fournit au système un échantillon de médias qui portent les mêmes caractéristiques que ceux cherchés. Typiquement, les caractéristiques prises en compte dans cette approche sont la *forme*, la *taille*, la *couleur*, la *texture*, la *position*, le *mouvement*, etc. Par exemple, le système *QBIC* (*IBM Query By Image Content*) [Flickner et al. 95] propose et utilise le modèle basé sur l'ensemble de caractéristiques suivantes : *plan* (*Sketch*), *position*, *couleur*, *texture*, *localisation*, *forme*, *objet mobile* et *mouvement de caméra* (voir la Figure 20) qui permet de faire des requêtes sur de larges bases d'images et de vidéos. Un autre système typique de cette approche est *VisualSeek* (un système de recherche d'images à base d'indexation automatique du contenu [Smith et al. 96]). Il propose un modèle basé sur la région (caractérisée par la *couleur*, la *forme*, la *localisation*, la *texture*, le *mouvement* et la *taille*) et les relations spatiales entre les régions (comme *adjacent*, *proximité*, *recouvrement* et *entourage* qui peuvent être inférées du modèle *2-D String* [Chang et al. 87]). Il y a encore de nombreux systèmes construits sur le modèle à base du contenu, comme le système *PhotoBook* [Pentland et al. 93], le système *VIRAGE* développé par Virage, ou bien les systèmes *CANDID* [Keylly et al. 95], *JACOB* [LaCascia et al. 96], etc.

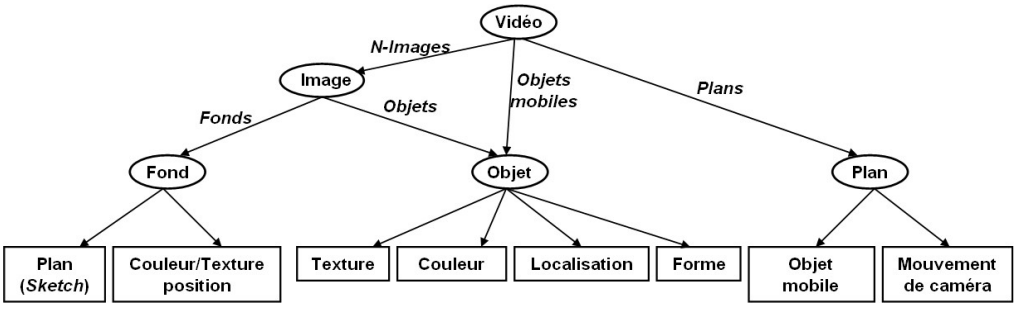

Figure 20. Modèle à base des caractéristiques du contenu de QBIC

Le principal avantage de tous ces systèmes vient du fait que les caractéristiques du contenu peuvent être extraites automatiquement. Toutefois, la requête formée des caractéristiques de base ne peut pas s'exprimer par des questions sémantiques comme "*Tien et sa femme sont assis sur un rocher au dessous des bambous*". De plus, certaines caractéristiques ne sont pas toujours extraites correctement, ou même doivent être extraites manuellement [Smith et al. 96]). C'est pourquoi certains auteurs comme [Tonomura et al. 94] proposent que des métadonnées soient éditées et ajoutées dans le flux du contenu de la vidéo pendant le processus de production de la vidéo. Il est ainsi plus facile de garder le contexte de production et donc l'analyse et l'extraction deviennent plus efficaces. Ainsi dans ce

travail, les informations du fonctionnement de la caméra sont enregistrées dans la source pour être utilisées par les outils de découpage en plan et par les outils d'extraction des informations plus sémantiques comme le mouvement d'objet, la direction du mouvement et les relations spatiales entre les objets.

III.3.2.1.2 Méthodes utilisant la sémantique du contenu

L'idée générale de la deuxième classe d'approche est de raffiner la sémantique du contenu en associant à des portions de contenu (appelés *événements*) des annotations sous la forme de texte naturel ou de mots clés. Les événements peuvent correspondre aux segments consécutifs du contenu (Figure 21a) [Chua et al. 95] [Ardizzone et al. 97]. Bien que ce modèle soit approprié au niveau le plus général de la description, il n'est pas assez flexible pour annoter en détail n'importe quel événement du contenu. Un modèle plus flexible doit donc accepter les recouvrements parmi des événements annotés (voir la Figure 21b) [Oomoto et al. 93] [Weiss et al. 94] [Jiang et al. 97]. Enfin, un troisième modèle encore plus évolué permet de collecter et classer les événements annotés dans des groupes qui permettent de décrire une structure plus sémantique du contenu (cf. Figure 21c) [Auffret et al. 98] [Vasconcelos et al. 98] [Decker et al. 99] [Tran et al. 00].

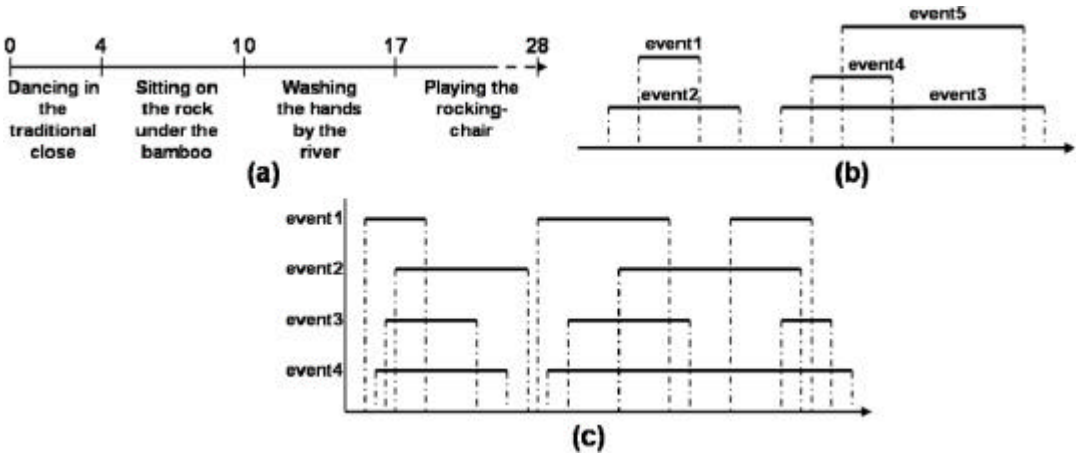

Figure 21. Trois modèles typiques de l'approche à base de la sémantique du contenu

Nous décrivons brièvement ci-dessous deux modèles représentatifs des travaux des deux dernières catégories d'approches : Video Algebra et AEDI.

Dans [Weiss et al. 94] [Weiss et al. 95] un modèle avec recouvrement des événements annotés est proposé pour la vidéo. Le modèle appelé Video Algebra représente un événement du contenu sous la forme d'une expression algébrique. Les vidéos sont décomposées en des expressions vidéo qui représentent la partie continue et temporelle d'une séquence vidéo comme des scènes, des plans, etc. Ces expressions sont créées, décrites, mises en relation entre elles par des opérations du modèle. Ce modèle fournit un ensemble complet d'opérations qui permettent de composer, rechercher, naviguer et jouer une vidéo.

AEDI (Audiovisual Event Description Interface version) [Auffret et al. 98] est une application indépendante développée par l'équipe de recherche en indexation de l'Institut National de l'Audiovisuel (INA). Son objectif est de fournir :

- un format de description de contenus audiovisuels,

- un format d'échange entre applications de ces descriptions,
- un encodage de métadonnées permettant un contrôle facile des documents audiovisuels.

Le modèle de description d'AEDI est une structure d'arbre (voir Figure 4) dont les éléments hiérarchiques sont : des *events*, des *units* et des *layers* pour représenter des événements, des annotations sémantiques et des regroupements. Les *events*, les *units* et les *layers* ont des propriétés définies par l'utilisateur. La liaison entre des noeuds représente des propriétés qui peuvent référencer d'autres objets.

- Un *event* représente un segment temporel contenu dans le document audiovisuel et peut être annoté par des objets *Unit*. Grâce aux propriétés de recouvrement ou de séquence, qui peuvent être associées à ces événements, AEDI couvre les modèles a et b de la Figure 21 : annotation sur des segments sans ou avec recouvrement.
- Un *unit* est un objet de l'annotation sémantique utilisé pour décrire un aspect non temporel d'un document audiovisuel. Par exemple, un *unit* peut représenter, un lieu spécifique, un personnage, un orateur, une technique de caméra utilisée dans un plan.
- Un objet *layer* est un point de vue du document ou une partie de ce document. Il permet de grouper logiquement des *events*, *units* et *layers* reliés. Cet objet correspond au regroupement sémantique des événements du troisième modèle de l'approche à base de la sémantique du contenu (cf. Figure 21c).

En conclusion, les travaux ci-dessus sont non complets et suivent les approches spécifiques. Ils n'offrent pas un métalangage avec lequel les utilisateurs pourraient spécifier leur propre schéma de description. Par exemple, le modèle AEDI, qui est un des plus complets dans ce type d'approche, fournit une extensibilité à travers la section *definitionSection*, mais elle ne permet pas de s'affranchir du cadre des trois objets *layer*, *event*, *unit* du modèle.

III.3.2.2 Les contributeurs

A partir de l'appel à contribution pour la construction du standard de description du contenu des informations multimédias MPEG-7, au début 1998, plusieurs propositions ont été faites pour contribuer à ce standard.

III.3.2.2.1 Proposition pour MPEG-7 DDL

Dans le travail de J. Hunter et al. ([Hunter 99]) un langage de définition de description (DDL) est présenté pour être une proposition de MPEG-7 DDL (cf. la section III.3.1.3). Le schéma est construit principalement sous la forme d'un schéma RDF. Il est complété par des relations, des spécifications spatiales et temporelles, et des capacités puissantes de description de types de données basées sur le schéma orienté objet SOX, etc.. Le langage s'appuie également sur les bonnes caractéristiques d'autres schémas comme SMIL. Par exemple, les définitions des blocs parallèle et séquentiel utilisent celles de SMIL. Le travail a défini dans un premier temps un langage qui satisfait les besoins[8] de MPEG-7

[8] MPEG-7 Requirements Document V.7, Doc ISO/IEC JTC1/SC29/WG11 MPEG98/N2461, MPEG Atlantic City Meeting, October 1998.

DDL. Cependant ce travail n'a pas été finalisé, au niveau de l'intégration fine des standards et surtout n'a fait l'objet d'aucune mis en œuvre. De plus, certains des aspects complexes du standard ne sont pas encore pris en compte comme les descriptions bas niveau des objets : la forme, la texture, le mouvement, etc. ainsi que les descriptions conceptuelles. Toutefois c'est une des propositions qui a servi de base au groupe MPEG-7 pour définir le standard sous forme d'un schéma XML avec des extensions.

III.3.2.2.2 Description à base d'objets et d'événements

[Paek et al. 99a] proposent pour MPEG-7 un schéma de description en XML de contenu d'image et de vidéo sous forme d'un ensemble d'objets et d'événements. Le travail propose deux descripteurs principaux : *Object* et *Event*. Le descripteur *Object* est le descripteur de base du schéma de description d'image. Le descripteur *Event* est le descripteur de base du schéma de description de vidéo. Les *objects* resp. *Events* peuvent être regroupés par définition de structures hiérarchiques reflétant l'organisation physique ou logique du média décrit. Ainsi, la structure physique de vidéo de la Figure 11 est décrite de la façon suivante :

```
<event_set>
   <event id='scene1' type='SCENE'>...</event>
   <event id='shot1' type='SHOT'>...</event>
   <event id='shot2' type='SHOT'>...</event>
   <event id='shot3' type='SHOT'>...</event>
   <event id='shot4' type='SHOT'>...</event>
</event_set>
<event_hierarchy type='PHYSICAL'>
   <event_node event_ref='scene1'>
      <event_node event_ref='shot1'/>
      <event_node event_ref='shot2'/>
      <event_node event_ref='shot3'/>
      <event_node event_ref='shot4'/>
   </event_node>
</event_hierarchy>
```

D'après la section précédente (III.3.2.1), on peut reconnaître clairement que ce schéma est hérité des modèles antérieurs (e.g., les notions *object* et *event* se trouvent dans les modèles QBIC, AEDI, etc.). L'intérêt principal de ce travail est qu'il propose de fusionner les deux approches précédentes (pour décrire des caractéristiques de bas niveau du contenu, ou pour la description de sa sémantique) puisque chaque composant de la description peut être soit de type physique, soit de type logique. Cependant le schéma proposé n'est pas complet notamment parce que l'intégration des modèles pour les images (*object*) dans les descriptions de vidéo (*event*) ne permet pas d'exprimer les caractéristiques dynamiques des objets de la vidéo (comme la déformation, le mouvement, les propriétés de la caméra, etc.). Dans la version suivante de ce travail [Benitez et al. 99] la description de l'objet vidéo est améliorée par le descripteur video_object avec l'attribut TYPE= 'GLOBAL'|'SEGMENT'|'LOCAL' qui peut décrire trois types d'objets vidéo : un objet logique (GLOBAL) représenté dans toute la vidéo (par exemple, le personnage Simba dans le film le Roi Lion) ; un objet temporel physique (SEGMENT) correspond à une région dans une suite d'images ; et enfin un objet LOCAL réfère à une région de l'image. La version [Paek et al. 99b], propose un graphe des relations d'entités <entity_relation_graph> pour décrire hiérarchiquement les

relations parmi des objets. Par exemple les descriptions spatiale et sémantique entre deux objets phenix (4) et Ao dai (0) de la Figure 10 peuvent être décrites :

```
<entity_relation_graph>
   <entity_relationr type="SPATIAL">
      <relation> On </relation>
      <entity_node object_ref="phenix"/>
      <entity_node object_ref="Ao dai"/>
   </entity_relation>
   <entity_relation type="SEMANTIC">
      <relation> décoration de </relation>
      <entity_node object_ref="phenix"/>
      <entity_node object_ref="Ao dai"/>
    </entity_relation>
</entity_relation_graph>
```

En résumé, ces contributions pour MPEG-7 ont fourni des schémas de haut niveau pour décrire le contenu de l'image et de la vidéo. Ce qui est synthétisé dans [Smith et al. 00] comme la notion de modélisation conceptuelle du contenu de l'image et de la vidéo. Actuellement plusieurs schémas de ce travail sont adoptés par MPEG-7 comme outils standard, par exemple, *Object DS*, *Event DS*, *Graph DS* et *Relation DS*. Cependant ce travail ne considère pas encore des descriptions spatio-temporelles de bas niveau comme l'évolution des caractéristiques dynamiques des objets et des relations entre eux. C'est le cas par exemple, de la déformation d'un objet à cause de son mouvement, d'un changement de relation spatiale qui survient lorsque, par exemple, une voiture double une autre voiture : d'abord *A* est derrière *B*, puis *A* est à côté de *B*, et enfin *A* est avant *B*.

III.3.2.2.3 Description à base de facettes multiples

Le modèle Infopyramid [Li et al. 98] propose une approche basée sur des facettes multiples pour décrire les contenus multimédias. Il permet de fournir des moyens riches pour accéder au contenu multimédia à travers différentes modalités (*multi-modalities*), différentes résolutions (*multi-résolutions*) et différentes abstractions (*multi-abtractions*).

- Dans *multi-modalitiés*, Infopyramid considère que le contenu multimédia peut être composé de différents types de média, ou même être enregistré en différents formats. Par exemple, la vidéo peut contenir des flots de vidéo, d'image, d'audio ou bien de légende textuelle. Les applications de recherche de média construites au-dessus du modèle *Infopyramid* peuvent supporter différentes modalités de requêtes sur un contenu multimédia, ou bien peuvent transformer un média dans la modalité demandée par la requête, dans le cas où la modalité demandée n'existe pas dans la base des médias.
- En *multi-résolutions*, pour adapter les présentations aux différents contextes de restitution, différents niveaux de résolution peuvent être spécifiés pour chaque type de média. Infopyramid permet d'ajouter des caractéristiques physiques et des informations sémantiques à chaque niveau de la résolution. La description complète constitue donc une pyramide des caractéristiques et des sémantiques.
- En *multi-abstractions Infopyramid* permet de décrire le contenu multimédia depuis ses caractéristiques physiques jusqu'à son contenu sémantique. Ces différents niveaux d'abstraction rendent possible l'accès aux médias selon un

large éventail de méthodes de recherche : à base de contenu jusqu'à des questions sémantiques.

La Figure 22 (issue de [Li et al. 98]) représente une pyramide de descriptions d'un journal de la télévision. La pyramide du journal a différentes modalités (le texte, l'image, la vidéo et l'audio) en diverses résolutions.

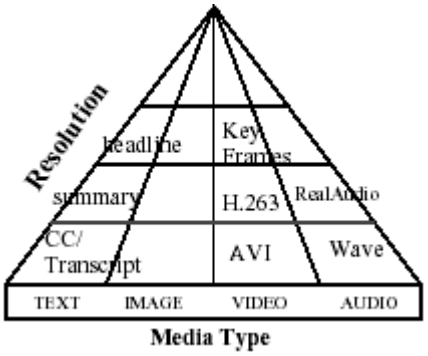

Figure 22. La pyramide d'un journal de la télévision

En résumé, la description basée sur *Infopyramid* fournit des méthodes riches pour accéder au contenu multimédia. Ce modèle supporte fortement non seulement l'adaptation du contenu multimédia, mais aussi la conversion parmi des modalités multimédias. Le modèle *Infopyramid* est donc adapté aux besoins d'une utilisation flexible du contenu multimédia. Ce travail a été repris en partie dans l'outil de description des variations du contenu multimédia de MPEG-7 (Variation DS et VariationSet DS).

III.3.2.3 Les développeurs

Dans le contexte de l'accroissement des besoins de traitement plus sémantique de données multimédias, le domaine de l'indexation et de la recherche de contenus multimédias a donné lieu à de nombreuses applications. On peut remarquer que le positionnement de ces applications vis-à-vis des standards de description de contenu multimédia dépend de la nature de ces applications :

- Pour les applications ayant des objectifs spécifiques ou touchant à un domaine précis, l'option de l'adoption des standards n'est pas réaliste. En effet, d'une part ces standards nécessitent de construire d'abord une infrastructure pour les supporter qui peut être coûteuse et complexe. D'autre part les standards sont souvent généraux, ce qui implique de les hériter et les raffiner.
- En revanche, des grands projets ont souvent des ambitions de diffusion large. Ils optent donc fortement pour les standards,
- Enfin certaines d'applications de portée générale ont besoin de supporter plusieurs standards. Par exemple, la gestion, l'indexation et la recherche sur les bases de données multimédias (texte, vidéo, audio, HTML, SVG, SMIL, etc.) demandent d'adopter une liste de standards comme DC, MPEG-7, RDF, etc.

Ces trois classes d'applications sont illustrées dans les sous sections ci-dessous.

III.3.2.3.1 Les modèles spécifiques

CARNet [Zelenika 01] est un système d'accès aux médias à la demande qui utilise un modèle propriétaire pour décrire sa base multimédia. Le modèle consiste en deux parties :

- Un modèle de description du contenu multimédia (*Media Description Scheme-MDL*) qui permet de décrire le média selon le schéma du processus de production du contenu média (définition du contenu, production et serveur) comme ci-dessous :

```
<?xml version="1.0" encoding="windows-1250"?>
...
<media type="video/mpeg" secure="yes" edit="yes" stream="yes" download="yes" embed="no">
   <title>...</title>  <file name="abcd.mpg" xml="abcd.mpg.xml" size="..." location="..."/>
   <production>
        <author type="creator" label="" name="" email="" contact="" date=""/>
        <info type="process" label=""></info>
        ...
   </production>
   <content length="..." bitrate="" language="..." subtitles="...">  ...
        <track type="video" codec="" bitrate="" parameters="" template=""/>
        <track type="audio" codec="" bitrate="" parameters="" template=""/>
    </content>
   <server type="http" label="HTTP pristup" address="http://..." info="Apache 1.3.12"/>
   <server type="stream" label="Streaming" address="rtp://..." info="RTP server v.x.x.x"/>
</media>
```

- Un modèle pour décrire l'organisation des fichiers multimédias en répertoires (*Folder Description Scheme - FDL*) avec les métadonnées associées.

```
<?xml version="1.0" encoding="windows-1250"?>
...
<folder xml="folder.xml" url="/hr/" secure="no" edit="yes">
   <title>ABC D...</title>
   <info type="description" label="Opis"> ... </info>
   <info type="copyright" label="Copyright">(c) Y2k CARNet</info>
   <info type="relation" label="URL">http://mod.rdlab.carnet.hr</info>
   <author type="admin" label="..." name="..." email="..." contact="..." date="30-09-2002"/>
</folder>
```

Le modèle CARNet MoD a été directement construit en XML sans faire appel à DC, RDF et MPEG-7. Les principales raisons sont la complexité et la généralité des standards qui demandent un coût élevé d'implémentation. Il a opté donc pour XML qui fournit d'une part la possibilité de choisir et représenter directement le vocabulaire adapté précisément à ses besoins, et d'autre part l'utilisation d'outils largement éprouvés comme le parseur XML ou les transformations XML vers HTML.

De même le modèle de [Dumas et al. 00] sert à décrire une base de vidéos en utilisant directement la structure logique classique du cinéma (Frame, Shot, Scene et Sequence) au lieu d'utiliser l'outil très général qu'est le schéma de description d'un segment général de MPEG-7 (Segment DS). Un langage de requête pour cette base est construit donc facilement sur cet ensemble de termes de la structure vidéo.

Les inconvénients de ce type de modèle sont évidemment la spécificité et la non interopérabilité.

III.3.2.3.2 Les projets basés sur les standards

Dans cette section on présente deux projets AGIR [AGIR] et DICEMAN [DICEMAN] qui sont représentatifs de cet axe de travail. Ces projets, d'envergure internationale, sont ambitieux dans leurs objectifs. Ils s'appuient donc sur les standards et les technologies émergentes (MPEG-7, algorithmes d'analyse automatique des signatures de média) pour pouvoir être plus largement adoptés.

III.3.2.3.2.1 AGIR

AGIR (*Architecture Globale pour l'Indexation et la Recherche*) est un projet établi entre plusieurs établissements français (AAR, INA, CERESYS, IRIT, INRIA, INT, LIP-6, AFNOR). Il comporte toute la chaîne de traitement des données multimédias : extraction des signatures de médias, langage de description multimédias et applications. L'objectif du projet est de développer des technologies et des outils nécessaires pour mettre en oeuvre une "*Architecture pour l'Indexation et la Recherche*" par le contenu de données multimédia, conformes aux exigences exprimées dans le contexte de la normalisation internationale. La Figure 23 (schéma extrait) présente les composants principaux de cette architecture.

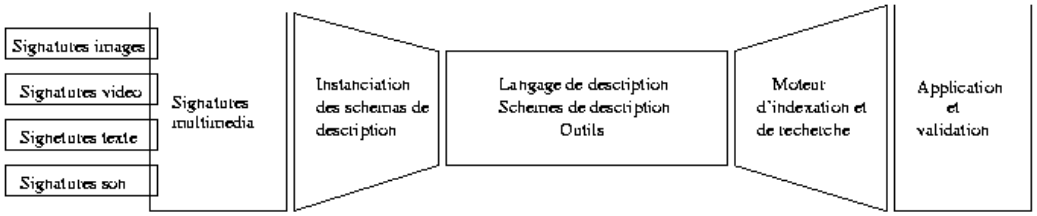

Figure 23. Les composants d'AGIR

Dans le cadre d'AGIR, l'enjeu global est d'obtenir des outils de production de descriptions basés sur la norme MPEG-7 pour les contenus multimédias. À un niveau plus modeste, le travail présenté dans ce mémoire suit la même démarche en proposant une chaîne complète de traitement d'informations multimédias, mais pour une application différente.

III.3.2.3.2.2 DICEMAN

DICEMAN (*Distributed Internet Content Exchange using MPEG-7 descriptors and Agent Negociations*) est un projet européen qui vise à développer un modèle de référence pour l'indexation, la description et l'échange de contenus audiovisuels en se basant sur MPEG-7. Les établissements participants sont : CSELT (IT), KPN (NL), Teltec (IE), IBM (DE), Riverland (BE), IST (PT), UPC (SP) et l'INA en France.

L'objectif principal de DICEMAN consiste à permettre l'échange de contenus audiovisuels sur Internet, et répond donc à un problème majeur auquel sont confrontés les départements d'archives, leurs clients, et de manière plus générale, l'ensemble des détenteurs de contenus. Pour cela, le projet met en place des innovations techniques concernant :

- la description et l'indexation de contenus audiovisuels ;
- l'indexation automatique et semi-automatique ;
- les interfaces utilisateur avancées pour l'indexation et la recherche ;

- les bases de données multimédias indexées ;
- la recherche et la négociation de contenus par agent.

III.3.2.3.3 Le modèle ABC – un point commun entre différents modèles de métadonnées

Le travail présenté dans [Lagoze et al. 01] est considéré comme définissant un point commun pour les différents modèles de métadonnées et sa structure s'appuie sur divers modèles de métadonnées existants comme *Dublin Core*, *RDF*, *UML*, *etc.* Le modèle *ABC* fournit un ensemble de notions de base pour :

- Pouvoir comprendre et analyser des vocabulaires de métadonnées existants et leurs descriptions ;
- Donner des repères pour les débutants qui veulent développer leurs propres vocabulaires ;
- Offrir des outils de traduction automatique entre des vocabulaires de métadonnées.

Une collaboration avec le CIMI Consortium[9] a été mise en place pour expérimenter le modèle ABC. Les quatre modèles de métadonnées de CIMI (1. Australian Museums Online – AMOL ; 2. Natural History Museum of London – NHM ; 3. National Museum of Denmark – NMD ; 4. Research Libraries Group/Library of Congress – RLG/LoC) ont été mis en correspondance avec le modèle ABC. A l'aide de feuilles de transformation XSLT, les métadonnées des images dans ces quatre formats sont transformées en des descriptions ABC, puis un outil de recherche exploitant le modèle ABC peut réaliser des recherches à travers toutes les bases d'images CIMI (voir La Figure 24, issue de [Lagoze et al. 01]).

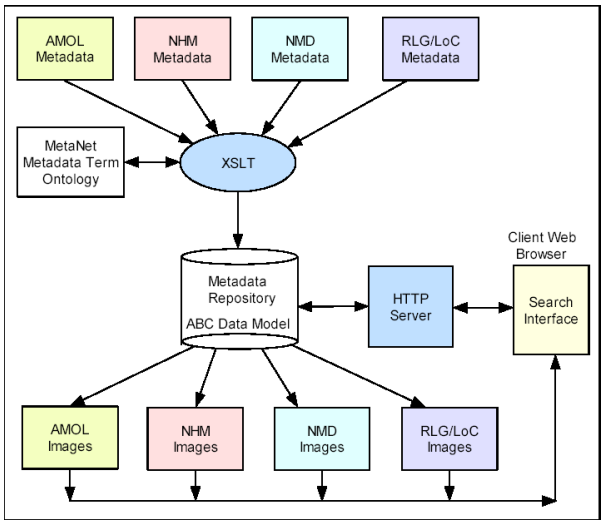

Figure 24. L'architecture basée sur le modèle ABC

III.3.3 Synthèse de la description du contenu de multimédia

Comme on vient de voir, la description du contenu multimédia a suscité un nombre très important de travaux de recherche. On peut classer ces travaux selon la nature des informations décrites :

[9] CIMI Consortium, 2001 http://www.cimi.org.

- les informations biographiques (Dublin Core), c'est l'approche de l'indexation du document textuel ;
- les relations entre des ressources (RDF), c'est l'approche pour organiser des bases sémantiques des ressources (*Web* sémantique) ;
- les caractéristiques de bas niveau (la forme, la couleur, la texture, le mouvement, etc.), c'est l'approche des outils de recherche basés sur le contenu (*QBIC, VisualSeek, etc.*) ;
- le segment temporel auquel est associé une annotation sémantique, c'est l'approche souvent optée par les bases de vidéos.

Selon le fondement choisi, chaque modèle réalise un sous-ensemble des besoins de description de média. Par exemple, le modèle basé sur l'objet du contenu multimédia permet de décrire physiquement et logiquement les objets de l'image et la vidéo, ainsi que les relations sémantiques parmi les objets ; la modélisation conceptuelle du contenu audiovisuel fournit la description des relations spatiales, temporelles, spatio-temporelles et sémantiques entre des objets. Enfin, les modèles qui s'appuient directement sur la structure de média sont basés sur les segments temporels ; etc.

Toutefois, aucun modèle parmi ceux que nous avons étudiés ne peut donner une solution complète pour décrire le contenu multimédia. Seul MPEG-7 intègre toutes les approches de description du contenu multimédia pour construire des outils standard de la description des informations audiovisuelles à travers tous les niveaux. De plus, il fournit aussi un langage (MPEG-7 DDL) pour adapter ces outils standard selon des besoins spécifiques. MPEG-7 fournit les solutions de base pour un large éventail d'applications. Cependant, en répondant à une classe étendue de besoins, ce standard devient complexe et inefficace pour certaines applications spécifiques.

L'arrivée de MPEG-7 a insufflé un nouvel essor des activités dans le domaine multimédia.

Pour contribuer à cette norme et la valoriser à travers le travail présenté dans cette thèse, nous avons choisi d'adopter MPEG-7 comme format de base de notre modèle de description du contenu multimédia. Nous verrons dans le chapitre 5 comment nous l'avons adapté à notre environnement d'édition et de présentation de documents multimédias.

III.4 Modèles de document multimédia

Dans la section II.2.3 qui discute de la conception du document multimédia nous avons qu'un modèle de document multimédia concerne en général cinq axes : *contenu*, *logique*, *temporel*, *spatial*, *hyperlien*, et *animation*. Dans cette section, nous allons envisager l'état actuel de ces axes.

Il existe un nombre important de modèles standard de document multimédia comme HTML, HyTime, MHEG-5/MHEG-6 et SMIL ainsi que des modèles non standard comme Firefly [Buchanan et al. 93], OCPN [Little et al. 90], CMIF/CMIFed [van Rossum et al. 93], ZYX [Boll et al. 99e], Madeus [Layaïda 97] et [Jourdan et al. 00], auxquels il faut ajouter de nombreux modèles pour des applications spécifiques comme [Celentano et al. 99], [Hsu et al. 99], [Stefan et al.

01], [Dattolo et al. 01], etc. Toutes ces propositions ont été bien synthétisées dans la littérature du domaine comme dans [Wahl et al. 94], [Blakowski et al. 96], [Layaïda 97], [Boll et al. 99b], etc. Par conséquent, nous ne faisons pas dans cette section la description, ni la synthèse des modèles existants du document multimédia. Nous n'étudions que des limitations dans chaque aspect des modèles existants qui empêchent la composition fine dans les documents multimédias.

Notre étude va être basée sur un exemple de besoin de composition d'une présentation multimédia dans laquelle un ensemble de fonctions de modélisation seront identifiées. Ces fonctions sont ensuite décrites et illustrées avec des modèles existants représentatifs : MHEG, HyTime, SMIL et Madeus. Madeus est le modèle principal qui a servi de base aux travaux présentés dans cette thèse (cf. la section III.4.8).

III.4.1 Scénario de l'exemple

Nous avons une vidéo et plusieurs images de mon mariage. En fait, mon mariage a été à la fois filmé et photographié. C'est pourquoi, chaque scène de la vidéo peut être représentée par une image. Nous souhaitons, à partir de ces médias, construire une présentation où chaque plan dans la vidéo va être synchronisée avec une image. Chaque plan possède un titre (sous forme textuelle) et associé à une courte description dans un fichier HTML. Nous voulons que les titres et les descriptions des scènes soient présentés de manière synchrone avec les scènes. Pour mieux comprendre cet exemple, nous citons un extrait de la vidéo, il contient quatre plans (cf. la Figure 11) associés à chacun de ces plans, un média texte contient le titre, une partie du document HTML en donne une description et une image le représente (cf. la Figure 25).

Figure 25. L'ensemble de médias et les correspondances parmi eux.

Les fonctions de composition multimédia nécessaires au scénario de ce document sont les suivantes :

Synchronisation : quand une scène est présentée, son titre est mis en évidence en changeant de couleur ou de police, l'image correspondant est agrandie ou subit un effet de transition (par exemple, par un effet de fondu à net), la partie correspondante à la description dans le fichier HTML est elle aussi mise en évidence ;

Animation[10] : le changement de la couleur ou la police d'un texte média ; le changement de la netteté de couleur d'une image média ; la mise en évidence sur une partie de texte dans le document HTML ;

Hyperlien : quand l'utilisateur clique sur un titre ou sur une description dans le document HTML, la présentation est synchronisée sur le plan correspondant ; et lorsque dans la vidéo apparaît un objet (la fleur dans le premier plan, cf. la Figure 11 et la Figure 12) l'utilisateur peut cliquer sur la région correspondante pour avoir plus d'informations sur l'objet.

Synchronisation spatio-temporelle : dans le quatrième plan (Playing the rocking-chair) nous mettons un texte temporisé dont le contenu se modifie dans le temps (My love,/ I know that/ I will always/ love you) et se situe toujours près du visage d'un personnage même quand celui-ci bouge pour montrer que c'est bien lui qui parle (cf. la Figure 26).

Figure 26. La synchronisation spatio-temporelle entre le texte média et le personnage de la vidéo

On peut noter que certains aspects de ce scénario peuvent être réalisés par des modèles existants. Cependant, les solutions fournies sont souvent spécifiées de façon absolue. Par exemple, on peut spécifier des temps absolus, qui correspondent aux temps du début et de la fin des quatre plans du vidéoclip, pour les présentations des textes, les images et même les hyperliens. Mais ces solutions absolues sont très limitées : difficiles à spécifier ; difficiles à maintenir ; manquant de relations directes parmi les présentations des médias ; etc. En plus, certains scénarios de l'exemple ne peuvent pas être spécifiés par des modèles existants comme les hyperliens sur un objet de la vidéo ou le texte qui suit le personnage de la vidéo.

[10] L'animation est un effet pour changer un ou plusieurs attributs de présentation d'un média.

III.4.2 Spécification du contenu

Les modèles existants ne permettent que très peu de spécifier le contenu des médias. En fait, par la technologie de l'identificateur universel (*URI*), les modèles font simplement référence à un flot de média brut (MPEG-1/2, AVI, MOV, JPEG, PNG, GIF, etc.) :

```
<video src= "rtp://opera.inrialpes.fr/people/Tien.Tran-Thuong/videos/TienWedding.mpg" >
```

Avec une telle technique, le modèle du document multimédia considère les médias comme de gros grains d'information. Il faut noter que, depuis l'arrivée de MPEG-7 on peut accéder directement dans le contenu de média, cependant cet avantage jusqu'à maintenant n'est déployé que pour récupérer un fragment de média [Rutledge et al. 01b] :

```
<video src= "rtp://opera.inrialpes.fr/TienWedding.mpg#mpeg7(clip="scene3 ")" >
```

Cela permet aussi d'éviter de couper la vidéo, toutefois le fragment récupéré par cette technique ne peut lui-même qu'être utilisé à gros grain pour composer le document. Par exemple, on peut utiliser cette technique pour récupérer l'extrait vidéo présenté ci-dessus dans notre exemple (celui qui contient quatre plans) à partir de la vidéo entière de notre mariage. Par contre, on n'a aucun moyen pour récupérer des informations sur les quatre plans ainsi que les objets de l'extrait pour composer la présentation désirée.

Par conséquent, un modèle de document multimédia permettant d'accéder à l'intérieur du contenu d'un média élémentaire est nécessaire. Il existe déjà des techniques pour accéder directement à l'intérieur du contenu d'un média (indexation/annotation), mais il n'existe pas encore de modèles multimédias pouvant prendre en compte ces techniques. Cela sera discuté plus en détail dans les parties sur les modèles temporels et spatiaux de documents multimédias.

III.4.3 Logique de présentation

Actuellement, il n'existe pas de modèle général pour décrire la structure logique des documents multimédias. La structure logique est donc directement issue des modèles temporel et spatial. Cela rend difficile à la réalisation de document multimédia parce que l'auteur ne peut pas toujours bien gérer la structure logique du document à partir ses structures temporelle et spatiale. Pour résoudre cette difficulté certaines applications offrent souvent différents modèles de document prédéfinis (*template*), par exemple pour les documents de présentation (*slideshow*) comme *PowerPoint*, pour les albums photos avec musique comme *RealPresenter*, etc.. Cependant cette solution rend le document souvent simple et limité au scénario spécifique. Une autre solution plus flexible consiste à décrire la structure logique séparément selon des schémas spécifiques qui sont fortement dépendants d'applications particulières. Ceci est l'approche prise dans le domaine de la transformation de document multimédia (cf. les sections II.2.3 et II.3.3). Nous ne développons pas en détail cet axe parce que la recherche d'un modèle général de la structure logique est en dehors des limites du travail présenté dans cette thèse.

III.4.4 Structure temporelle

Le temps est un axe particulier du document multimédia qui met en jeu les synchronisations temporelles parmi des présentations des médias. Il est bien étudié et modélisé dans un bon nombre de travaux de recherche [Allen 83], [Vilain et al. 86], [Little et al. 93], [Wahl et al. 94], [Duda et al. 95], [Layaïda 97], [Vazirgiannis et al. 98], etc. Dans cette section nous n'avons pas l'objectif de re-décrire ces modèles temporels. Nous allons plutôt chercher à classifier les modèles temporels présentés dans [Blakowski et al. 96] pour envisager leur capacité à exprimer le scénario présenté ci-dessus.

[Blakowski et al. 96] ont proposé les six critères suivants pour évaluer les modèles temporels :

1 La solution doit permettre de vérifier la cohérence et de maintenir des synchronisations.
2 La solution doit d'une part représenter des médias comme les unités logiques, d'autre part elle doit permettre la spécification de relations temporelles qui font référence à une partie de média.
3 Des relations temporelles entre médias doivent être décrites facilement.
4 L'intégration de médias continus (*time-dependent*) et de médias statiques (*time-independent*) doit être supportée.
5 La synchronisation hiérarchique doit être supportée pour pouvoir exprimer des scénarios complexes.
6 La définition des conditions de la qualité de service (QoS) doit être supportée.

Bien que la plupart de ces critères soient satisfaits par les modèles actuels, il y a encore très peu de modèles qui prennent en compte le deuxième critère. Pourtant cette condition est la plus importante pour notre travail. Les modèles de la catégorie "à base de points référencés" peuvent répondre à cette condition. La Figure 27 présente une spécification de la synchronisation entre la vidéo et les animations des quatre images spécifiées par le modèle de cette approche.

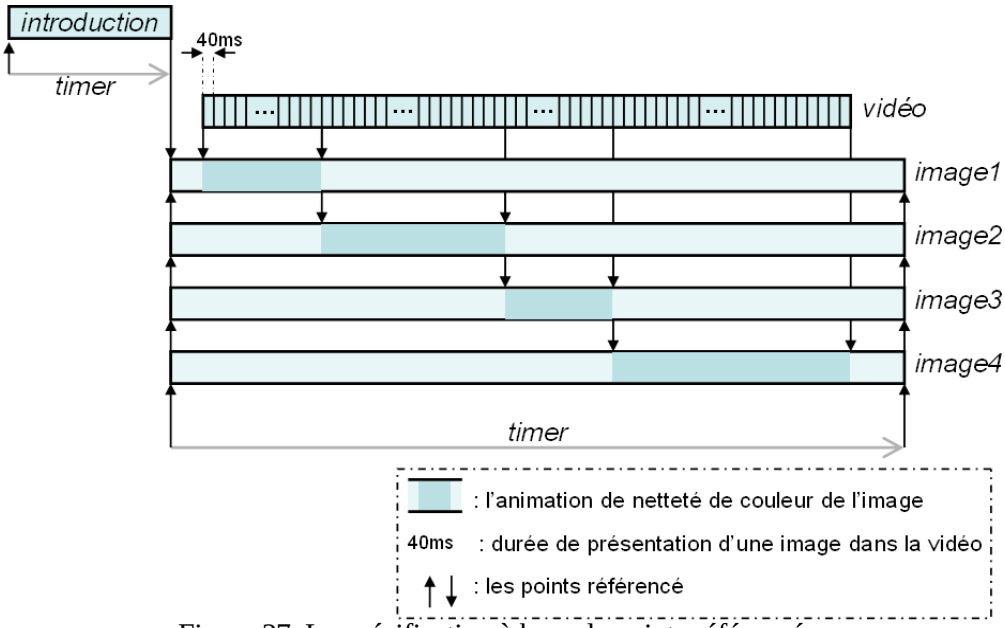

Figure 27. La spécification à base de points référencés.

Dans cette approche le début, la fin d'un média continu ou tout autre instant dans le média peuvent être référencés par un point temporel pour synchroniser des médias. Les instants internes des médias sont spécifiés sous forme de leurs unités logiques de données (LDU - *Logical Data Unit*), comme la durée de présentation d'une image dans la vidéo ou les numéros des images dans la vidéo. À partir de ces images unitaires, les points référencés correspondant aux débuts et aux fins des quatre plans dans la vidéo sont spécifiés et puis ils sont synchronisés avec les débuts et les fins des animations des images. Cette approche peut répondre au deuxième critère. Il permet de mettre directement des relations entre des parties de médias. Cependant la spécification des points référencés internes de média basé sur les LDUs est difficile à effectuer car les LDUs ne représentent pas la sémantique des parties internes de médias. De même ce modèle ne peut pas spécifier les points référencés internes dans le cas d'un média indéterminé. De plus, cette approche a les inconvénients de la spécification à base de points : manque de souplesse et lourdeur d'expression.

Les modèles à base d'intervalles permettent aussi de décrire des relations fines avec des parties de médias. La Figure 28 représente une spécification utilisant cette approche. Les animations sont mises en relation avec le média vidéo à travers des délais de sorte à assurer la synchronisation de chacune avec le plan correspondant.

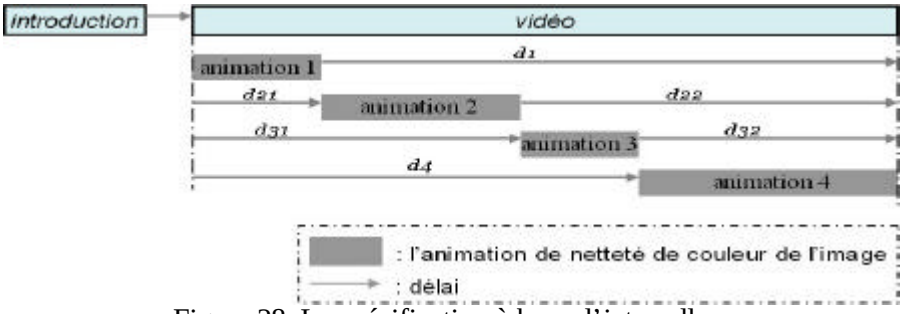

Figure 28. La spécification à base d'intervalles.

Bien que l'approche à base d'intervalles offre une plus grande flexibilité d'expression grâce aux relations, elle ne permet pas de spécifier directement des relations temporelles avec des parties de médias. Dans l'exemple, à cause du manque de modélisation des plans sous forme d'intervalle, on ne peut pas spécifier directement les relations avec ces plans. Les délais sont alors utilisés ce qui est peu sémantique.

En conclusion, les modèles temporels actuels fournissent un faible niveau d'expression pour spécifier les relations fines entre des parties de médias. De plus, la spécification de ces relations est complexe car la plupart des spécifications sont absolues et non significatives comme la spécification d'un point référencé ou d'un délai correspondant à une partie de média. Bien que l'utilisateur soit aidé dans sa tâche de spécification par des interfaces graphiques de haut niveau, la détermination des points référencés ou des délais est toujours difficile, de plus les spécifications automatiquement générées par les interfaces ne conservent pas la sémantique des relations.

En résumé, les modèles actuels ne peuvent pas représenter des parties de média comme des unités logiques. Alors bien qu'ils permettent de spécifier des synchronisations fines avec des parties de média, ces synchronisations ne peuvent présenter la sémantique des fragments. Le deuxième critère n'est pas pris en compte parfaitement.

La technique *Anchor* peut être utilisée pour résoudre le problème de représentation sémantique des portions de médias. Les relations avec des parties de médias sont spécifiées à travers les éléments Anchor. Dans [Hsu et al. 99], un AIU (*Anchorable Information Unit*) peut représenter une partie de média, par exemple, une phrase dans un document textuel, une scène dans une vidéo, une région dans une image. En fait, la sémantique d'un AIU dépend de la structure de chaque type de média comme cela est présenté dans le tableau ci-dessous (la Figure 29). Un document multimédia peut être spécifié grâce aux relations entre ces AIUs d'objets médias (vidéo, document, image, schéma, audio, etc.). Pour notre exemple, les AIUs *shot* permettent d'identifier les plans de la vidéo et on spécifie leurs synchronisations avec les animations par des relations temporelles "*equals*". Ce système a été utilisé pour créer des documents techniques multimédia de manuelles utilisations.

Media	AIUs
Free texte	Word, phrase, sentence, tec.
Photo image	Visible parts, area, frame, etc.
Video	Shot, scene, clip, etc.
Audio	Sample, bite, phrase, segment, track, clip, etc.
…	…

Figure 29. *AIU* caractéristiques (issue de [Hsu et al. 99]).

De la même façon, l'élément *Anchor* est utilisé dans des modèles standards comme HyTime et SMIL 1.0 (dans SMIL 2.0 l'élément *Anchor* est remplacé par les éléments *a* et *area* pour garder la correspondance avec ceux de HTML) pour spécifier une portion de média. Cependant, cette technique est actuellement utilisée pour spécifier des hyperliens, tandis que la capacité de synchronisation fine en utilisant cette technique a fait peu l'objet de développements.

De toute façon, cette approche est encore limitée du fait d'une spécification absolue des ancres. L'exemple suivant présente la spécification d'une présentation temporelle de l'objet *MyLove* de la vidéo.

```
<video src="rtp://opera.inrialpes.fr/people/Tien.Tran-Thuong/videos/TienWedding.mpg">
   <area id="MyLove" begin="17.85s" end="29s" ... />
</video>
```

On voit bien dans cet exemple le caractère absolu de la spécification à travers les attributs *begin* et *end*.

La sortie récente de la norme SMIL 2.0 propose l'utilisation de l'attribut *fragment* pour l'élément *area*. Cet attribut peut référencer directement une portion de média. De ce fait il fournit la façon plus logique de spécifier des portions de médias. Cependant, cette technique est limitée aux médias structurés comme HTML, SVG et SMIL.

Dans ce travail, nous présentons une solution pour présenter des portions de médias. Elle est basée non seulement sur les médias structurés comme la technique de l'attribut *fragment* de SMIL 2.0, mais aussi sur les descriptions du contenu des médias. En fait, la description du contenu des médias fournit des informations riches que l'utilisateur peut exploiter pour spécifier des portions de médias. De plus cette approche n'est pas limitée aux documents de structure explicite (HTML, SVG, SMIL). Des médias qui sont considérés jusqu'à maintenant comme des boîtes noires tels que la vidéo et l'audio peuvent être raffinés pour spécifier des relations fines entre parties de médias. Cette technique va être décrite dans le chapitre V.

III.4.5 Structure spatiale

Tandis que les modèles temporels utilisent différentes approches (point, intervalle et événement) souvent complexes, la plupart des modèles spatiaux de document multimédia (HyTime, MHEG, SMIL, ZYX, Madeus) sont basés sur la notion de *région*. Celle-ci est simplement un rectangle dans lequel un objet visuel peut être présenté. La synchronisation spatiale peut être classée en deux catégories selon qu'elle utilise un formalisme absolu ou relationnel. Bien que les deux approches permettent de placer finement un objet dans une région d'un autre, toutes ces spécifications spatiales sont absolues, la machine ne peut pas donc traiter la sémantique de ces spécifications. Par exemple, nous voulons placer le texte *My love* **au-dessus** d'un objet vidéo. Que ce soit par une approche absolue ou relationnelle nous pouvons obtenir le placement désiré (Figure 30). Mais ces spécifications ne sont satisfaisantes que pour des médias visuels statiques. Elles sont inadaptées pour la vidéo dans laquelle les objets vidéo sont souvent en mouvement. En effet la sémantique des spécifications n'est pas maintenue lorsque l'objet est en mouvement ou est déformé (le texte n'est plus **au-dessus** de l'objet vidéo). Le problème est le changement des informations spatiales au fil du temps.

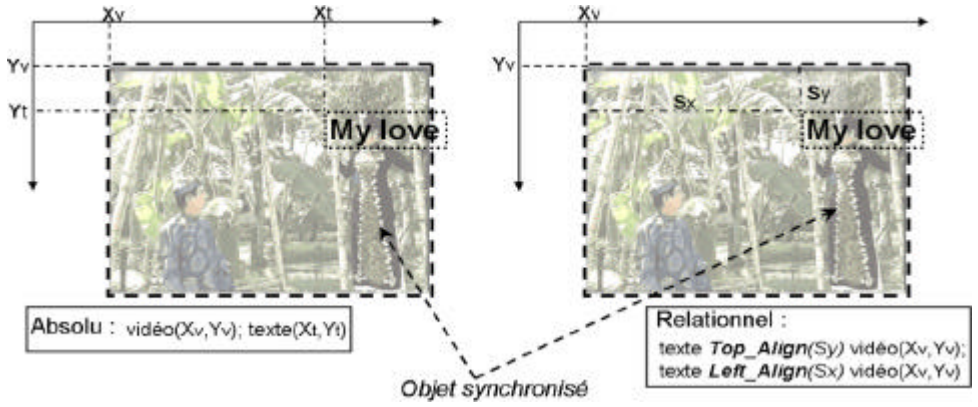

Figure 30. Spécifications d'une synchronisation spatiale avec un segment spatial de média.

La technique des ancres pour identifier un segment spatial de média pourrait être utilisée avec la spécification de relations spatiales pour résoudre ce problème de l'évolution des informations spatiales dans le temps. Malheureusement, cette technique est actuellement limitée à une spécification absolue. Par exemple, l'attribut *coords* peut être ajouté dans l'élément *Mylove* dans l'exemple précédent pour spécifier les coordonnées spatiales de cet objet vidéo :

```
<video src="rtp://opera.inrialpes.fr/people/Tien.Tran-Thuong/videos/TienWedding.mpg">
   <area id="MyLove" begin="17.85s" end="29s" coords = "X1, Y1, X2, Y2" ... />
</video>
```

Une telle spécification ne représente qu'une sous-région fixe dans une région. Elle ne peut pas représenter sémantiquement un objet dans un média. Tandis que l'utilisation de l'attribut *fragment* de l'élément *area* reste limité aux médias dans lesquels la structure est explicite (ex. HTML). Nous proposons que l'identificateur d'objet dans un média (comme les éléments *anchor*, *a*, *area*) soit lié directement à une description de l'objet dans ce média au lieu qu'il le soit à travers de l'attribut *coords* comme ci-dessus. La description de l'objet peut contenir des informations complètes de l'objet qui peuvent être utilisées pour identifier l'objet. Par exemple, le descripteur *movingRegion* de MPEG-7 peut fournir non seulement des caractéristiques physiques de bas niveau (couleur, texture, mouvement), mais aussi des descriptions sémantiques de l'objet (un personnage, une voiture ou un animal, etc.). Cette proposition sera décrite dans le chapitre V.

III.4.6 Hyperlien

Le concept d'hyperlien est bien modélisé dans les modèles existants MHEG, HyTime, HTML, CMIF, SMIL, etc. dans lesquels la technique *anchor*, *a* ou/et *area* est proposée principalement pour définir des hyperliens sur des portions spatiales, temporelles et même spatio-temporelles. Bien que cette technique permette de définir des hyperliens sur des sous-portions d'un média, elle est limitée aux sous portions fixes à cause de la spécification absolue (à travers des attributs *begin*, *end* et *coords*) des éléments *anchor* et *area* :

```
<video src="rtp://opera.inrialpes.fr/people/Tien.Tran-Thuong/videos/TienWedding.mpg">
   <area id="MyLove" begin="17.85s" end="29s" coords = "X1, Y1, X2, Y2" href="http://www.example.org/" />
</video>
```

L'identification plus logique des sous-portions que nous avons donnée dans les sections III.4.3 et III.4.4 ci-dessus peut permettre de placer des hyperliens sur des objets mobiles de la vidéo de façon plus logique que celle de la technique *anchor* ou *area* actuelle.

III.4.7 Animation

L'animation est un composant important dans une présentation multimédia. Dans l'exemple de document HTML+Time qui peut être consulté sur Internet[11], on voit toute l'importance des animations sur les images. La modélisation de l'animation réalisée sous forme d'une liste des composants (*animateur*, *objet* et *temps*), i.e., un animateur (fonction d'animation) peut animer un objet média pendant un certain temps. Il y a deux façons pour créer des animations : par programmation et par déclaration. Comme tout notre travail (et l'étude menée dans ce chapitre) suit une approche déclarative, nous présentons ci-dessous les modèles d'animation de cette catégorie.

Actuellement il y a deux approches de modélisation des animations : celles qui définissent l'animateur comme un élément générique dont le rôle est de changer d'état de présentation des objets médias dans le temps et celle qui utilisent un ensemble d'animations prédéfinies (comme dans SMIL).

III.4.7.1 Modèles avec animateur générique

Dans cette approche tous les animateurs sont décrits par un seul élément générique (la classe *action* dans MHEG ou la classe *modificateur* dans HyTime) dont la sémantique doit être implémentée dans chaque application spécifique.

Dans MHEG [Price 93], [Meyer et al. 95], un objet média peut être animé à travers des spécifications des actions (*action object*) qui sont des instances de la classe générique *action*. En principe, les actions sont spécifiées comme des exécutions d'objets médias pour la préparation, le démarrage, la modification et la terminaison des objets. De la même façon, le standard HyTime [HyTime:ISO 97] définit un modificateur générique (*Object Modifier*) pour modifier la présentation des objets médias. Cependant *HyTime* fournit des façons plus riches et plus souples pour planifier les modificateurs : soit ils sont attachés directement aux événements, soit ils sont planifiés dans un plan d'exécution de la présentation.

En résumé, par cette approche les modèles fournissent une façon pour planifier les actions de changement d'état de présentation des objets médias, tandis que la sémantique du changement (la sémantique de chaque animateur) est laissée à la charge des applications spécifiques.

III.4.7.2 Modélisation avec un jeu d'animateurs prédéfinis

Le modèle d'animation de SMIL [Schmitz et al. 01] identifie l'ensemble d'animateurs de base suivant :

- *animate* est une animation générale,

[11] Démo d'animation en HTML+Time :
http://www.ludicrum.org/plsWork/demos/orbit2.htm

- *set* permet simplement d'affecter pendant la période définie par l'animation une valeur à un attribut,
- *animateMotion* définit un chemin pour l'animation de mouvement,
- *animateColor* définit le changement de la couleur à travers le temps.

Un ensemble d'attributs d'animation associés aux éléments ci-dessus est aussi spécifié. Ces attributs fortement trois groupes principaux :

- les attributs pour cibler l'animation : *attributeName* spécifie l'attribut animé par l'animation ; et *targetElement* spécifie explicitement l'élément cible de l'animation.
- les attributs pour spécifier la fonction de l'animation. Ils comprennent : *values* qui contient une liste de valeurs ; et *calcMode* qui spécifie la fonction pour changer les valeurs dans la liste spécifiée par l'attribut *values*. Cette fonction peut être d'un des types suivants : *discrete, linear, paced* ou *spline*.
- les attributs permettant de spécifier le mode d'application de l'animation comme : le nombre de répétitions de l'animation (*repeatCount* ou *repeatDur*) ; l'état de l'objet après l'animation (*fill* avec une des valeurs : *remove, freeze, hold*, etc) ; l'accumulation (ou non-accumulation) de l'effet quand l'animation est répétée (l'attribut *accumulate* peut avoir la valeur soit *sum*, soit *none*) ; ou bien l'addition ou la non-addition quand la valeur de l'animation est combinée avec la valeur de l'objet (l'attribut *additive* peut avoir pour valeur : *sum* et *replace*).

Avec ces animations prédéfinies, le modèle d'animation de SMIL simplifie la spécification des animations des documents multimédias. Cependant, pour intégrer une animation dans une présentation, SMIL propose deux solutions : à travers l'attribut *targetElement* qui spécifie un ou plusieurs objets affectés par l'animation ; ou à l'inverse l'animation peut être mise dans le contenu de l'objet animé. Ces deux solutions limitent la capacité de réutilisation de l'animation, car une animation créée peut animer seulement un objet spécifié, aucun objet ne peut la réutiliser. De plus la façon de planifier une animation à travers des attributs temporels (*begin*, *dur* et *end*), quand elle est intégrée dans la présentation, fait perdre l'abstraction d'une animation. Une animation attachée à des informations temporelles devient un instant qui ne peut pas être réutilisé. Ces inconvénients rend fastidieuse la spécification d'animations dans un document SMIL. Par exemple[12], la spécification ci-dessous essaye de présenter cinq phrases dans l'ordre et appliquer sur chaque phrase la même animation qui change linéairement la couleur de blanc à noir. Une telle spécification est clairement inefficace à cause des cinq répétitions de la même animation et peut ensuite produire un travail fastidieux quand l'auteur veut modifier ces animations, par exemple, par le changement de la couleur de noir vers rouge.

[12] L'exemple se trouve sur le site :
http://www.ludicrum.org/plsWork/demos/H+Tdemos.html

```
<t:seq>
   <p timeContainer="par" timeAction="display">
      <t:animateColor attributeName="color" from="white" to="black" dur="3s"
                      autoReverse="true"/> In case you were wondering...</p>
   <p timeContainer="par" timeAction="display">
      <t:animateColor attributeName="color" from="white" to="black" dur="3s"
                      autoReverse="true"/>There is no script on this page. Everything ...</p>
   <p timeContainer="par" timeAction="display">
      <t:animateColor attributeName="color" from="white" to="black" dur="3s"
                      autoReverse="true"/> SMIL 2.0 Timing, Time manipulations and ...</p>
   <p timeContainer="par" timeAction="display">

      <t:animateColor attributeName="color" from="white" to="black" dur="3s"

                      autoReverse="true"/> Animated Filters are used for transitions, ...</p>
   <p timeContainer="par" timeAction="display">
      <t:animateColor attributeName="color" from="white" to="black" dur="3s"
                      autoReverse="true"/>Pretty cool, isn't it?</p>
</t:seq>
```

III.4.7.3 Synthèse

En résumé aucun modèle existant ne fournit un modèle de spécification abstraite d'une animation : 1) dans HyTime un modificateur définit indépendamment le temps, mais il fait une référence à l'objet modifié, le modificateur est dont créé pour un seul objet (voir la Figure 31a) ; 2) dans le modèle de MHEG une action est définie directement sur l'axe de temps réel et fait aussi une référence à l'objet activé, l'action est donc créée pour un seul objet dans un temps concret (voir la Figure 31b) ; 3) un animateur de base de SMIL est défini aussi sur un temps concret et soit fait référence à l'objet animé soit est intégré dans le contenu de l'objet animé. L'animateur de SMIL présente aussi le même problème que l'action de MHEG (voir la figure Figure 31b).

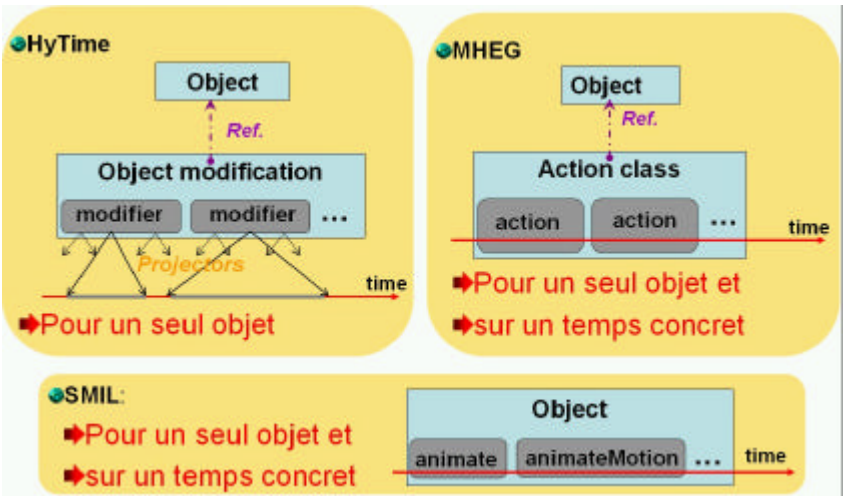

Figure 31. Spécification non abstraite d'animation dans les modèles de HyTime, MHEG et SMIL

L'édition d'une animation est difficile. De plus, dans une présentation un type d'animation peut être utilisé plusieurs fois. C'est pourquoi la réutilisation d'animation est importante. Pour cela, une spécification plus abstraite des animations dans la présentation multimédia qui permet de les réutiliser est nécessaire.

La contribution de cette thèse est donc de fournir un tel modèle pour spécifier des animations plus abstraites. Ce modèle abstrait est basé sur l'ensemble des éléments d'animation de base du module d'animation de SMIL [Schmitz et al. 01] (voir la section V.4).

III.4.8 Intégration des éléments de modélisation multimédia

Nous avons étudié ci-dessus les limitations des approches basiques de chaque dimension spécifique du document multimédia. Dans cette section une étude des limitations au niveau de la modélisation du document multimédia est effectuée. En fait, une intégration sophistiquée des modèles élémentaires des dimensions spécifiques est aussi très importante.

[André et al. 89] ont prouvé que l'utilisation d'une structure hiérarchique permet de spécifier des représentations complexes mais faciles à gérer. Les modèles actuels comme MHEG, CMIF, SMIL, ZYX et Madeus adoptent tous un tel modèle hiérarchique. Cependant, l'intégration des dimensions spécifiques de document multimédia n'est pas encore bien faite. En conséquence une spécification d'un document multimédia peut devenir complexe, redondante et donc difficile à maintenir.

En fait, dans les modèles existants, cette intégration est souvent complètement ou partiellement mélangée. Par exemple, dans XHTML+TIME, les spécifications spatiale et temporelle sont directement attachées aux éléments médias qui spécifient le contenu de l'objet média. Une telle spécification conduit rapidement à des documents complexes. A l'inverse, MHEG offre plus de souplesse grâce à la spécification du contenu d'un média qui est totalement indépendante des spécifications d'exécution temporelle et spatiale. Cela permet de définir plusieurs actions sur un même média. Mais la spécification spatiale est encore liée fortement soit dans la spécification de présentation d'un contenu média (*virtuel views*[13]) soit dans la spécification de l'exécution temporelle. C'est pourquoi la spécification spatiale est toujours nécessaire pour chaque présentation d'objet, même pour les objets présentés dans une même région. En revanche, dans le document SMIL, des canaux spatiaux sont définis séparément dans l'en-tête de document. Cela permet de réutiliser un canal spatial pour afficher plusieurs objets. Toutefois, la spécification temporelle est mélangée avec la définition du contenu de média et conduit à une plus grande complexité de la spécification temporelle. La distinction entre la définition des éléments médias avec leurs spécifications de présentation (spatiale, temporelle, lien, etc.) est prise en compte dans le modèle HyTime. Cela permet à un objet de pouvoir être présenté plusieurs fois dans un document. Cependant, les spécifications spatiale et temporelle sont couplées dans un seul objet de présentation. Cela induit des redondances non seulement dans la spécification temporelle mais aussi dans la spécification spatiale.

Ainsi, d'une manière ou d'une autre, tous ces modèles ne peuvent pas éviter la redondance dans la spécification. Le modèle de Madeus 2.0 [Villard et al. 00], par son approche qui sépare chacune des dimensions du document multimédia fournit un modèle très souple (cf. la Figure 32). Il permet non seulement d'éviter les

[13] Plusieurs présentations peuvent être définies sur un même contenu de média.

redondances dans toutes les dimensions mais aussi de réutiliser les spécifications de toutes les dimensions (contenu, acteur, temporel et spatial) :

- Contenu ***(content)*** : nous avons la même approche avec MHEG et HyTime qui permettent de définir le contenu de média indépendamment des informations de présentation. Ceci permet de réutiliser au maximum le contenu d'un média. Par cette approche les contenus de média peuvent être organisés dans une base de données multimédias vers lesquelles plusieurs présentations spécifiques peuvent faire des références.
- Objet de présentation (***Actor***) : comme l'approche de multi-vues (*virtuel views*) sur un contenu de média du modèle de MHEG, cette partie permet de spécifier tous les attributs de présentation d'un média (la police, la taille, le type, le lecteur, le volume, le composant *alpha* d'une couleur) à l'exception des informations temporelles et spatiales. Ces dernières sont l'évolution de Madeus modèle qui permet d'éviter les redondances dans les spécifications temporelle et spatiale du document. A noter qu'un l'objet de présentation a une référence sur une spécification du contenu de média.
- Spécification temporelle (***temporal***) : un point particulier très intéressant du modèle est que des spécifications temporelles sont définies indépendamment dans cette partie. Par cette spécification, le modèle est très différent des modèles multimédias actuels (HyTime, MHEG, SMIL, ZYX, etc.) qui mélangent des informations temporelles avec d'autres informations comme le placement spatial, les attributs de présentation ou même le contenu de média (SMIL). Un élément temporel spécifié référence une ou plusieurs spécifications d'objets de présentation.
- Spécification spatiale (***Spatial***) : selon la même approche que SMIL qui permet de définir des régions spatiales indépendamment les autres, cette partie du modèle est réservée pour spécifier proprement les régions spatiales utilisées pour afficher les médias. Un élément spatial spécifié référence au moins la spécification d'un objet de présentation dans la partie *Actor*.

La spécification du document multimédia est donc plus condensée, plus flexible et plus facile à gérer et à maintenir. De plus, cette souplesse permet de simplifier le traitement du document multimédia. Par exemple, les formateurs spatial ou temporel peuvent effectuer proprement le formatage sur le scénario spatial ou temporel du document ; le modèle supporte la transformation multimédia dans laquelle la feuille de transformation peut être séparée en différentes feuilles correspondant aux dimensions du document multimédia. Cette approche permet de diminuer la complexité de chaque feuille de transformation et donc du traitement de la transformation [Villard et al. 00].

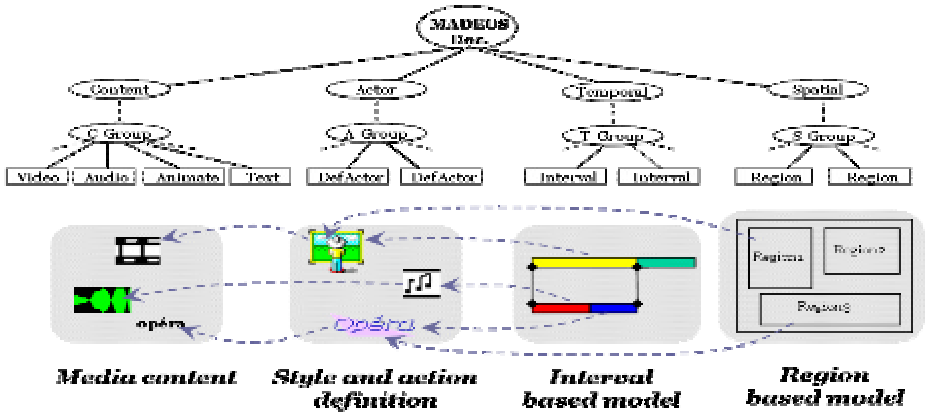

Figure 32. Approche de modélisation souple du document multimédia de Madeus

Une telle modélisation souple du modèle Madeus2.0 fournit un bon fondement sur lequel les travaux présentés dans cette thèse sont basés (voir le chapitre V).

III.5 Synthèse et objectifs de travail

Nous avons étudié dans ce chapitre les modélisations globales du domaine multimédia à travers trois niveaux : l'analyse de média, la description du contenu de média et la modélisation du document multimédia.

Nous avons considéré l'analyse du contenu des médias selon une classification des médias basée sur différents effets et modes de présentation du contenu de média : visuel et sonore ; textuel et image ; statique et continu, pour lesquelles les structures de base de chaque type du contenu de média sont identifiées. Bien que l'analyse du contenu des médias suscite beaucoup de travaux, elle reste encore limitée aux structures de bas niveau des médias. L'analyse automatique de la structure logique est encore peu fiable même sur des domaines d'analyse spécifiques comme la détection des visages et du mouvement du corps humain. Il est donc nécessaire d'avoir des outils pour vérifier et pour modifier les résultats d'analyse.

A travers l'étude des standards (DC, RDF et MPEG-7) et des travaux sur la description des contenus de média, nous pouvons noter que MPEG-7 est l'outil standard le plus important qui permet de décrire le plus complètement la structure du contenu des médias à partir des caractéristiques physiques de base jusqu'aux structures les plus logiques ; de plus, l'intégration des standards DC, RDF et MPEG-7 offre une bonne solution pour organiser toutes les informations de médias au sein d'une base de données multimédias sémantique.

Enfin, dans la partie de la modélisation des documents multimédias, nous avons identifié les limitations des modèles actuels à partir des besoins de la modélisation de chaque dimension spécifique jusqu'à leur intégration dans le modèle global.

En résumé, le modèle multimédia doit permettre d'incorporer efficacement les trois phases des applications multimédias : l'analyse, la description et le traitement. L'analyse permet de générer automatiquement les descriptions, qui sont nécessaires pour traiter efficacement les médias dans des applications multimédias. Cependant, comme nous le verrons dans l'étude des applications multimédias actuelles présentée dans le chapitre suivant, l'intégration de ces trois phases n'est

correctement effectuée que pour les applications d'indexation et d'accès aux données multimédias. Par contre, cette intégration n'est pas encore réalisée pour les applications d'édition et de présentation de document multimédias. La raison principale est la limite des modèles actuels comme nous l'avons montré dans ce chapitre. Le travail présenté dans cette thèse (chapitres V et VI) vise justement à proposer un modèle de structuration de médias et de composition de document qui répond aux besoins des applications de document multimédia.

Chapitre IV. Applications multimédias

Le chapitre précédent a fourni une vue globale sur les modélisations multimédias et les besoins d'un modèle intégral pour les trois étapes principales des applications multimédia (cf. la Figure 9) ont été identifiés. Ce chapitre a pour objectif d'étudier plus précisément les différentes catégories d'applications multimédias. En fait, à travers l'étude des applications multimédias existantes, nous identifierons les fonctions assurées par ces applications ainsi que leurs limites, notamment leur capacité d'exprimer et de traiter finement les médias. Nous pouvons retrouver dans ces applications les modèles étudiés dans le chapitre précédent.

Le chapitre est organisé comme suit : Nous commençons par l'identification des critères de base pour effectuer cette étude. Ensuite les applications sont présentées selon une classification en trois groupes : applications d'analyse et d'indexation des médias ; applications de production d'un média ; et applications d'intégration multimédia.

IV.1 Introduction

Les applications multimédias constituent un domaine très large, elles peuvent être classées selon les catégories suivantes : l'analyse et le traitement des média, le stockage dans des bases de données multimédias, la diffusion de présentations multimédias, le multimédia à la demande, la production multimédia et enfin l'intégration multimédia. En fait, les applications sont le lus souvent le résultat d'une intégration de ces catégories à différents niveaux peut donner une meilleure efficacité. Par exemple, l'utilisation de l'analyse et du traitement des médias peut aider à classifier automatiquement les médias dans une base de données multimédias (MAVIS), l'utilisation d'une base de données multimédias peut aider à trouver plus facilement des médias pertinents pour consulter, reproduire un autre média, ou l'intégrer dans une présentation multimédia (Microsoft Office XP ou Microsoft Picture It !), ou même l'utilisation de descriptions du contenu des médias peut aider à éditer des synchronisations plus fines dans des présentations multimédias [Rutledge et al. 01b] [Tran_Thuong et al. 02a], etc. Dans ce travail, nous nous concentrons sur le niveau de l'intégration multimédia. Nous étudions donc ce que peut apporter l'intégration de différentes applications multimédias dans un environnement d'édition et de présentation de document multimédia. Notre étude est basée sur la liste de critères ci-dessous, qu'une application idéale d'intégration multimédia pourrait avoir :

- L'accès à une base de données multimédias employant de multiples propriétés de média (temporel, spatial et sémantique).

- L'analyse, la génération des descripteurs et l'indexation du contenu de média.
- L'identification et l'accès fins à des segments spatiaux, temporels ou même spatio-temporels de média.
- L'intégration, la synchronisation, le lien et la structuration fine pour la production de présentation multimédia.

Ces critères non seulement s'appuient sur les caractéristiques traditionnelles d'un système d'intégration multimédia, qui sont la composition hiérarchique, la synchronisation et les hyperliens, mais ils demandent aussi à l'application idéale de fournir des capacités de récupérer facilement des médias pertinents à composer, et de les exploiter le plus finement possible pour permettre de composer plus facilement une présentation multimédia complexe et sophistiquée.

IV.2 Applications multimédias

Dans cette section nous envisageons les applications multimédias selon les critères ci-dessus. Le choix d'étudier de façon globale les applications multimédias, même si traditionnellement elles ne le sont pas car les objectifs de départ sont très différents, est motivé par la volonté de les confronter. Les applications sont classées en trois catégories : indexation, production de média et intégration de document multimédia.

IV.2.1 Indexation multimédia

Dans notre société moderne, les activités de la vie sont de plus en plus attachées aux données multimédias, c'est pourquoi les applications multimédias comme l'indexation qui aident à archiver, rechercher et consommer les informations multimédias plus facilement, plus rapidement et plus efficacement deviennent indispensables.

Les applications d'indexation multimédias sont variées. Elles dépendent de chaque base de données multimédias, de l'approche d'indexation (basée sur les caractéristiques du contenu, sur les annotations sémantiques, sur les informations biographiques, etc. cf. Chapitre III). En général, l'architecture d'une application d'indexation multimédia automatique doit pouvoir être dessinée comme le schéma de la Figure 33 (issue de [Carrive 00]). Le schéma présente le principe d'une application d'indexation multimédia automatique, ainsi que les composants principaux de données (base multimédia, base d'annotations, modèles multimédias, etc.) et les opérations principales d'indexation (extraction de primitives, interprétation et documentation) avec leurs entrées et leurs sorties.

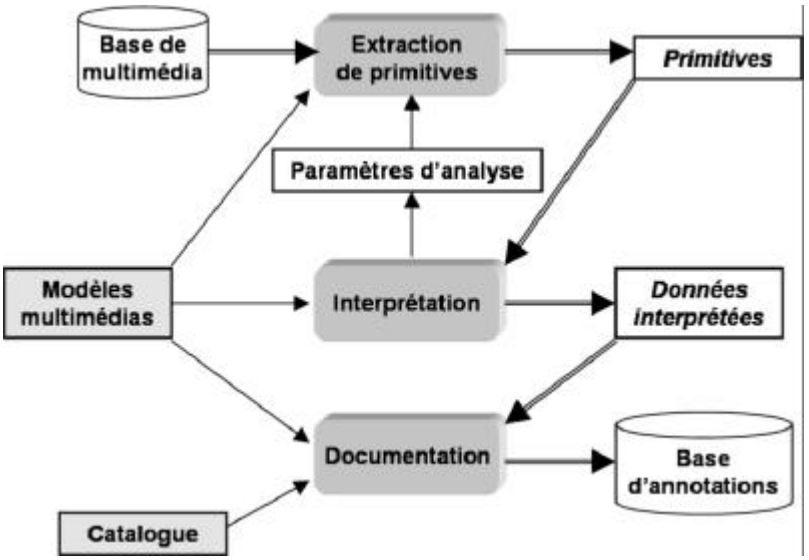

Figure 33. Schéma général d'une application d'indexation

Nous retrouvons la notion de "Modèles multimédias" du Chapitre I. En fait, le composant "Modèles multimédias" joue un rôle très important dans l'application multimédia, il caractérise la stratégie globale d'une application. Il permet de mettre en place non seulement les opérations automatiques comme l'extraction de primitives et l'interprétation, mais aussi des opérations semi-automatiques (comme la documentation). Le composant "Extraction de primitives" peut s'appuyer sur le modèle pour extraire les caractéristiques les plus élémentaires des médias. Puis l'interprétation, qui travaille sur les primitives extraites pour inférer des caractéristiques plus sémantiques, doit aussi se référer au modèle multimédia. Enfin le composant "Documentation" permet de visualiser les résultats des opérations automatiques pour permettre à un documentaliste de les vérifier et de les modifier. Il utilise le modèle pour afficher des indexations et aider le documentaliste à modifier ces indexations ou à créer les descriptions plus sémantiques.

Nous envisageons ensuite dans cette section des applications existantes d'indexation multimédia. Les applications sont classées selon la caractéristique de leurs objectifs.

IV.2.1.1 Système d'indexation/récupération à base de contenu

Les systèmes dans cette catégorie permettent d'effectuer l'indexation des médias automatiquement. Ils permettent aussi d'utiliser des techniques textuelles à partir des mots-clés associés aux médias. Ils permettent donc de rechercher un média en se basant soit sur des caractéristiques visuelles et graphiques comme l'image similaire, la texture, la couleur, etc. soit sur les informations techniques comme le mouvement de caméra, soit sur les mots-clés. Les applications QBIC [Flickner et al. 95], VisualSeek [Smith et al. 96], CueVideo [Ponceleon et al. 99], etc sont les applications typiques de cette catégorie.

Ainsi, QBIC est un système d'indexation automatique et d'interrogation basé sur le contenu de l'‘image et de la vidéo. Son modèle de structure de médias est simple :

- Les images se composent d'objets (des sous-régions d'une image), et
- Les vidéos se composent des plans qui sont une suite d'images et peuvent contenir des objets en mouvement.

En se basant sur ces simples modèles le système QBIC permet de segmenter totalement automatiquement des objets dans une image (le contour, la couleur, la texture). Le système permet de détecter des plans dans une vidéo, puis de créer une image représentative pour chaque plan. Un ensemble d'éléments de représentation des objets en mouvement peut être également généré automatiquement. Le système QBIC permet aussi l'interaction de l'utilisateur pour annoter manuellement les éléments détectés automatiquement. Le système accepte à la fois la recherche basée sur le texte et la recherche basée sur le contenu de média.

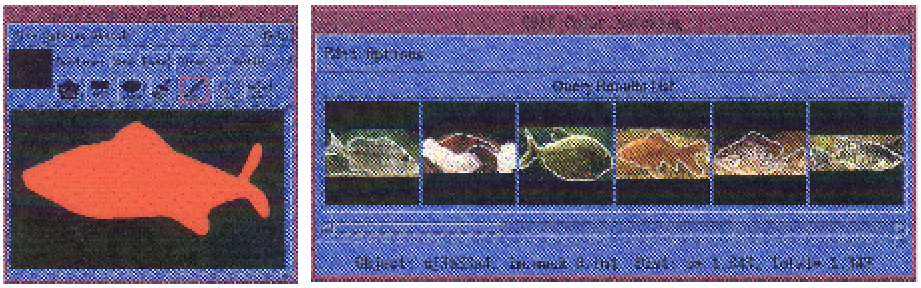

Figure 34. Exemple d'une requête de QBIC basée sur la forme

Très similaire au système QBIC, le système VisualSeek permet lui aussi d'indexer automatiquement une base d'images et de vidéos selon des caractéristiques visuelles. Il permet donc des recherches basées sur le contenu visuel. De plus, le modèle de données du système VisualSeek permet de décrire des relations entre objets (la description basée sur la représentation 2D-String [Chang et al. 87]), c'est pourquoi, l'utilisateur du système peut dessiner des questions visuelles qui utilisent non seulement les caractéristiques visuelles des régions mais aussi les relations spatiales entre elles (cf. Figure 35).

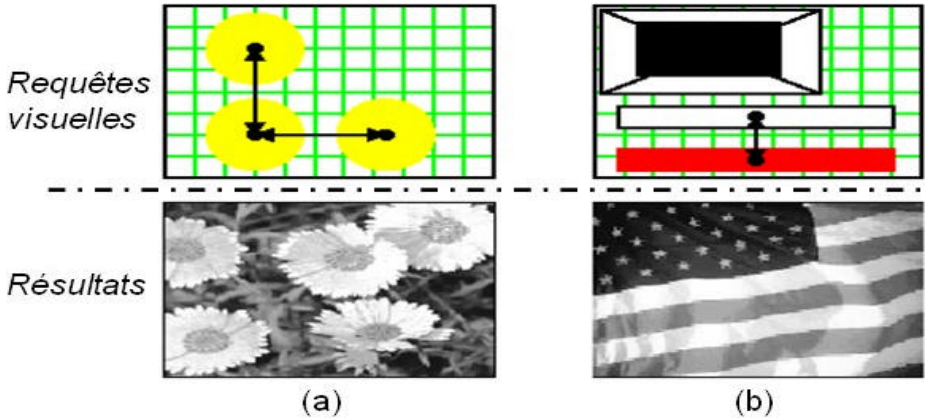

Figure 35. Exemple de requêtes de VisualSeek a) multiples régions avec localisations relatives, b) multiples régions avec localisations absolues et relatives.

IV.2.1.2 Système de gestion et de présentation vidéo (*video browsing*)

Les vidéos, qu'elles soient numériques ou analogiques, sont très souvent associées à des données textuelles : par exemple, le script d'un film, les métadonnées (titre,

auteur, données, résumé, etc.) dans une base d'archives. Il est donc important de pouvoir créer et exploiter ces informations avec les vidéos elles-mêmes.

CueVideo est une version améliorée du système *QBIC* pour analyser, indexer, chercher et présenter plus efficacement une base de médias vidéo. Le système *CueVideo* permet de créer les différents types de résumé vidéo comme : l'abstraction hiérarchique (*Storyboard*) et/ou l'abstraction séquentielle (*Smart fast play*). La partie la plus intéressante de ce système est le composant de recherche qui est basé sur des documents issus de l'analyse du flux audio extrait de la vidéo (*Speak to text*). Les textes générés des parties audio permettent de récupérer facilement les segments vidéo correspondants. Cette technique peut être intégrée dans les outils de recherche sur l'Internet comme *Yahoo*, *Altavista* ou *Google*, pour récupérer des vidéos clips à partir de questions textuelles. De même, elle technique peut être implémentée dans les systèmes de présentation de vidéo pour fournir la fonction *Control-F* (recherche) pour aider l'utilisateur à récupérer des segments dans la vidéo. La transformation de l'audio en texte pour indexer est aussi utilisée dans le système [Backfried et al. 01], un système d'archivage multimédia (cf. la Figure 36 issue de [Backfried et al. 01]). Ce système permet non seulement de convertir l'audio en texte, mais aussi de classer ces textes en différentes entités (NED) comme personne, localisation, organisation, etc. et en différents sujets (TD), pour enfin de produire des indexations en format XML.

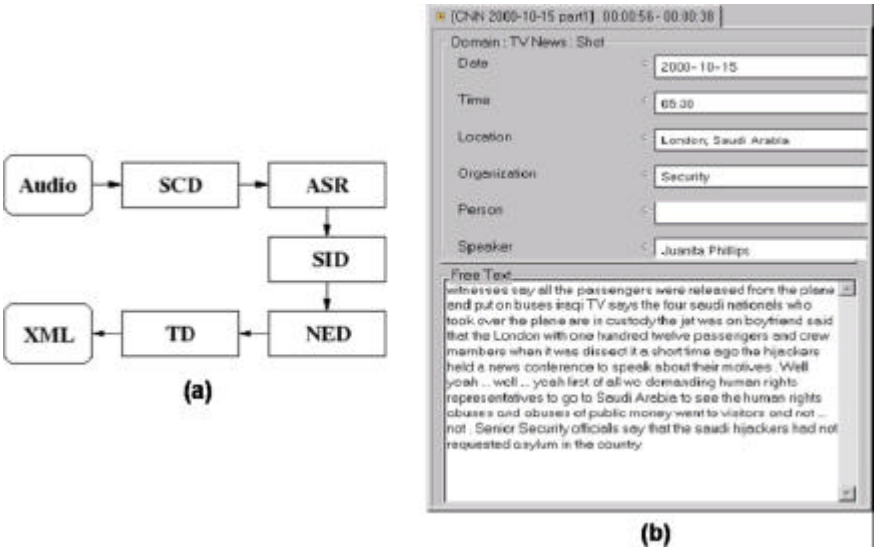

Figure 36. Exemple d'indexation automatique, a) le schéma de conversion d'une audio en documents XML. Ses composants sont : SCD (Speaker Change Detection), ASR (Automatic Speech Recognition), SID (Speaker Identification), NED (Named Entity Detection), et TD (Topic Detection), b) texte reconnu : entité détectée et reconnaissance de parole.

DiVAN [Tirakis et al. 99] est un projet d'archivage et diffusion de vidéo. Le projet vise à réaliser l'interconnexion de trois grandes institutions européennes d'archivage vidéo : l'INA, la RAI et le ERT. Le projet non seulement a l'objectif d'indexer automatiquement la vidéo selon un haut niveau sémantique comme l'unité de l'histoire, mais aussi fournit une grande souplesse d'indexation sémantique grâce à un composant de gestion des modèles. En effet, il y a différents genres de vidéo comme un journal télévisé, un magazine de cinéma, une comédie

ou une fiction, etc. Ces catégories de vidéo ont différentes logiques de structure de présentation. Le composant de gestion des modèles permet de gérer différents modèles correspondant à ces structures logiques, et il permet aussi d'ajouter facilement de nouveaux modèles pour de nouveaux types de vidéo.

Ce type d'application est très intéressant pour un rédacteur de documents multimédias. En effet ce dernier a souvent besoin d'un outil pour consulter le contenu de la vidéo avant de l'ajouter dans le document. L'outil de gestion et de présentation vidéo permet de rechercher, d'analyser et de naviguer efficacement dans le contenu des vidéos.

IV.2.1.3 Liens entre des médias

MAVIS [Lewis 96] est une autre approche intéressante d'indexation multimédia. Il permet non seulement de récupérer des médias en se basant sur le contenu, mais aussi de naviguer dans la base de multimédias à travers des liens génériques. Un lien générique ne contient que des caractéristiques du contenu de média source dans l'ancre source au lieu de contenir la localisation d'un média source. Lorsqu'un lien est édité, le média source ou le segment média source sélectionné est analysé pour extraire ses caractéristiques. Ce lien peut être appliqué à n'importe quel média source qui a des caractéristiques semblables aux caractéristiques génériques du lien. Les liens génériques sont sauvegardés dans une base de données séparée des médias, et ils sont triés pour augmenter les performances lors de la recherche.

Les liens génériques peuvent être activés de la façon suivante : en visualisant un média (image, vidéo et même audio) l'utilisateur de MAVIS peut en sélectionner n'importe quel segment pour suivre les hyperliens associés et accéder aux segments de même caractéristique. Dès que la demande est envoyée, le contenu du segment de média sélectionné est analysé pour trouver des caractéristiques clés à rechercher. Si des caractéristiques semblables aux caractéristiques clés sont trouvées dans la base des liens génériques, une liste des destinataires des liens génériques correspondants est alors présentée à l'utilisateur.

Une base de données multimédias indexée par cette approche est très utile pour les applications d'édition de documents multimédias. Par une requête, le rédacteur d'un document peut récupérer un ensemble de médias pertinents à composer, de plus, en se basant sur les liens entre les médias récupérés, les synchronisations entre ces médias sont facilement spécifiées, ou même on pourrait envisager de développer un outil de génération automatique d'un scénario de synchronisation entre ces médias.

IV.2.1.4 Synthèse

Une application d'indexation multimédia présente les capacités suivantes :

- ♦ gérer une base de données multimédias,
- ♦ analyser automatiquement et en temps réel des médias,
- ♦ reconnaître automatiquement et en temps réel des caractéristiques et des éléments sémantiques dans le contenu d'un média,
- ♦ classifier automatiquement et en temps réel des médias et des segments médias selon des catégories ou des sujets sémantiques,
- ♦ créer des liens entre des médias,

- ♦ accéder rapidement et efficacement à la fois aux médias d'une base de données multimédias et au contenu d'un média,
- ♦ localiser rapidement, précisément et efficacement un média et un segment média.

Avec ces capacités, un outil d'édition de documents multimédias peut créer facilement des documents complexes et sophistiqués. En effet, des médias pertinents à intégrer sont récupérés facilement. Les éléments sémantiques dans le contenu d'un média sont facilement localisés pour le synchroniser finement avec d'autres médias. La capacité de reconnaissance basée sur le contenu permet de spécifier des synchronisations et des hyperliens génériques qui sont très utiles dans des applications temps réel.

Par exemple, soit un document multimédia contenant un flux réel d'une vidéo, comme dans une application de vidéo-conférence. Un rédacteur n'a aucun moyen pour définir des synchronisations avec ce flux vidéo. Avec les possibilités offertes par un système d'indexation multimédia, des synchronisations ou des hyperliens génériques peuvent être spécifiées comme lancer une musique lors de l'apparition d'un objet ou d'un personnage. En utilisant les capacités d'analyse et de reconnaissance en temps réel de l'indexation, le système de présentation sera capable d'instancier ces spécifications sur les objets de la vidéo qui présentent des caractéristiques similaires aux objets visuels.

En pratique, ce type d'édition n'est pas encore opérationnel pour plusieurs raisons, dont les principales sont :

- ♦ la performance,
- ♦ la qualité des systèmes d'analyse,
- ♦ la différence entre les modèles des applications indexation et les applications multimédias.

Donc, les utilisations actuelles des systèmes d'indexation dans les applications de document multimédia sont réduites à l'accès à des médias individuels.

IV.2.2 Production de média

Des applications de production média comme *GoLive, Adobe Premiere* de Adobe, *MediaStudio Pro, VideoStudio, DVD Movie Factory* de *Ulead* ont les caractéristiques principales suivantes :

- ♦ Assembler différents clips de médias,
- ♦ Capturer facilement les flux vidéo et audio,
- ♦ Décoder et encoder des médias,
- ♦ Convertir un format d'encodage de médias dans un autre format,
- ♦ Analyser le contenu de média pour extraire des éléments de structure comme la détection de changement de scène,
- ♦ Traiter les médias comme les couper ou en changer la qualité (par des filtres), faire des effets, animations et translations sur des éléments de média
- ♦ Exporter en plusieurs formats de diffusion.

En général, une application de production média répond aux besoins de traitements des médias individuels pour créer des médias plus jolis. Cependant, une application de ce type reste limitée dans les aspects suivants :

Bien que l'outil de production média permette d'analyser et traiter des médias individuels, ces analyses et traitements sont de bas niveau. Il ne supporte pas l'analyse au niveau de l'interprétation des éléments sémantiques comme dans les systèmes d'indexation multimédia. Par exemple, il peut détecter le changement des scènes mais il ne peut pas détecter des unités logiques (*story unit*) ; de même il peut détecter un contour d'une région, mais il ne peut pas déterminer la sémantique de la région ; etc. C'est pourquoi l'édition d'un média reste de bas niveau : l'auteur ne peut pas exprimer des spécifications générales comme sélectionner des segments vidéo contenant l'apparition d'un personnage donné, ou bien filtrer toutes les scènes violentes, etc. L'auteur doit donc chercher manuellement les segments ou les régions intéressantes pour les ajouter dans le nouveau média ou les enlever de celui-ci.

Un outil de production média n'est pas attaché à une base multimédia. C'est pourquoi les ressources disponibles à éditer sont limitées à l'ensemble des médias collectés manuellement. Le système peut ajouter de nouveaux médias, mais il n'offre aucune fonction de recherche parmi un ensemble de médias.

La structure du nouveau média créé est simple et souvent spécifiée sous forme de placements absolus des clips médias dans le temps. Cependant, on peut noter que le système MAD [Friedlander et al. 96] est une exception dans cette famille de systèmes. Il fournit une vue du scénario à travers laquelle l'auteur peut visualiser et éditer la structure logique d'un scénario de média. Cependant le résultat final consiste toujours en un média physique linéaire, c'est-à-dire caractérisé par un seul flux temporel.

Les outils de production média utilisent la vue temporelle (*timeline view*) pour visualiser et éditer le scénario temporel du média. Cependant, dans la plupart des cas, leur vue temporelle est simple. Elle représente des clips médias dans des différents canaux et selon une structure plate, non hiérarchique. Il n'y a aucune relation temporelle entre les canaux des clips médias. De plus, la planification temporelle des clips médias est spécifiée selon des placements absolus.

Enfin, bien que la production de média puisse créer des présentations riches, ces présentations ont une limitation importante, car elles ne sont pas interactives.

En résumé, les outils de production de média ont la capacité d'analyser le contenu des médias, mais cette analyse n'est pas aussi complète que celle des applications d'indexation multimédia. Les outils de production de média ont la capacité d'intégrer un ensemble de clips médias, mais cette intégration est limitée à des scénarios linéaires et aux présentations passives. Par contre, le point fort des outils de ce type est leur capacité de traitement du contenu des médias pour obtenir des effets d'animation ou de transition, ou pour faire le recodage et transcodage de formats.

IV.2.3 Environnement auteur d'intégration de document multimédia

Les systèmes d'intégration de documents multimédias sont nombreux et variés. On peut citer *RealSlideShow*, *PowerPoint*, *GRiNS*, *Director*, *Madeus*. Les différences entre ces systèmes sont dues à la fois au modèle de document qu'ils utilisent et à leur approche d'édition et de visualisation du scénario de présentation multimédia. En général une application d'intégration de document multimédia peut fournir des présentations multimédias ayant des capacités d'interaction et de changement dynamique de scénario de présentation (cf. Chapitre II). Elle possède des outils permettant d'éditer des scénarios riches de présentation multimédia. Dans cette partie nous analysons seulement quelques environnements typiques pour en identifier les limitations générales. Des présentations et comparaisons plus complètes de différents environnements auteur sont données dans [Tardif 00] et [Navarros 01].

IV.2.3.1 GR*i*NS

GR*i*NS [GR*i*NS] est un éditeur de documents SMIL. Il fournit les vues qui permettent d'éditer facilement des documents dans tous les axes de spécification : le temps, l'espace, les hyperliens, les attributs, les effets d'animation et de transition, etc. Toutes ces vues sont synchronisées. La vue la plus intéressante du système est la vue temporelle hiérarchique (cf. la figure37a). Cette vue permet de visualiser très facilement le scénario temporel du document (spécifié selon le modèle de SMIL). Par cette vue, l'auteur peut aussi percevoir très facilement le contenu du document grâce à des icônes représentant le contenu des médias. L'édition dans cette vue est aussi très conviviale. L'auteur peut prendre et déposer un nouveau contenu directement dans la vue. La modification des informations temporelles d'un objet média peut s'effectuer facilement par déplacement et retaillage de la boîte représentant l'objet.

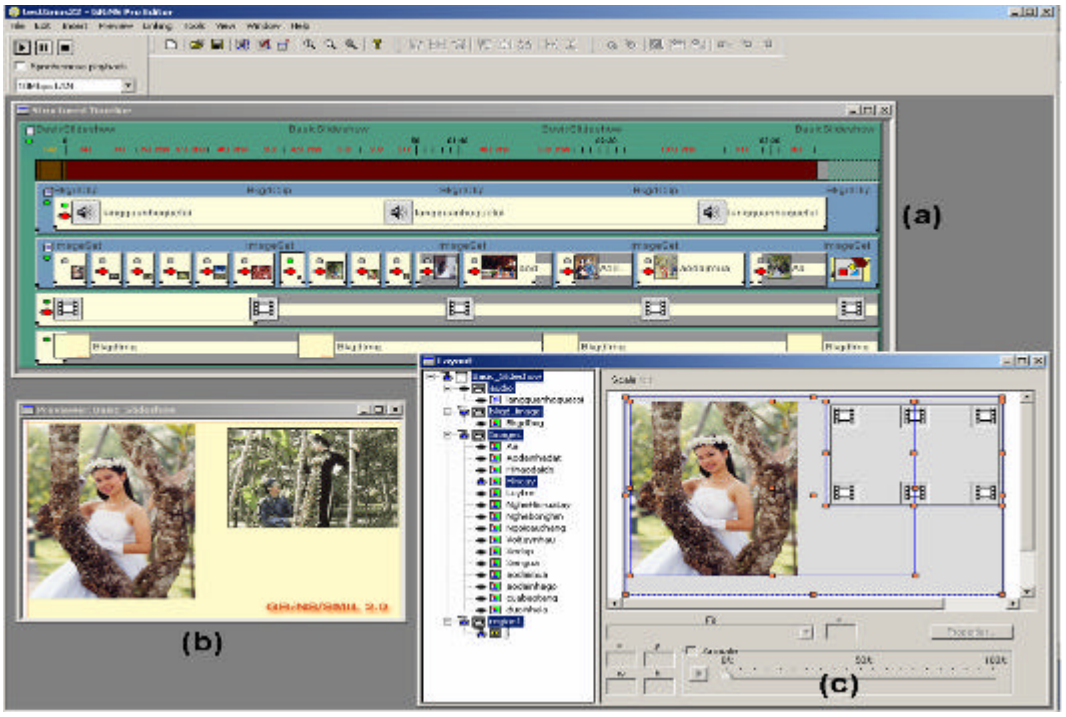

Figure 37. Éditeur *GRiNS*, a) Vue temporelle hiérarchique, b) Vue de présentation, c) Vue des régions et de sa structure.

GR*i*NS fournit les vues qui permettent de parfaitement créer des documents multimédias SMIL à partir un ensemble de médias, néanmoins il n'est adapté qu'à la composition de scénario à gros grain. En effet, il est difficile avec cet outil de composer des synchronisations sophistiquées ou fines dans les documents. De plus l'éditeur GRiNS ne fournit aucun moyen pour la recherche ou le traitement des médias.

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

IV.2.3.2 PowerPoint

Nous envisageons maintenant l'application PowerPoint, un outil largement connu pour éditer des transparents utilisés comme support à des présentations orales. Partant d'une approche de modèles prédéfinis elle correspond parfaitement au besoin de simplicité d'utilisation pour créer des transparents. Le scénario global d'un document PowerPoint est limité à une présentation séquentielle de transparents, mais sur chaque transparent, PowerPoint permet d'éditer richement un ensemble de médias. L'auteur peut intégrer un grand éventail de types d'objets médias dans un transparent. Les médias intégrés dans un transparent peuvent être organisés selon les modèles prédéfinis (voir la Figure 38c), ou peuvent être arrangés librement selon des placements effectués de l'auteur. Des présentations

temporelles des médias dans un transparent peuvent être animées et planifiées selon différents enchaînements au choix : commencer sur un clic, commencer en même temps avec le transparent précédent et commencer après le transparent précédent avec un délai. L'outil supporte aussi un nombre important d'effets prédéfinis de transitions en entrée ou en sortie d'un transparent.

De plus PowerPoint fournit de nombreuses fonctions lors de l'édition, les plus intéressantes d'entre elles sont :

- Le lien avec une base de clips image indexée automatiquement : cela permet à l'auteur de trouver facilement un média souhaité à intégrer dans la présentation. Des requêtes sont simplement des mots. La base est créée par l'assemblage d'une collection d'images (*clips arts*) fournies par Microsoft Office et les images existant sur l'ordinateur de l'utilisateur, puis cette base est indexée automatiquement.
- L'intégration de plusieurs outils de production de médias dans l'environnement auteur comme un outil d'édition de courbes statistiques (*Chart*), ou un outil d'édition des images (*Image PainBrush*, *Image bitmap*).
- L'intégration de fonctions de traitement des médias de la présentation, par exemple la modification d'attributs des images (luminosité, contraste, …) ou le formatage de mots de façon artistique (*word Art*).

En conclusion, cet outil répond aux différents critères identifiés au début du chapitre, mais pour une seule classe de documents : des transparents. Cette limitation vient de son approche à base de modèles prédéfinis qui permet une mise en œuvre plus simple et une utilisation facile. Cependant cette application montre l'intérêt de l'utilisation des bases multimédias indexées et des outils de production de média dans les environnements d'intégration et de composition multimédia.

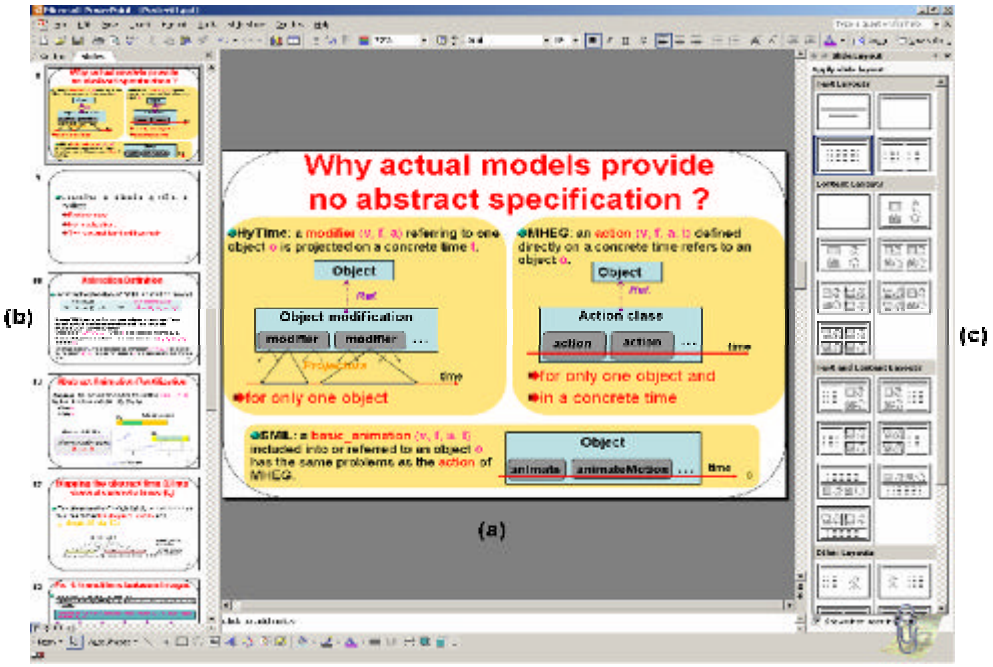

Figure 38. Environnement de PowerPoint avec a) un transparent édité selon l'approche WYSIWYG, b) la structure séquentielle des transparents, c) le jeu de transparents prédéfinis.

IV.2.3.3 Madeus

Nous terminons cette étude par la présentation de l'environnement auteur Madeus, éditeur développé au sein du projet Opéra. Nous avons choisi de présenter Madeus ici, parce que nos contributions présentées dans les chapitres suivants seront expérimentées à partir de ce système. En fait, Madeus est utilisé comme un environnement d'expérimentation modulaire et extensible dans l'équipe Opéra [Villard 02]. Madeus est basé sur Kaomi, une boîte à outils issue aussi des travaux du projet Opéra. Cette boîte met en oeuvre des principes d'édition (environnement multivues, édition directe, ...) très appropriés pour développer des applications d'édition multimédia. Différents éditeurs ont été réalisés au-dessus de cette boîte à outils : SMIL Editeur, MHML Editeur, Workflow Editeur et Madeus [Tardif 00] [Navarros 01].

Madeus est un éditeur dont le modèle de document est Madeus qui a été présenté dans la section III.4.8. Comme la plupart des outils auteur, Madeus fournit des fonctions d'édition à travers les vues principales suivantes :

- La vue présentation et édition spatiale permet de jouer et d'éditer le document. Elle s'appuie sur un service de préchargement des médias pour assurer une bonne qualité de présentation du document (respect des synchronisations). Cette vue offre également des fonctions d'exécution évoluées, permettant par exemple à l'auteur de demander l'exécution du document à un instant t, ou au début d'un objet. De plus cette vue est fortement synchronisée avec la vue temporelle.
- La vue temporelle hiérarchique permet de visualiser la structure des objets. L'auteur peut directement éditer le placement temporel des objets dans cette

vue par des manipulations directes : décalage à gauche et à droite, allongement ou réduction de la durée. Lors de chaque opération d'édition, Madeus assure le maintien en continu de toutes les relations que l'auteur a déjà exprimées.

- La vue structure hiérarchique permet de visualiser et d'éditer les différentes structures du document.

Les intérêts du système sont non seulement la visualisation et l'édition directe dans la structure temporelle et la structure spatiale du document, mais aussi l'édition des relations spatiales et temporelles. Madeus assure le formatage et la vérification de la cohérence tout au long du processus d'édition. Ce mode de spécification par relations fournit à l'auteur plus de confort que l'édition absolue.

IV.2.3.4 Synthèse des environnements d'intégration de document multimédia

Les outils auteur de document multimédia constituent des environnements qui permettent d'intégrer un nombre important de types de médias dans une présentation multimédia. Ils fournissent aussi des outils appropriés pour éditer des scénarios riches de document (structure hiérarchique, hyperliens, interaction, animation, transition, etc.). Néanmoins, pour être plus intéressant pour l'utilisateur, ces environnements doivent offrir de nouvelles capacités comme :

- la recherche directe des médias appropriés à composer,
- le traitement direct du contenu des médias,
- la synchronisation fine sans utilisation de compositions absolues.

L'outil Madeus présente les mêmes limites identifiées dans les systèmes de composition multimédia précédents (pas d'intégration fine de média, pas d'accès par le contenu) et nous verrons dans le chapitre VI. quelles extensions nous y avons apportées pour répondre aux objectifs de notre thèse.

IV.3 Synthèse

Nous avons envisagé dans ce chapitre trois catégories d'applications multimédias : l'indexation, la production et l'intégration de médias. L'étude est basée sur l'objectif d'obtenir un système d'intégration de document multimédia idéal. Aucune des catégories de système multimédia envisagées ci-dessus ne prend en compte tous les critères de ce système idéal. Mais l'ensemble de ces critères du système idéal peut se trouver en partie dans chacune des catégories de système multimédia. Malheureusement, les systèmes multimédias ne peuvent pas être couplés pour assembler les points forts de chaque catégorie. La Figure 39 présente synthétiquement les situations des applications multimédias étudiées dans ce chapitre. L'axe vertical présente les capacités d'analyse des médias, ces capacités d'analyse progressent à partir de l'analyse des primitives du contenu de média jusqu'à l'interprétation de haut niveau sémantique du contenu de média. L'axe horizontal présente la capacité d'intégration, la progression de cet axe représente l'augmentation de capacité d'intégration des médias selon le nombre de types médias et la richesse des scénarios.

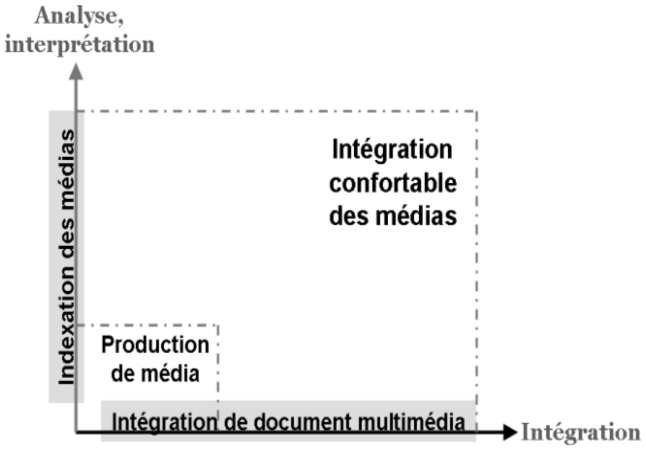

Figure 39. Schéma synthèse de situation des applications multimédias

L'application d'indexation peut collecter des médias et organiser une base de données multimédia par l'analyse et puis l'interprétation des caractéristiques de la structure. Ce type d'application permet de récupérer des médias, mais ne peut pas donner une présentation multimédia ou produire un nouveau média à partir des médias trouvés.

L'application d'intégration de document multimédia actuellement ne permet que d'assembler des médias dans un scénario de présentation. Bien que la structure globale de ce scénario puisse être riche, elle est encore grossière parce que la composition fine n'est pas supportée. De plus, la recherche, l'analyse et le traitement de média sont manquants dans ce type d'application.

La troisième catégorie, la production de média, se situe au milieu de deux précédentes catégories. Elle permet d'analyser le contenu de média mais ne permet pas d'en interpréter les caractéristiques sémantiques du contenu. Elle permet aussi d'intégrer des médias, mais l'éventail des types de média supportés n'est pas large, et le scénario qui peut être édité n'est que linéaire. En fait, on peut dire que ce type d'application est une faible intégration de deux applications d'indexation et d'intégration de documents multimédias.

Dans ce schéma, nous avons aussi placé notre objectif d'obtenir un outil d'intégration des médias plus confortable qui puisse intégrer aux maximum les capacités de l'analyse, de l'interprétation, du traitement et de l'intégration des médias.

Ainsi, l'objectif de ce travail est d'expérimenter un environnement auteur plus confortable pour l'auteur qui consiste à enchaîner les trois types d'applications multimédias. Cette approche évite d'intégrer profondément différents outils d'applications multimédias dans un environnement comme PowerPoint. Une telle intégration peut rendre l'environnement lourd et complexe. Notre enchaînement entre des outils d'application multimédia fonctionne comme suit (voir la Figure 40) :

- L'outil d'indexation donnera des descriptions du contenu de média à la partie d'intégration,
- L'outil d'intégration utilisera ces descriptions qui permettront de composer plus finement les médias entre eux,

♦ Le document résultat de la partie d'intégration peut être utilisé par l'outil de production pour produire des présentations dans différents formats, par exemple SMIL, XHTML+SMIL, SVG, ou même des formats de vidéo comme MPEG-4.

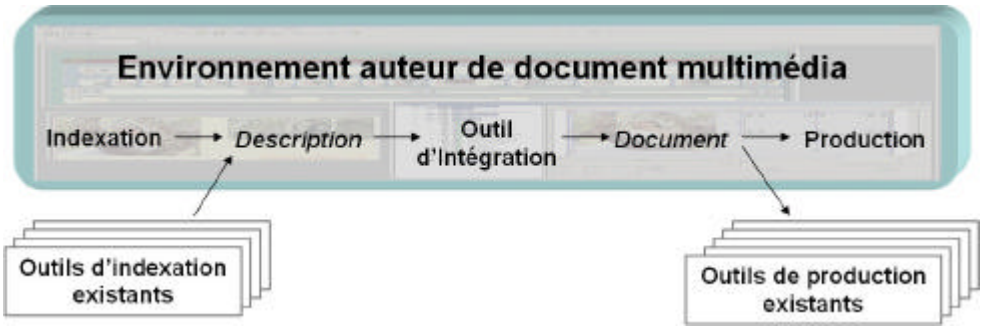

Figure 40. Architecture de l'environnement auteur d'intégration confortable

Notre approche d'enchaînement est donc basée sur l'échange des données entre les parties. Une telle approche est flexible, mais nécessite une modélisation des données riche et la plus indépendante possible des applications. De plus l'environnement peut utiliser des outils d'indexation et de production existantes, si ces outils utilisent des modèles de données compatibles avec l'environnement auteur.

Le cœur de notre travail se compose donc de deux parties :

♦ Partie modèle : Créer un modèle de description du contenu des médias et un modèle de document multimédia adaptant le modèle de description du contenu des médias à la spécification des compositions fines,

♦ Partie application : Créer l'environnement auteur adoptant ces modèles.

À noter que, si notre modèle de description du contenu des médias est compatible avec des standards de description comme MPEG-7, notre environnement peut utiliser directement le résultat d'indexation des systèmes d'indexation existants qui sont compatibles avec ce standard. De même, la transformation basée sur des documents XML est maintenant très développée, alors si le document multimédia est en format XML, on peut profiter des processeurs de transformation existants pour transformer notre document résultat en différents formats de présentation comme SMIL, XHTML+SMIL ou même MPEG-4.

Dans les chapitres suivants, nous présentons en détail ces modèles et l'environnement auteur construit à partir de ces modèles.

Chapitre V. Modèles de description du contenu des médias et de leur intégration dans les documents

V.1 Introduction

Le Chapitre I. a donné un état de l'art de la modélisation multimédia. Bien que la modélisation multimédia ait atteint un niveau très riche et développé, il existe une lacune entre le modèle de description du contenu de média et le modèle de document multimédia. La **modélisation du contenu** des médias permet de consommer plus efficacement le contenu multimédia tandis que **la modélisation du document multimédia** vise à faciliter l'édition et la présentation des documents multimédias. Malheureusement le manque de liens entre eux induit des limitations dans les systèmes existants : l'auteur utilisant un environnement d'édition ne peut pas accéder finement au contenu d'un média pour une synchronisation fine ou une composition sophistiquée ; par ailleurs, le système de requête des informations multimédias ne peut pas renvoyer comme résultat de recherche une présentation, mais seulement des médias individuels.

Les contributions présentées dans ce chapitre proposent un modèle de document multimédia en harmonie avec les modèles de description du contenu des médias afin de diminuer la distance entre eux, i.e., les descriptions du contenu de média doivent faciliter l'accès au contenu des médias pour offrir des informations servant à composer des documents sophistiqués ; à l'inverse, le modèle de document multimédia doit être étendu pour pouvoir utiliser les descriptions.

Nous allons présenter premièrement dans ce chapitre des modèles de description du contenu des médias. Ces modèles sont ensuite représentés à l'aide du standard MPEG-7 pour pouvoir adapter plus largement des résultats de description déjà existants. Dans un second temps, nous présentons une extension du modèle Madeus permettant d'utiliser ces descriptions. Cette extension supporte par ailleurs un modèle d'animation abstrait qui sera présenté dans la quatrième partie. Enfin, dans la partie de synthèse, nous ferons une évaluation globale de ces contributions.

V.2 Modèles de description du contenu des medias

Le but est de spécifier plus sémantiquement des portions de média pour composer des synchronisations fines (cf. III.4.4, III.4.5, III.4.6). Dans ce but, les modèles de description du contenu des médias adaptés à la composition de document multimédia doivent satisfaire les critères suivants :

- ♦ Le modèle doit permettre de décrire en détail la structure logique du contenu des médias, afin de localiser facilement grâce à cette structure des portions du contenu de média.
- ♦ La plupart des structures du contenu des médias sont hiérarchiques, le modèle doit donc permettre de les décrire.
- ♦ La composition dans le document multimédia est basée principalement sur des informations spatiales et temporelles (l'apparition d'un personnage ou le début d'un plan vidéo), le modèle doit donc permettre de décrire ces informations.
- ♦ Les synchronisations sémantiques, par exemple *afficher le texte "Simba" au-dessus du personnage Simba chaque fois celui-ci apparaît dans la vidéo*, sont nécessaires en composition de documents multimédias, le modèle doit donc permettre de décrire des classifications sémantiques comme le personnage *Simba*.
- ♦ des synchronisations génériques, par exemple *lancer une musique quand un objet rouge apparaît*, sont nécessaires pour définir des synchronisations avec des flux média non-déterminés comme la vidéo diffusée en "live" (*temps réel*). Le modèle doit donc permettre de décrire les caractéristiques des éléments (*objet rouge*) du contenu et les modèles de ces caractéristiques.

Un nombre important de travaux a été consacré à la description du contenu de ce média (cf. III.3) et parmi eux, l'apparition de la norme MPEG-7 (III.3.1.3) est la plus importante. MPEG-7 prend en compte des modèles de description existants pour fournir des outils standard de modélisation du contenu des médias sous forme des schémas XML (*XML Schemas*). Nous avons choisi d'utiliser les schémas standard fournis par MPEG-7 pour décrire notre modèle, afin de viser à un large éventail d'applications (nos descriptions peuvent être utilisées par différentes applications et à l'inverse, notre application peut aussi récupérer facilement des descriptions externes). Néanmoins, les schémas de MPEG-7 sont à la fois complexes et génériques, ils doivent être restreints et étendus pour adopter le contexte particulier de la rédaction et de la présentation de document.

Nous avons travaillé sur ce sujet il y a deux ans et avons proposé un modèle pour la vidéo compatible avec notre modèle de document multimédia [Roisin et al. 99] [Roisin et al. 00]. Nous avons commencé par la modélisation de la vidéo, média le plus riche. Les premiers résultats concernant l'ouverture du contenu de la vidéo à la composition ont démontré l'intérêt du travail et nous ont encouragé à étendre l'approche aux autres médias. Nous avons donc tenté de modéliser les trois autres types de média : l'audio, l'image et le texte non structuré.

Dans la suite de la section nous décrivons notre modèle de description du contenu des médias de la manière suivante : nous présentons premièrement

l'approche générale de la description ; dans un deuxième temps nous discutons plus en détail chaque structure de description en nous concentrant sur la structure du contenu qui sert de base à la composition du document multimédia ; nous discutons ensuite de la représentation des modèles à partir des schémas standards de MPEG-7 ; la conclusion et l'évaluation de nos modèles sont données à la fin de la section.

V.2.1 Structure générale de la modélisation

La structure globale des modèles s'appuie sur trois aspects principaux de description : les *métadonnées*, la *gestion du contenu* et la *description du contenu*.

- Les *métadonnées* permettent de décrire des informations concernant la description elle-même, comme l'auteur de la description, la date de création de la description, le langage de description, les notes de description, etc.
- La *gestion du contenu* permet de décrire des informations concernant :
  a. le média lui-même (l'identificateur et la localisation, le format physique, la qualité du contenu),
  b. l'utilisation du contenu (le droit, le moyen, la disponibilité, la trace de l'utilisation du contenu),
  c. l'utilisateur du contenu (le nom et les références de l'utilisateur),
  d. la création du contenu (la date de création, le lieu de création, le nom et les coordonnées de créateur, les outils de création, les matériels liés, etc.),
- La *description du contenu*. L'information du contenu de média peut être représentée en plusieurs niveaux : les informations *physiques* constituées du flux binaire (ou textuelle) du contenu de média qui n'est utilisable que par l'ordinateur et les informations de *description* qui permettent de transformer les informations physiques en connaissances exploitables par l'utilisateur, ce qui permet de renforcer "l'interface" entre l'homme et la machine et permet d'exploiter donc plus facilement le contenu de média. Pour fournir plusieurs niveaux d'abstraction en exploitation du contenu de média, nous proposons un modèle à trois niveaux : *concret*, *significatif* et *catégoriel* (*structure*, *sémantique* et *thésaurus*, voir la Figure 41) :
  a. Le niveau *concret* est la description de la **structure** du contenu. La modélisation de la structure du contenu est la partie la plus importante dans notre modèle. Elle permet de décrire l'organisation narrative qui peut représenter directement et complètement l'information du contenu. Cette modélisation est souvent basée sur la structure logique classique de média (la *séquence*, la *scène*, le *plan* pour la vidéo ; la *scène*, l'*arrière-plan*, l'*objet*, la *région* pour l'image ; le *chapitre*, la *section*, le *paragraphe*, la *phrase*, l'*expression*, le *mot*, le *caractère* pour le texte ; etc.). Les descriptions de ce niveau sont indexées directement sur le flux *physique* de média. Elles permettent à l'auteur de spécifier des synchronisations avec chaque représentation concrète du contenu. Par ailleurs, nous proposons encore un modèle de description du résumé du contenu pour aider le lecteur à superviser rapidement le contenu. L'auteur

peut utiliser cette description pour intégrer le résumé de média dans un document.

b. Le niveau *significatif* est une abstraction plus haute qui permet de lier des descriptions de bas niveau précédentes avec le monde réel. Par exemple, un ensemble d'objets de la vidéo peut correspondre à un personnage, un endroit ou une chose concrète dans le monde réel ; ou la relation conceptuelle entre eux comme le personnage *A* est le responsable du personnage *B* ; ou *A* est l'époux de *B*. Nous proposons de créer dans ce niveau un **modèle sémantique** qui permet de décrire la signification des descriptions au niveau de la structure. En fait, il permet de déclarer des termes ou des relations conceptuelles pour représenter un groupe d'occurrences décrites dans la partie structure. Ce modèle sémantique permet de spécifier des synchronisations plus générales et plus concises. Par exemple le fait de spécifier un lien sur le personnage *Tien* dans sa vidéo de mariage est beaucoup plus simple que la spécification de plusieurs liens sur chacune de ses apparitions (voir la Figure 41). La description de cette partie est indexée sur la partie de description de la structure.

c. Au plus haut niveau, on propose de créer un **modèle de thésaurus** pour permettre de catégoriser des connaissances représentées dans la partie sémantique. Par exemple le terme *Doctorant* peut représenter la catégorie professionnelle de deux personnages *Hang* et *Tien* (voir la Figure 41). Le modèle permet aussi de référencer des connaissances dans un thésaurus existant.

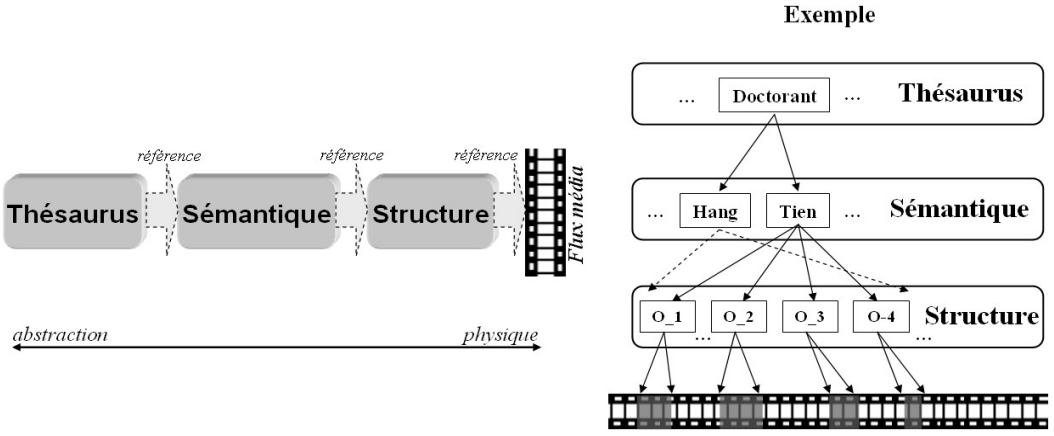

Figure 41. Modèle et l'exemple de description du contenu de média.

Nous présentons ci-dessus notre approche pour la structure globale du modèle. Cette structure globale couvre plusieurs aspects de la description à partir des descriptions du contenu jusqu'à les descriptions de la gestion du contenu.

La description de la gestion du contenu est le moyen le plus traditionnel pour indexer les médias, mais cette indexation n'offre qu'un accès global au média. Notre objectif vise un accès fin des médias donc ce niveau de description n'est pas prioritaire. De plus, ce type de description est déjà bien modélisé par les travaux existants comme le standard Dublin Core ou même la norme MPEG-7 (voir les

sections III.3.1.1 et III.3.1.3). Nous décidons donc appliquer directement ces standards (voir la section V.2.4.1) et ne pas les décrire plus en détail ici.

Par contre, la description du contenu est une approche plus récente pour indexer plus finement dans le contenu des médias. Elle est donc très importante pour nos besoins de composition fine de document multimédia. Cependant, la modélisation complète des trois niveaux (*structure, sémantique* et *thésaurus*) de la description du contenu des médias demande un grand effort que dans un premier temps de notre travail nous ne pouvons pas faire. Nous avons préféré nous concentrer sur la modélisation de la **structure du contenu** ce qui est la plus prioritaire pour les besoins de composition multimédia. Nous présentons cependant mais de la façon plus rapide les idées générales de ces modèles puisqu'ils permettent de spécifier des synchronisations plus génériques et condensées.

La suite de la section est organisée de la manière suivante : nous discutons rapidement des modélisations thésaurus et sémantiques ; on se concentrera principalement ensuite sur les modèles de description de la structure du contenu des média ; enfin nous présentons la représentation de ces modèles en se basant sur les schémas de MPEG-7.

V.2.2 Les modèles de Thésaurus et de Sémantique

Comme on l'a vu, la modélisation fournit la capacité de définir des termes significatifs pour représenter une description ou un ensemble des descriptions de plus bas niveau.

La partie thésaurus peut référencer des thésaurus existants pour bénéficier de lexiques standard, par exemple *WordNet*[14], une base lexicale pour la langue anglaise ; ou le travail présenté dans [Carrive 00] pour classifier des plans d'un journal télévisé.

La norme MPEG-7 fournit des modèles et des schémas très complets pour décrire ces aspects thésaurus et sémantique. Nous décrivons le principe d'utilisation de ces schémas pour définir les termes significatifs et des relations conceptuelles dans la partie de l'implémentation des modèles, la section V.2.4.1.

V.2.3 Modèle de la structure du contenu des médias

Pour faciliter à l'accès au contenu d'un média, l'application d'édition de document multimédia demande des descriptions claires et riches de la structure narrative et logique du contenu. Les modèles de média (vidéo, audio, image et texte) partagent la même structure générale ci-dessus. Par contre, les logiques de structure du contenu étant très fortement dépendantes de chaque type de média, chacun aura son type média et donc un propre modèle de structure de son contenu.

Dans cette section, nous présentons les modèles des structures narratives/logiques. Nous nous concentrons principalement sur le modèle de la vidéo, puis nous décrivons rapidement les modèles d'autres médias.

---

[14] WordNet a lexical database for the English language, http://www.cogsci.princeton.edu/~wn/.

V.2.3.1 Modèle de la structure du contenu de la vidéo

La vidéo est le média qui contient le plus d'information, et sa structure narrative est la plus riche par rapport aux autres médias. On peut trouver de nombreux modèles descriptifs de la structure de ce média dans littérature (cf. III.3). Néanmoins ces modèles sont souvent dépendants des approches d'indexation (basés sur les caractéristiques du contenu, la sémantique du contenu ou la structure narrative du contenu). Dans cette section, nous choisissons une approche d'assemblage de ces modèles pour décrire à la fois la structure logique, les objets et même les événements du contenu de la vidéo.

1. Structure logique de la vidéo

Notre modèle de décomposition du contenu de la vidéo en unités temporelles hiérarchiques (*Séquence*, *scène*, *plan et transition*) est semblable à la structure dramatique classique de la vidéo. Ce type de modèle peut se trouver aussi dans de nombreux travaux existants [Jacopo et al. 96] [Hammoud et al. 98] [Hunter 99] [Dumas et al. 00]. Un tel type de description permet à l'auteur d'exploiter facilement le contenu de la vidéo grâce aux unités logiques. Cependant ces travaux sont dédiés principalement à l'indexation, et de ce fait ne définissent pas de contraintes temporelles entre les éléments de la structure ce qui est très important. L'explicitation de cette structure temporelle intra-média est nécessaire pour assurer la cohérence de la structure pendant l'édition des descriptions et surtout pendant la composition multimédia. Précisément, la structure comprend les éléments suivants :

- le *plan* est défini comme une série d'images, du moment où la caméra commence à filmer jusqu'au moment elle s'arrête, sa durée est en générale de quelques secondes.
- la *scène* est une unité dramatique composée d'une série de plans, montrant l'action en un lieu et un temps narratif spécifiques.
- la *séquence* est l'unité dramatique la plus haute de la vidéo, composée de plusieurs scènes, liées ensemble par leur contenu émotif et narratif.
- la *transition* décrit le passage d'un élément (*séquence*, *scène* ou *plan*) à l'élément suivant. Une transition peut être de l'un des types suivants : *cut*, *fade-in*, *fade-out*, *dissolve*, *wipe* ou *warp*.

En général, les éléments présentés ci-dessus correspondent à des segments temporels dans la vidéo. Il est noté qu'un segment temporel de la vidéo représente une suite d'images de la vidéo. On leur associe donc les propriétés suivantes :

- Le début, la durée et la fin du segment pour représenter le segment vidéo dans le temps.
- Des caractéristiques visuelles comme l'histogramme de couleurs ou de texture des images d'un segment de la vidéo ; une image clé du segment ; etc.

Ces éléments temporels sont organisés selon une structure hiérarchique et sont liés par des relations temporelles (voir la Figure 42). Les segments similaires peuvent être représentés/regroupés par un élément dans la partie sémantique (*Cluster*, voir la Figure 43). On peut comparer ce regroupement à un index d'un livre où un terme de l'index peut être présent dans plusieurs parties du livre.

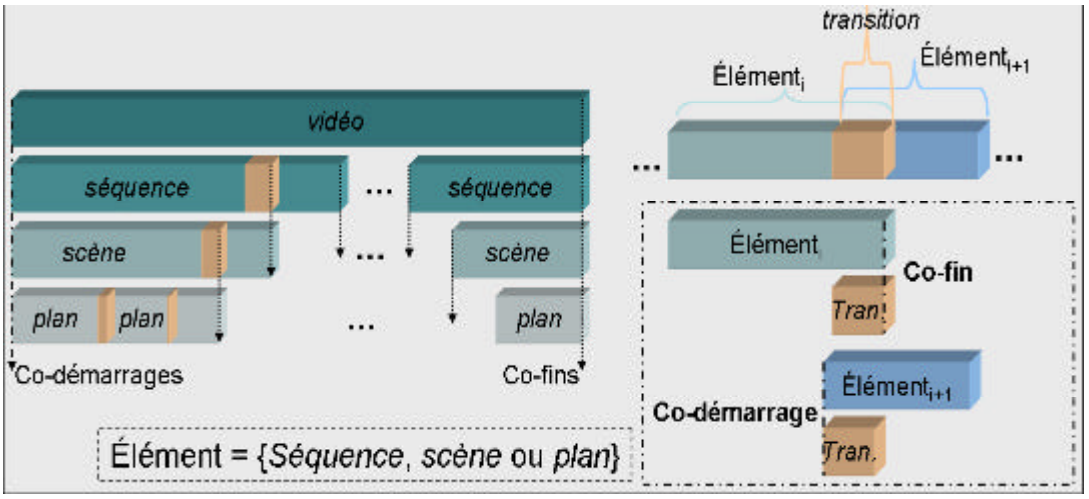

Figure 42. Structures hiérarchiques et relationnelles de la vidéo.

Structure hiérarchique

Les éléments sont organisés dans la structure hiérarchique comme suit : une vidéo contient en ensemble de *séquences* ; une *séquence* est composée d'éléments de scène; et une *scène* inclut des éléments de plan (voir la Figure 42a).

Relations temporelles entre les éléments de vidéo

Les relations temporelles entre les éléments relient les entités d'une vidéo dans l'ordre temporel défini par l'exécution de la vidéo. Nous proposons de modéliser ces relations temporelles en utilisant les relations d'Allen [Allen 83] de la manière suivante :

- les relations entre un premier plan P d'une scène S_1 et cette scène, entre une première scène S_2 d'une séquence Q_1 et cette séquence, et entre une première séquence Q_2 et la vidéo sont les relations de démarrage parallèle (*starts*) (voir la Figure 42a) :

$$\mathbf{Début}_P = \mathbf{Début}_{S1} ; \mathbf{Début}_{S2} = \mathbf{Début}_{Q1} ; \mathbf{Début}_{Q2} = \mathbf{Début}_{vidéo}$$

- les relations entre un dernier plan P d'une scène S_1 et cette scène ; entre une dernière scène S_2 d'une séquence Q_1 et cette séquence, entre une dernière séquence Q_2 et la vidéo sont les relations de fin parallèle (*finishes*) (voir la Figure 42a) :

$$\mathbf{Fin}_P = \mathbf{Fin}_{S1} ; \mathbf{Fin}_{S2} = \mathbf{Fin}_{Q1} ; \mathbf{Fin}_{Q2} = \mathbf{Fin}_{vidéo}$$

- la transition d'un élément (*séquence*, *scène* ou *plan*) avec l'élément suivant est modélisée par deux relations co-démarrage (*starts*) et co-fin (*finishes*) : l'élément avant E_a est en relation co-fin avec la transition T ; l'élément suivant E_s est en relation co-démarrage avec l'élément de transition (voir la Figure 42b) :

$$\mathbf{Fin}_{Ea} = \mathbf{Début}_T ; \mathbf{Fin}_T = \mathbf{Début}_{Es}$$

Un des objectifs de notre modèle est de faciliter des modifications de l'utilisateur. Ces relations temporelles permettent d'assurer la cohérence de la structure vidéo quand l'utilisateur réalise des opérations d'édition. Cependant, ces relations n'ont pas besoin d'être définies explicitement dans les descriptions des éléments vidéo. Par exemple, lorsque les descriptions du contenu de la vidéo sont représentées dans une vue temporelle, les relations peuvent être déduites directement à partir des informations temporelles et hiérarchiques des éléments décrits pour contraindre l'édition à travers la vue temporelle.

L'utilisation des relations d'Allen pour modéliser la structure de la vidéo se trouve aussi dans [Hammoud et al. 98] [Carrive et al. 00]. Cependant, dans ces modèles, les relations temporelles sont utilisées comme des *templates* pour détecter automatiquement des segments sémantiques (scène) de la vidéo.

Clusters

Les plans similaires peuvent être groupés et représentés par un élément sémantique dans la partie de description sémantique (voir la Figure 43).

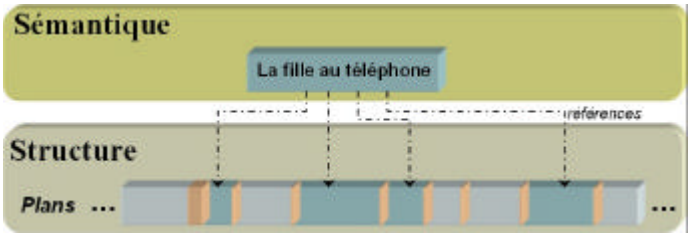

Figure 43. Exemple de groupement de segments similaires.

2. Structure logique d'un plan

Le plan est l'unité la plus petite dans la théorie classique du film. La détection des plans est toujours le premier travail effectué lors de l'analyse d'une vidéo. Ensuite les plans peuvent être utilisés comme des unités de base pour construire d'autres unités qui peuvent représenter plus sémantiquement le contenu de la vidéo. Néanmoins la description de ces composants de haut niveau permet seulement de spécifier des synchronisations à gros grain (avec des plans, des scènes ou des séquences) dans des documents multimédia. Cependant dans un plan il y a d'autres informations intéressantes, par exemple, des objets et des relations spatio-temporelles entre ces objets, ou des événements et des enchaînements de ces événements. Ces informations peuvent permettre de spécifier des synchronisations plus riches et à grain plus fin. Il est donc intéressant d'être capable de décrire le contenu des plans. Nous proposons donc qu'un plan soit principalement composé des éléments suivants (voir la Figure 44) :

- L'élément *segment* décrit une situation particulière dans la vidéo que l'on considère comme une partie intéressante : une explosion du moteur, un avion qui décolle, une démonstration, une tempête, etc. cet élément *segment* est similaire à la notion *événement* des modèles comme AEDI (voir la section III.3.2), mais dans notre cas, nous ne considérons que des suites d'images dans un plan. Il peut référencer un élément dans la partie sémantique pour y associer une signification du monde narratif réel.

- ♦ Un élément *occurrence* est un élément spatio-temporel pouvant décrire une personne ou un objet qui apparaît dans le plan. Par cette modélisation nous avons la même approche de description q u'en utilisant les caractéristiques de bas niveau (voir la section III.3.2.1.1). Par contre notre *occurrence* se concentre particulièrement sur les descriptions spatio-temporelles pour la composition multimédia. L'élément *occurrence* peut référencer un élément, par exemple un personnage, dans la partie sémantique qui représente cette personne ou cet objet.
- ♦ L'élément *Spatio-TemporalLayout* décrit la disposition/relation spatio-temporelle entre les occurrences d'un plan. Il référence un élément dans la partie sémantique pour décrire la signification réelle de la relation. On peut trouver un tel élément dans le modèle à base d'objets [Paek et al. 99a] (voir la section III.3.2.2.2), cependant notre élément ne concerne que les relations spatiales et surtout spatio-temporelles que ce dernier modèle ne supporte pas.

Au niveau hiérarchique, ces éléments sont contenus dans un plan (voir la Figure 44a). C'est pourquoi nous proposons d'utiliser la relation temporelle *à l'intérieur* qui contraint ces éléments à être toujours contenus dans un plan. Les relations entre les instants de début et fin de ces éléments sont (voir la Figure 44b) :

$$\mathbf{Début}_{élément} = \mathbf{Début}_{plan} + \mathbf{d}_1\ (\mathbf{d}_1 = \mathbf{0})\ ;\ \mathbf{Fin}_{plan} = \mathbf{Fin}_{élément} + \mathbf{d}_2\ (\mathbf{d}_2 = \mathbf{0})$$

Il n'y a aucune contrainte temporelle entre les éléments dans un plan. En effet les éléments du plan peuvent apparaître librement dans le plan et sont *a priori* indépendants les uns des autres. Leur disposition temporelle est donc quelconque à l'intérieur du plan.

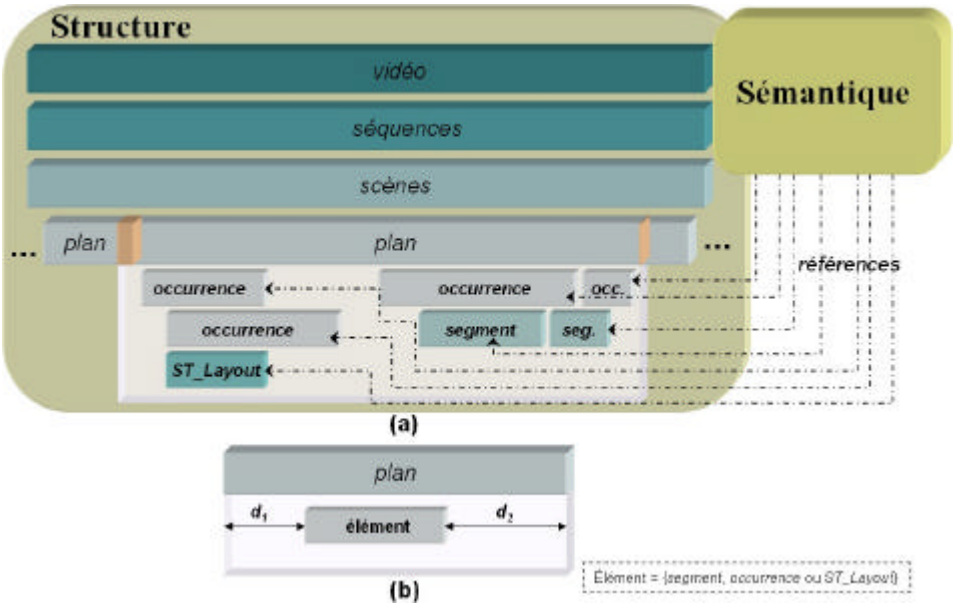

Figure 44. Structures hiérarchiques et relationnelles des éléments au niveau du plan.

Parmi les trois types d'éléments du plan, l'élément *segment* est simplement un fragment temporel libre dans un plan vidéo. Par contre, deux autres éléments (*occurrence* et *ST_Layout*) sont des segments spatio-temporels dont les structures sont assez complexes et que nous décrivons ci-dessous.

3. Structure logique d'une occurrence

L'élément *occurrence* décrit une personne ou un objet qui apparaît dans le plan. Il se réfère à un élément d'objet vidéo dans la description sémantique qui correspond à cette personne ou cet objet. La description des occurrences nous permet d'associer des actions aux apparitions d'objets vidéo comme ajouter un hyperlien, filtrer la vidéo selon des critères d'apparition d'objet, rechercher des fragments d'objet contenant des objets, etc. Dans notre modèle, une description d'occurrence est composée des éléments suivants :

- ♦ Les éléments de description des caractéristiques visuelles de l'occurrence comme la disposition spatiale des couleurs, l'histogramme de couleur, la texture, la forme et le contour.
- ♦ L'élément de localisation de l'occurrence dans l'espace et le temps qui permet de décrire des régions clef et des paramètres d'une fonction d'interpolation pour déterminer des positions intermédiaires de l'occurrence.
- ♦ Enfin il peut y avoir des sous-occurrences à l'intérieur d'une occurrence comme les composants de l'objet [Paek et al. 99a], par exemple, les bras d'un personnage, ses vêtements, son chapeau, etc.

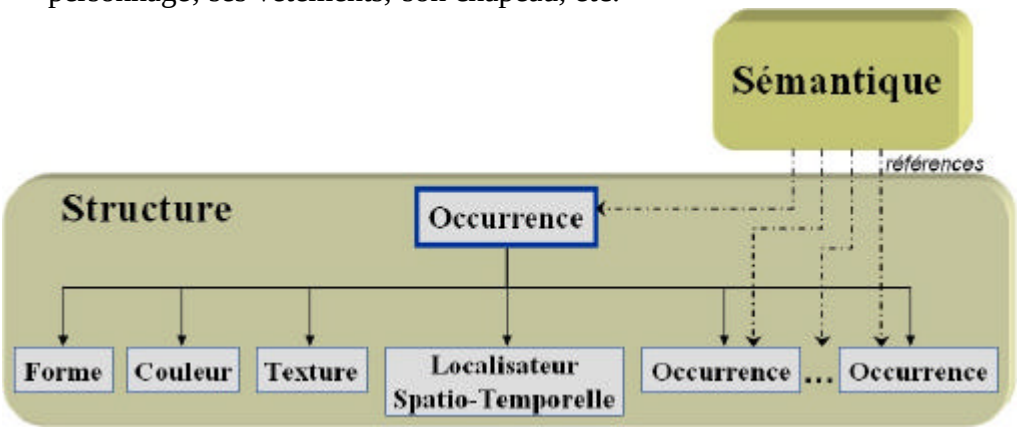

Figure 45. Structuration d'occurrence.

4. Disposition spatio-temporelle entre des occurrences

La disposition spatio-temporelle décrit des relations spatiales parmi des occurrences qui apparaissent en même temps dans un plan. En raison du comportement dynamique intrinsèque des informations du contenu de la vidéo, les relations spatiales peuvent changer dans le temps. Par exemple (voir la Figure 46), au début d'un plan il y a une voiture *Taunus* qui suit une voiture *Volvo*, puis la *Taunus* s'approche de la *Volvo*, ensuite la *Taunus* double la *Volvo* par le côté droit et enfin la *Taunus* roule devant la *Volvo*. Pour décrire ces changements spatio-temporels, nous devons décomposer la description selon des périodes dans lesquelles les relations spatiales entre les occurrences ne sont pas changées. Dans l'exemple ci-dessus, nous pouvons donc décomposer la disposition spatio-temporelle des deux voitures en trois parties correspondant à : la *Taunus* est derrière la *Volvo*, la *Taunus* est à droite de la *Volvo* et la *Taunus* est devant la *Volvo*.

Figure 46. Exemple de la disposition de deux voitures dans un plan vidéo.

Chacune de ces dispositions logiques doit être exprimée sous forme d'un jeu de relations spatiales directionnelles (*gauche*, *droite*, *au-dessus*, *au-dessous*, *nord*, *sud*, *est*, *ouest*, etc.) ou topologiques (*égal*, *intérieur*, *contenir*, *couverture*, *recouvrir*, etc.) entre des régions spatiales représentant les occurrences. Ainsi, l'exemple ci-dessus peut être décrit selon les relations directionnelles de la façon suivante : la *Taunus* **est à droite de** la *Volvo*; la *Taunus* **est au-dessus de** la *Volvo*; et la *Taunus* **est au-dessous de** la *Volvo*. De telles descriptions des relations directionnelles ne sont pas capables de représenter les relations spatiales de l'espace réel filmé qui est par nature en trois dimensions (3D). Pour combler cette lacune sans faire appel à des modèles 3D qui sont complexes et lourds, nous proposons d'associer à ces descriptions des éléments sémantiques de la partie de description sémantique.

V.2.3.2 Description du contenu des autres médias

La description fine du contenu de la vidéo nous a permis d'expérimenter des synchronisations fines entre des éléments décrits dans la structure du contenu de la vidéo et d'autres médias. L'extension de notre approche aux autres médias améliore la puissance de composition fine entre des fragments de média. Par exemple, la modélisation fine du contenu de l'audio et du texte permet de composer et présenter des documents multimédias complexes comme un *Karaoke*, dans lequel des phrases du texte doivent être présentées en synchronisation avec des segments audio correspondants.

V.2.3.2.1 Description du contenu de l'audio et de l'image

Le contenu de ces types de médias n'est pas aussi riche et complexe que celui de la vidéo, c'est pourquoi la modélisation du contenu de ces médias n'est pas très compliquée. Nous avons repris directement des modèles génériques fournis par MPEG-7 pour décrire le contenu de l'audio et de l'image. Nous les présentons plus en détail dans la partie de l'implémentation des modèles, en sections V.2.4.2.5 et V.2.4.2.6.

V.2.3.2.2 Description du contenu du texte

Pour décrire le contenu des textes, nous avons choisi un modèle classique de la structure de document textuel qui est une décomposition hiérarchique en *chapitre*, *section*, *paragraphe*, *phrase*, *expression*, *mot* et *caractère*.

Ces schémas de description visent à décrire du contenu du texte, ils ne participent pas à la structuration du contenu. C'est pourquoi même s'il est très facile d'intégrer les structures de l'indexation dans le contenu du texte, celles-ci

sont séparées du contenu. L'indexation est effectuée à travers des attributs (*BeginChar* et *EndChar*) qui indiquent les caractères du début et de la fin d'un segment du texte (voir la Figure 47). Un tel modèle d'indexation garde proprement le contenu original du texte, parce qu'il n'interagit pas avec ce dernier. De plus, cela permet d'avoir plusieurs descriptions sur un même contenu du texte.

Pour décrire un texte non structuré (*free text*) les valeurs d'indexation sont les positions des caractères au début et à la fin du segment. De telles valeurs absolues ne sont pas adaptées à l'édition, car si une partie de texte est modifiée, il faut changer tous les descripteurs suivants. Le coût de changement peut être diminué si les positions sont relatives au niveau de leur segment.

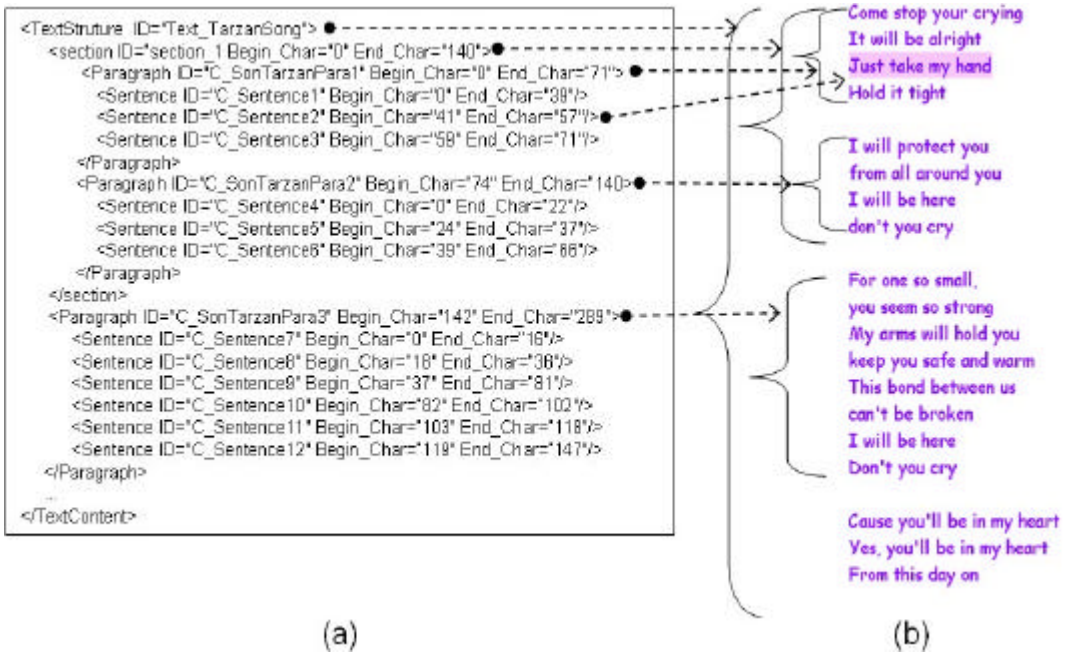

Figure 47. Descriptions séparées avec le contenu du texte a) description de la structure du texte, b) le contenu du texte.

Ce mécanisme de description permet aussi de décrire des documents textuels structurés comme HTML. Il est très utile dans le cas où l'utilisateur veut spécifier une synchronisation avec un segment que la structure du document n'identifie pas. L'exemple présenté dans la Figure 48 est un extrait du code d'un document HTML.


```
<td><font face="comic sans ms"><b><img src="…/oneday1.jpg"
align=left> If one day you feel like crying…<br>Call me.<br>I don't
promise that I will make you laugh,<br>But I can cry with you.<br>
</b> </font> </td> </tr>
```

Figure 48. Extrait de présentation et code d'un document HTML.

Bien évidemment ce document doit être complété par une description de sa structure pour permettre de spécifier des synchronisations fines comme, par exemple, afficher la phrase "*I don't promise*" en même temps que le segment de musique correspondant est joué. La description d'un tel document textuel peut être la suivante :

```
<HTMLContent>
    <MediaManagement>
        <MediaLocator>
            < !--référence vers le document HTML (Figure 48) -->
            <MediaUri>http://.../JustSayWow/IfOneDay.html</MediaUri>
        </MediaLocator>
        ...
    <MediaManagement>
    ...
    <TextStructure>
        <TexteSegment id="phrase_3" xsi:type="PhraseType"  BeginChar="33" EndChar="48" />
        ...
    </TextStructure>
</HTMLContent>
```

Dans le cas du texte structuré (HTML) on peut utiliser XPath[15] pour indexer le segment de texte. Par exemple le segment de texte ci-dessus peut être décrit comme suit :

```
<TexteSegment id="phrase_3" xsi:type="PhraseType"
    BeginChar="/html[1]/body[1]/…/td[n]//br[2],0 "
    EndChar="/html[1]/body[1]/…/td[n]//br[2],14" />
```

Ce mécanisme peut être comparé avec celui mis en œuvre dans *Annotea*, du W3C[16], qui utilise XPointer pour localiser le segment annoté dans un document [Kahan et al. 01]. Mais le problème de l'édition n'est pas résolu, si la structure du

[15] XML Path Language (XPath), http://www.w3.org/TR/xpath.

[16] Annotea, http://www.w3.org/2001/Annotea/.

document est changée, par exemple, par l'ajout d'un élément <*br*>, les descripteurs suivants doivent être modifiés. De nombreuses solutions sont possibles pour limiter cette situation : utilisation des id s'ils existent, des informations de contenu, etc.

V.2.4 Implémentation des modèles

Notre travail de modélisation a commencé à l'époque où la norme MPEG-7 était encore dans une phase de définition des besoins. Nous avons donc encodé notre modèle directement en XML [Roisin et al. 99], [Roisin et al. 00]. Puis à partir du milieu de l'année 2001 (après la réunion du groupe MPEG-7 le 10 mars 2001 à Singapore), le travail de MPEG-7 étant devenu plus mature, nous avons décidé d'adopter cette norme pour décrire notre modèle. Les sous-sections suivantes décrivent ces utilisations.

MPEG-7 fournit d'un côté des moyens très riches pour décrire des caractéristiques déterminées ou des schémas déterminés comme les méta informations, les informations de média, le schéma de descriptions sémantique, etc. que l'on peut utiliser directement. D'un autre côté, ces outils peuvent être trop génériques comme le schéma de description du contenu, ou même, ne satisfont pas nos besoins. Alors dans ce cas, on doit les étendre pour les adapter à nos buts. Nous détaillons notre utilisation de MPEG-7 dans les sous-sections suivantes qui décrivent les implémentations de nos modèles.

Nos schémas XML contiennent tous l'en-tête de déclaration ci-dessous qui définit les espaces de noms que nous utilisons dans les descriptions : MPEG-7 et Mdéfi.

```
<?xml version="1.0" encoding="utf-8"?>
<schema targetNamespace="http://www.inrialpes.fr/opera/MediaContent_Schema"
    xmlns:mdefi="http://www.inrialpes.fr/opera/MediaContent_Schema"
    xmlns:mpeg7="http://www.mpeg7.org/2001/MPEG-7_Schema"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
```

V.2.4.1 Implémentation de la structure générale

La structure générale des modèles (voir la section V.2.1) est implémentée en se basant sur le schéma racine <Mpeg7Type> de MPEG-7. Ce dernier fournit un élément racine <Mpeg7> qui est une extension du type complexe <Mpeg7Type> servant à décrire des contenus multimédias. Les schémas du type <Mpeg7Type> et de l'élément <Mpeg7> sont les suivants :

```
<complexType name="Mpeg7Type" abstract="true">

  <sequence>
    <element name="DescriptionMetadata" type="mpeg7:DescriptionMetadataType"
             minOccurs="0"/>
  </sequence>
  <attribute ref="xml:lang" use="optional"/>
</complexType>


<element name="Mpeg7">
    <complexType>
        <complexContent>
            <extension base="mpeg7:Mpeg7Type">
                <choice>
```

```
                    <element name="DescriptionUnit" type="mpeg7:Mpeg7RootType"/>

                    <choice minOccurs="1" maxOccurs="unbounded">
                        <element name="ContentDescription"
                            type="mpeg7:ContentDescriptionType"/>
                        <element name="ContentManagement"
                                type="mpeg7:ContentManagementType"/>
                    </choice>
                </choice>
                <attribute name="type" use="default" value="descriptionUnit">
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="descriptionUnit"/>
                            <enumeration value="complete"/>
                        </restriction>
                    </simpleType>
                </attribute>
            </extension>
        </complexContent>
    </complexType>
</element>
```

Cet élément racine <Mpeg7> permet de décrire soit une unité d'information du contenu <DescriptionUnit>, ou soit l'information complète du contenu <ContentDescription> ou/et <ContentManagement>. La description MPEG-7 complète permet de décrire un ensemble de médias, tandis que la description unique sert à décrire un morceau d'information extrait d'un contenu pour sa diffusion. Les deux cas ne conviennent pas à nos besoins. Parce qu'une description complète est trop grosse pour l'insérer dans un document. À l'inverse une description unique est trop simple, et ne peut pas fournir assez d'informations pour la composition multimédia.

C'est pourquoi nous avons décidé de créer notre élément <MediaDescription> pour représenter notre structure générale. Toutefois pour garder la compatibilité avec des descriptions MPEG-7, notre élément racine <MediaDescription> est une extension du type <Mpeg7Type>. De ce fait, notre élément <MediaDescription> hérite directement de l'élément de description de méta information <DescriptionMetadata>. Et comme la description de méta information est héritée directement du type <Mpeg7Type>, notre élément racine <MediaDescription> ne rajoute que quatre éléments : les informations sur le média <MediaInfos>, la sémantique du contenu <Semantics>, le thésaurus <Thesaurus> et le résumé (<Summary>). On peut remarquer que l'élément <MediaDescription> est un élément abstrait, qui sera hérité pour chaque type de média en rajoutant le schéma de la structure du contenu de chaque type de média.

La représentation en *XML Schema* de la structure générale d'une description de contenu de média est :

```
<!- Déclaration du schéma de la description du contenu des médias -->
<complexType name="MediaContentType" abstract ="true">
   <complexContent>
    <extension base="mpeg7:Mpeg7Type">
        <sequence>
         <!—la définition de l'élément de description des informations sur le media -->
         <element name="MediaManagement" minOccurs="0">
            <complexType>
                <sequence>
                    <choice minOccurs="0">
```

```
                <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
                <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
            </choice>
            <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
            <element name="UserDescription" type="mpeg7: UserDescriptionType"/>
            <element name="CreationInfo" type="mpeg7:CreationInformationType">
            <elem ent name="UsageInfo" type="mpeg7:UsageInformationType"/>
        </sequence>
    </complexType>
</element>

<!—la définition de l'élément sémantique -->
<element name="Semantics" type="mpeg7:WorldDescription Type" minOccurs="0"/>

<!--la définition de l'élément thésaurus -->

<element name="Thesaurus" type="mpeg7:ClassificationSchemeType" minOccurs="0"/>

<!—la définition de l'élément décrivant le résumé -->

<element name="Summary" type="mpeg7:ContentAbstractionType" minOccurs="0"/>

<!-- la définition de la structure du contenu sera rajoutée dans les sous classes -->

<sequence> </extension> </complexContent> </complexType>
```

Le schéma abstrait ci-dessus se base sur le schéma le plus général <Mpeg7Type> de MPEG-7 et utilise directement des schémas de modélisation de MPEG-7 pour décrire des informations sur le média, la sémantique, le thésaurus et le résumé du contenu de média. En effet MPEG-7 fournit les schémas pour modéliser :

- les informations de gestion de média (voir la section V.2.1) en utilisant un groupe de quatre schémas :
    a. mpeg7:MediaInformationType permet de décrire des informations sur le média dans lesquels les plus importantes sont le format et la localisation.
    b. mpeg7:UserDescriptionType permet de décrire des utilisateurs de ce média.
    c. mpeg7:CreationInformationType permet de décrire des informations concernant la création de média comme l'éditeur, l'auteur, etc.
    d. et enfin mpeg7:UsageInformationType permet de décrire des informations d'utilisation de média.
- la sémantique du contenu en utilisant l'outil mpeg7:WorldDescriptionType qui permet de définir des objets, des événements, des états, des lieux et des moments significatifs du monde réel ainsi que des relations entre eux pour représenter le récit significatif lié au contenu de média ;
- le thésaurus des informations en utilisant mpeg7:ClassificationSchemeType qui permet de définir des termes linguistiques pour un ensemble d'objets décrits ainsi que leur organisation ontologique ("classification") ;
- et enfin le résumé du contenu en utilisant l'outil mpeg7:SummaryDescriptionType qui permet de définir un ensemble de

segments clés (*key-clips*, *key-frame*, etc.) et qui peut transmettre des informations essentielles du contenu média.

Les outils utilisés ci-dessus sont très puissants. Ils peuvent décrire différents types d'information de différentes spécialités et de différentes disciplines. Leurs capacités dépassent de beaucoup nos besoins. Nous avons décidé de ne pas les restreindre pour garder la plus grande compatibilité possible avec des descriptions MPEG-7. Pour avoir plus d'informations, voir [Beek et al. 01].

V.2.4.2 Implémentation des modèles de la structure du contenu des médias

La norme MPEG-7 fournit le schéma générique <SegmentType> et ses extensions : le segment de la vidéo <VideoSegmentType>, le segment de l'audio <AudioSegmentType>, le segment de l'image <StillRegionType>, etc. qui peuvent être employés pour décrire la structure temporelle et spatiale du contenu de chaque type de média (la vidéo, l'audio et l'image). La Figure 49 (issue de [Beek et al. 01]) présente les extensions des différents segments spécialisés à partir du type *Segment DS* générique.

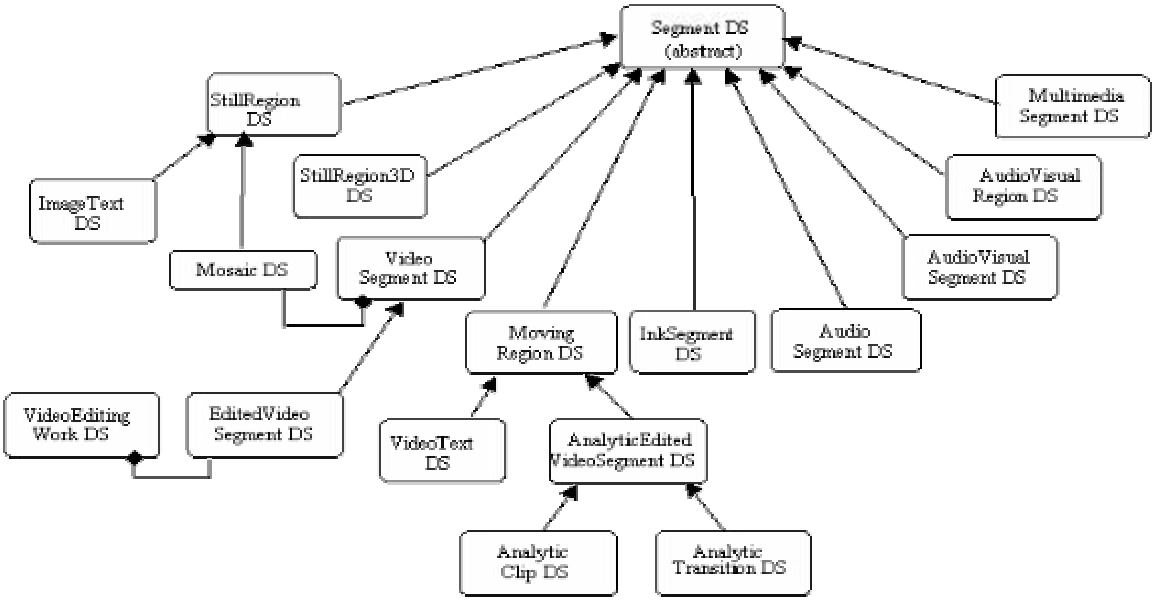

Figure 49. Ensemble des outils pour décrire les segments de contenu multimédia.

Ce schéma standard fournit des bases riches pour la description du contenu en s'appuyant globalement sur trois axes :

- Les attributs de segment sont liés à des informations spatio-temporelles, des informations de médias (localisation, création, utilisation, etc.), des caractéristiques et des masques; le poids d'importance pour comparer avec le segment ; le poids d'importance du segment donné par un point de vue ; etc.
- La décomposition structurelle décrit récursivement des sous-segments de contenu multimédia.
- Les relations entre segments décrivent des relations structurelles parmi des segments comme les relations temporelles, spatiales et spatio-temporelles, et d'autres relations.

Ces schémas répondent à l'objectif de pouvoir s'adapter le plus largement possible aux applications, ils sont donc génériques pour nos besoins, i.e., un schéma peut s'appliquer à décrire un segment arbitraire de média, par exemple un segment de la vidéo (<VideoSegmentType>) peut représenter un événement, un plan, une scène, une séquence ou même la vidéo entière. Une telle description générique ne peut pas représenter directement nos modèles qui se basent sur la structure narrative et logique des médias (voir la section V.2.3). Dans la suite de cette section nous présentons comment adapter ces schémas standards pour décrire les structures narratives des médias : la vidéo, l'audio, l'image et enfin le texte.

Les discussions sont organisées de la manière suivante : nous présentons d'abord la syntaxe, la sémantique et la capacité des schémas MPEG-7 et puis nous décrivons leur application dans nos modèles.

V.2.4.2.1 Le schéma Segment DS de MPEG-7

Le schéma *Segment DS* est le type abstrait le plus générique de l'ensemble des schémas MPEG-7 pour décrire la structure du contenu multimédia. La syntaxe XML du type *SegmentType* peut être consultée en annexe A. Il décrit des propriétés qui peuvent s'appliquer à toutes les descriptions des segments spécialisés. Ces propriétés peuvent être regroupées en trois parties : celles relatives à l'indexation, au management et celles qui permettent de décrire des relations. La structure du contenu du média décrit est spécialisée dans les sous-types (cf. les sections V.2.4.2.2, V.2.4.2.3, V.2.4.2.5 et V.2.4.2.6).

Le schéma *SegmentType* fournit principalement des moyens pour indexer richement un segment de média, par exemple, l'élément *StructuralUnit* sert à décrire le rôle du segment dans la structure du contenu multimédia (*information*, *résumé* et *rapport* dans un reportage ; ou *séquence*, *scène* et *plan* dans un film) ; l'élément *MatchingHint* permet d'associer un poids d'importance à un critère de comparaison ; l'élément *TextAnnotation* est utilisé pour mettre des commentaires sur le segment ; enfin, l'élément *PointOfView* définit son importance selon un point de vue spécifique ; etc. [Beek et al. 01].

Le schéma *SegmentType* de la norme MPEG-7 permet de décrire un **contenu multimédia** (voir la Figure 50a) qui peut contenir plusieurs segments incorporant différents types de média. Ces segments peuvent provenir de différentes sources, peuvent être de différents formats, etc. C'est pourquoi des éléments comme *DescriptionMetadata* (hérité de *mpeg7:DSType*), *MediaInformation*, *MediaLocator*, *CreationInformation* et *UsageInformation* (cf. V.2.1) sont tous intégrés dans le type *SegmentType* pour permettre à chacun des segments d'avoir sa propre information de management (voir la Figure 50a). Dans notre modèle, nous avons besoin de décrire la structure du contenu d'un seul média individuel dont les descriptions de métadonnées et les informations de management sont homogènes à tous les segments du média. C'est pourquoi ces descriptions sont spécifiées une seule fois dans la partie de description générale de notre modèle (voir la section V.2.1 et son implémentation dans la section V.2.4.1) (voir la Figure 50b). Ces éléments de méta information et de management de média deviennent donc inutiles au niveau de la description d'un segment.

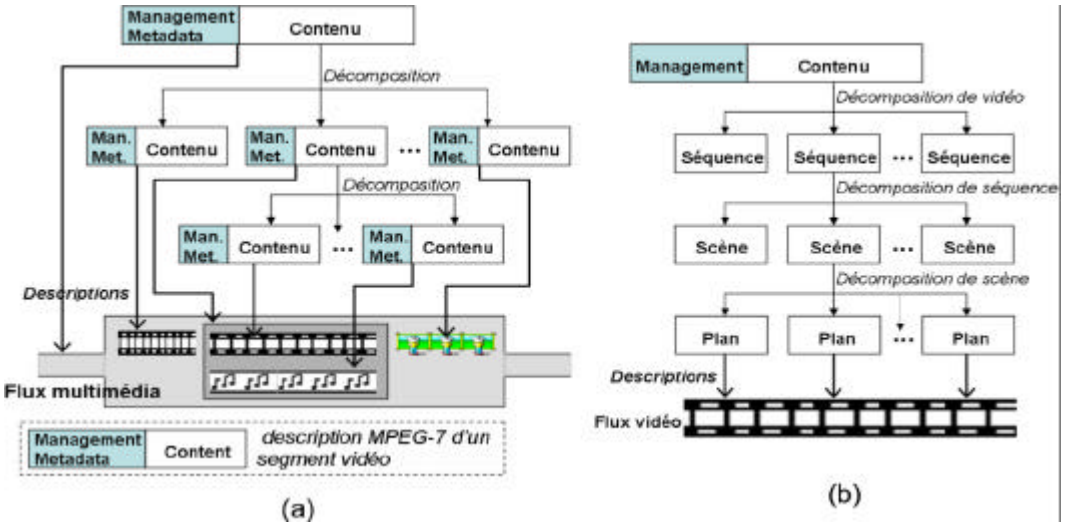

Figure 50. Différences entre (a) le modèle de description du contenu multimédia de MPEG-7 et (b) notre modèle de description du contenu d'une vidéo individuelle.

L'élément *Relation* permet de décrire les relations du segment avec d'autres segments, par exemple, que le segment A est à gauche du segment B. Cependant, au niveau de la structure logique, de la vidéo elle n'est pas utilisée. Dans la structure logique de la vidéo (voir la section V.2.3.1) les plans dans une scène, les scènes dans une séquence, les séquences dans la vidéo ont des relations temporelles spécifiques qui n'ont pas besoin d'être définies explicitement (voir la Figure 42 de la section V.2.3.1). Dans ce cas l'élément *Relation* est évidemment inutile.

Dans les sections qui suivent, nous présentons les schémas étendus de ce type et leurs utilisations pour représenter nos modèles. Ces utilisations prennent en compte les remarques ci-dessus.

V.2.4.2.2 Le schéma VideoSegment DS de MPEG-7 et l'implémentation de la structure logique de la vidéo

Le schéma *VideoSegment DS* est une extension du type *Segment DS*. Le schéma *VideoSegment DS* (cf. la Figure 49) décrit un intervalle temporel ou un segment de vidéo, qui peut correspondre à une séquence quelconque de trames, une simple trame, ou même une vidéo entière. Le segment vidéo peut être continu ou discontinu dans le temps (cf. [Beek et al. 01]). La syntaxe du type est la suivante :

```
<complexType name="VideoSegmentType">
        <complexContent>
            <extension base="mpeg7:SegmentType">
                <sequence>
                    <element name="MediaTime" type="mpeg7:MediaTimeType"minOccurs="0"/>
                    <element name="TemporalMask" type="mpeg7:TemporalMaskType"minOccurs="0"/>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
                        <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
                        <element name="TimeSeriesDescriptors" type="mpeg7:TimeSeriesType"/>
                        <element name="Mosaic" type="mpeg7:MosaicType"/>
                    </choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element name="SpatialDecomposition"
                                type="mpeg7:VideoSegmentSpatialDecompositionType"/>
                        <element n ame="TemporalDecomposition"
                                type="mpeg7:VideoSegmentTemporalDecompositionType"/>
                        <element name="SpatioTemporalDecomposition"
                                type="mpeg7:VideoSegmentSpatioTemporalDecompositionType"/>
                        <element name="MediaSourceDecomposition"
                                type="mpeg7:VideoSegmentMediaSourceDecompositionType"/>
                    </choice>
                </sequence>
            </extension>
        </complexContent>
</complexType>
```

En héritant des éléments du type *Segment DS* l'outil de description de segment de la vidéo rajoute des éléments pour décrire des propriétés temporelles, visuelles et liées à la décomposition :

- **Descriptions temporelles** : l'élément *MediaTime* permet de localiser temporellement le segment dans le contenu vidéo en spécifiant l'instant de début et la durée ; l'élément *TemporalMask* permet de décrire la fragmentation temporelle dans le cas où le segment est discontinu.
- **Descriptions visuelles** : de nombreux éléments permettent de décrire les caractéristiques visuelles de segment vidéo, par exemple les éléments *VisualDescriptor*, *VisualDescriptionScheme*, *Mosaic* pour : la couleur, la texture, l'*edge*, la mosaïque de segment vidéo ; l'élément *TimeSeriesDescriptors* est utilisé pour décrire une séquence temporelle avec ses caractéristiques visuelles.
- **Description de décomposition** : le segment vidéo fournit plusieurs façons de décrire la décomposition du segment qui consistent en des décompositions spatiales (*SpatialDecomposition*), temporelles (*TemporalDecomposition*) et même spatio-temporelles (*SpatioTemporalDecomposition*). En particulier, l'élément *MediaSourceDecomposition* permet de décrire plusieurs sources dans un segment multimédia, par exemple, un segment de film peut contenir des segments vidéo et des segments audio.

Ainsi, le type *VideoSegment DS* de MPEG-7 fournit une base riche pour décrire les segments de vidéo. Néanmoins, une application spécifique comme notre application de composition de document multimédia n'a pas besoin de toute la richesse du *Videosegment DS*. D'une manière générale, ce type cherche à couvrir tous les aspects de descriptions des segments vidéo. Il devient donc trop large et trop compliqué pour des applications spécifiques.

C'est pourquoi, notre schéma de la structure logique du contenu vidéo se base sur le *VideoSegment DS* MPEG-7 en éliminant les éléments suivants :

- Les éléments de méta information qui héritent de *Segment DS* (voir la section V.2.4.2.1 et la Figure 50).
- L'élément *Relation* qui hérite aussi de *Segment DS*, parce que les relations entre les segments de la structure logique n'ont pas besoin d'être décrits explicitement (voir la partie 1. *Structure logique de la vidéo* dans la section V.2.3.1 ; et la section V.2.4.2.1).
- L'élément *TemporalMask* qui est utilisé dans le cas où le segment décrit est discontinu car les segments vidéo de notre structure sont continus.
- les autres éléments comme *Annotation*, *MatchingHint*, *PointOfView*, *VisualDescriptor*, *VisualDescriptionScheme*, *TimeSeriesDescriptors*, et *Mosaic* décrivent des caractéristiques appropriées pour indexer et récupérer les segments. Pour notre application de synchronisation multimédia ils peuvent être utiles pour des synchronisations génériques. Nous avons spécifié ce critère pour notre modèle (cf. la section V.2). Mais comme nous n'avons pas encore expérimenté cette capacité, nous n'avons donc pas implémenté ces éléments.

Il est très simple de ne pas utiliser les éléments ci-dessus tout en restant compatible avec le type standard, parce que leurs spécifications dans la définition des types *SegmentType* *VideoSegmentType* les laissent optionnels (*minOccurs="0"*).

Par contre, les éléments *MediaTime*, *StructuralUnit* et la décomposition sont importants pour notre travail expérimental actuel. L'élément *MediaTime* est utilisé pour représenter les informations temporelles (le début et la durée). L'élément *StructuralUnit* est utilisé pour représenter le type (*Séquence*, *Scène*, *Plan*, etc.) de segment.

L'élément de décomposition doit être redéfini parce que les décompositions dans la structure logique de notre modèle vidéo ne sont que les décompositions temporelles, et de plus elles sont plus spécifiques. En effet chaque niveau de décomposition (vidéo, séquence et scène) ne doit contenir que des éléments particuliers (les plans dans une scène par exemple) et porter implicitement des relations temporelles correspondantes (voir la partie *"1. Structure logique de la vidéo"* dans la section V.2.3.1). Nous définissons donc les schémas de décomposition à partir du le schéma *TemporalSegmentDecompositionType* de MPEG-7 comme ci-dessous :

```
<complexType name="VideoStructureDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <sequence minOccurs="1" maxOccurs="unbounded">
                <element name="VideoSequence" type="mdefi:VideoSequenceType"/>
                <element name="VideoTransition" type="mdefi:VideoTransitionType" minOccurs="0"/>
            </sequence>
            <attribute name="overlap" type="boolean" use="fixed" value="true"/>
            <attribute name="gap" type="boolean" use="fixed" value="false"/>
        </extension>
    </complexContent>
```

```
</complexType>

<complexType name="VideoSequenceDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <sequence minOccurs="1" maxOccurs="unbounded">
                <element name="VideoScene" type="mdefi:VideoSceneType"/>
                <element name="VideoTransition" type="mdefi:VideoTransitionType" minOccurs="0 "/>
            </sequence>
            <attribute name="overlap" type="boolean" use="fixed" value="true"/>
            <attribute name="gap" type="boolean" use="fixed" value="false"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="VideoSceneDecompositionType">
    <complexContent>
        <extension base="mpeg7:TemporalSegmentDecompositionType">
            <sequence minOccurs="1" maxOccurs="unbounded">
                <element name="VideoShot" type="mdefi:VideoShotType"/>
                <element name="VideoTransition" type="mdefi:VideoTransitionType"minOccurs="0"/>
            </sequence>
            <attribute name="criteria" type="string" use="default" value="color"/>
            <attribute name="overlap" type="boolean" use="fixed" value="true"/>
            <attribute name="gap" type="boolean" use="fixed" value="false"/>
        </extension>
    </complexContent>
</complexType>
```

On peut remarquer qu'il n'existe pas de "trous" entre les segments de décomposition, et qu'ils peuvent se recouvrir comme dans le cas où des transitions ne sont pas un *cut*. C'est pourquoi les attributs *gap* et *overlap* ont leurs valeurs fixés (*gap="false"* et *overlap="true"*). Actuellement, la décomposition automatique d'une scène en plans est souvent basée souvent sur le critère d'histogramme de couleur. C'est pourquoi l'attribut *criteria* de la décomposition de la scène possède défaut la valeur *"color"*.

La décomposition d'un plan est plus libre (*gap="true"* et *overlap="true"*), et elle comprend à la fois d'éléments temporels : le *segment* ; d'éléments spatio-temporels : l'*occurrence* et les dispositions spatio-temporelles entre des occurrences (voir la partie *"2. Structure logique d'un plan"* dans la section V.2.3.1). C'est pourquoi notre schéma de décomposition de plan se base sur le schéma *SpatioTemporalSegmentDecompositionType* de MPEG-7 de façon suivante :

```
<complexType name="VideoShotDecompositionType">
    <complexContent>
        <extension base="mpeg7:SpatioTemporalSegmentDecompositionType">
            <choice maxOccurs="unbounded">
                <element name="Segment" type="mpeg7:VideoSegmentType"/>
                <element name="Occurrence" type="mdefi:OccurrenceType"/>
                <element name="OccurrenceSpatioTemporalLayout"
                        type="mdefi:OccurrenceSpatioTemporalLayoutType"/>
            </choice>
            <attribute name="overlap" type="boolean" use="fixed" value="true"/>
            <attribute name="gap" type="boolean" use="fixed" value="true"/>
        </extension>
    </complexContent>
</complexType>
```

Enfin, la syntaxe de notre modèle de la structure logique du contenu vidéo (cf. la Figure 42) est basée sur le type *VideoSegment DS* comme présenté ci-dessous. Précisément, nous définissons quatre nouveaux types *VideoStructureType*, *VideoSequenceType*, *VideoSceneType* et *VideoShotType* qui héritent de tous les éléments du type *mpeg7:VideoSegmentType*. De plus, pour chaque type on ajoute une décomposition appropriée telle que définie ci-dessus.

```
<complexType name="VideoStructureType">
   <complexContent>
      <extension base="mpeg7:VideoSegment">
         <sequence>
            <element name="VideoStructureDecomposition"
                     type="mdefi:VideoStructureDecompositionType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<complexType name="VideoSequenceType">
   <complexContent>
      <extension base="mpeg7:VideoSegment">
         <sequence>
            <element name="VideoSequenceDecomposition"
                     type="mdefi:VideoSequenceDecompositionType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<complexType name="VideoSceneType">
   <complexContent>
      <extension base="mpeg7:VideoSegment">
         <sequence>
            <element name="VideoSceneDecomposition"
                     type="mdefi:VideoSceneDecompositionType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>

<complexType name="VideoShotType">
   <complexContent>
      <extension base="mpeg7:VideoSegment">
         <sequence>
            <element name="VideoShotDecomposition"
                     type="mdefi:VideoShotDecompositionType"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

Le schéma de description du contenu de la vidéo se basant sur le schéma abstrait *MediaContentType* et rajoutant l'élément racine de la structure du contenu vidéo *VideoStructureType* est le suivant :

```
<complexType name="VideoContentType">
   <complexContent>
      <extension base="mdefi:MediaContentType">
         <sequence>
            <element name="VideoStructureElement" type="VideoStructureType"/>
         </sequence>
      </extension>
```

```
    </complexContent>
</complexType>
```

V.2.4.2.3 Le schéma MovingRegion DS de MPEG-7 et l'implémentation d'occurrences

Le type *MovingRegion DS* est une extension du type *Segment DS* pour permettre de décrire une région 2D en mouvement dans un plan. La syntaxe du type est définie par le schéma suivant :

```
<complexType name="MovingRegionType">
    <complexContent>
        <extension base="mpeg7:SegmentType">
            <sequence>
                <element name="SpatioTemporalLocator"
                        type="mpeg7:SpatioTemporalLocatorType" minOccurs="0"/>
                <element name="SpatioTemporalMask" type="mpeg7:SpatioTemporalMaskType"
                        minOccurs="0"/>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
                    <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
                    <element name="TimeSeriesDescriptors" type="mpeg7:TimeSeriesType"/>
                </choice>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="SpatialDecomposition"
                            type="mpeg7:MovingRegionSpatialDecompositionType"/>
                    <element name="TemporalDecomposition"
                            type="mpeg7:MovingRegionTemporalDecompositionType"/>
                    <element name="SpatioTemporalDecomposition"
                            type="mpeg7:MovingRegionSpatioTemporalDecompositionType"/>
                    <element name="MediaSourceDecomposition"
                            type="mpeg7:MovingRegionMediaSourceDecompositionType"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Notre type de description d'une occurrence est un cas d'utilisation simple du type *MovingRegion DS*, toutefois, pour garder la compatibilité avec ce type, notre type *OccurrenceType* est réalisé comme une extension du type *MovingRegion DS*.

Comme dans le cas du type *VideoSegment DS*, le type *MovingRegion* fournit aussi une base riche pour décrire des régions en mouvement. Cette richesse peut être groupée en trois parties : la localisation (*SpatioTemporalLocator*, *SpatioTemporalMask*), la caractéristique (*VisualDescriptor*, etc.) et la décomposition (*SpatialDecomposition*, etc.). Nous ne pouvons pas tout exploiter dans le cadre de notre travail. Actuellement, notre système ne se concentre que sur les deux type d'éléments plus utilisés par la composition multimédia : les éléments de décomposition et l'élément *SpatioTemporalLocator*.

Pour la décomposition de plan, nous proposons d'utiliser l'élément *SpatioTemporalDecomposition* ce qui est le plus approprié pour décrire la décomposition spatio-temporelle d'une occurrence composite en différentes sous occurrences.

L'élément *SpatioTemporalLocator* permet de déterminer la position d'une occurrence à un instance quelconque. Cela permet de composer des

synchronisations spatio-temporelles avec une occurrence ou de définir des hyperliens sur un objet vidéo. L'élément *SpatioTemporalLocator* décrit des régions se déplaçant dans un segment vidéo par un ensemble de **régions de référence** et **ses mouvements** (voir la Figure 51 qui est reprise de [Cieplinski et al. 01]). Les régions peuvent être sous forme quelconque (rectangle, cercle, ellipse ou polygone, etc.). Les **mouvements** peuvent être décrits par deux types de descriptions : soit par la trajectoire de la figure (*FigureTrajectory*) définie comme une suite de positions ; ou soit la description de la trajectoire est donnée par une fonction d'un paramètre (*ParameterTrajectory*). Ces deux types de description sont choisis selon les conditions de mouvement de l'objet. En général, si le mouvement est régulier et le modèle de mouvement est connu, *ParameterTrajectory* est préférable en raison de sa compacité. Par contre, si le mouvement est complexe avec de plus des déformations, la description *FigureTrajectory* est plus appropriée.

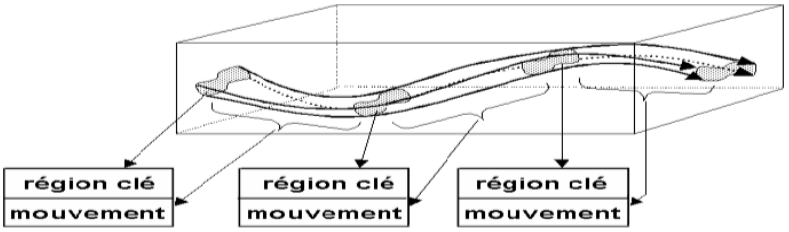

Figure 51. Description de région spatio-temporelle.

Héritant de ces propriétés descriptives de *mpeg7:MovingRegionType*, notre type *OccurrenceType* peut servir à la composition riche appliquée aux objets vidéos. Dans le cadre d'expérimental de cette thèse, nous n'avons exploité cette puissance descriptive que pour composer des synchronisations spatio-temporelles fines avec un objet vidéo, mettre des hyperliens sur un objet vidéo, et extraire la présentation d'un objet vidéo. Il y a aussi d'autres applications qu'on peut déployer en utilisant les *descriptions des caractéristiques* du segment. Une application intéressante est la création de synchronisations génériques qui se basent sur des caractéristiques du segment pour éditer des scénarios multimédia non prévus, par exemple, déclencher une alarme (*audio*) lorsque que la vitesse de mouvement d'un objet (*MovingRegion*) qui a les caractéristiques physiques d'une voiture. Ce type d'application peut être particulièrement approprié pour composer des synchronisations avec un flux multimédia temps réel dont le contenu est imprévisible. Une telle application n'est pas simple car elle demande une intégration de trois domaines multimédia (*analyse*, *description* et *intégration/synchronisation*), et dans ce travail nous ne proposons que les premières étapes pour cette intégration.

V.2.4.2.4 Le schéma de la disposition spatio-temporelle entre des occurrences

La norme MPEG-7 fournit des schémas de description des relations entre des segments média pour les trois types de relations : temporelles, spatiales et spatiotemporelles. Si les schémas et les relations sont bien définis pour les relations spatiales et temporelles (relations spatiales directionnelles, relations temporelles de Allen, etc.), le schéma de description des relations spatio-temporelles est actuellement très pauvre.

Un ensemble de relations peut être décrit à travers une structure de graphe dans laquelle les nœuds correspondent aux segments et les arcs correspondent aux relations entre des segments. Comme une structure de graphe peut décrire des relations plus générales entre des éléments qu'une structure d'arbre. Nous utilisons l'outil mpeg7:GraphType pour construire nos descriptions des relations spatio-temporelles entre segments. Nous proposons ici de spécifier ces relations sous forme d'une suite d'intervalles temporels pendant lesquels les occurrences sont en relation spatiale stable.

```
<complexType name="OccurrenceSpatioTemporalLayoutType ">
        <sequence >
            <element name="Segment" type="mpeg7:VideoSegmentType" minOccurs="0"
                     maxOccurs="unbounded"/>
            <element name="Relationships" type="mpeg7:GraphType" maxOccurs="unbounded"/>
        </sequence>
</complexType>
```

Le schéma de description consiste en des segments de décomposition d'une relation spatio-temporelle et un graphe de relations qui peuvent décrire des relations spatiales, temporelles, de contenu, etc. entre des segments vidéo. L'exemple des relations spatio-temporelles entre deux voitures dans la Figure 46 (A=*Taunus*, B=*Volvo*) peut être présenté sous forme graphique comme dans la Figure 52.

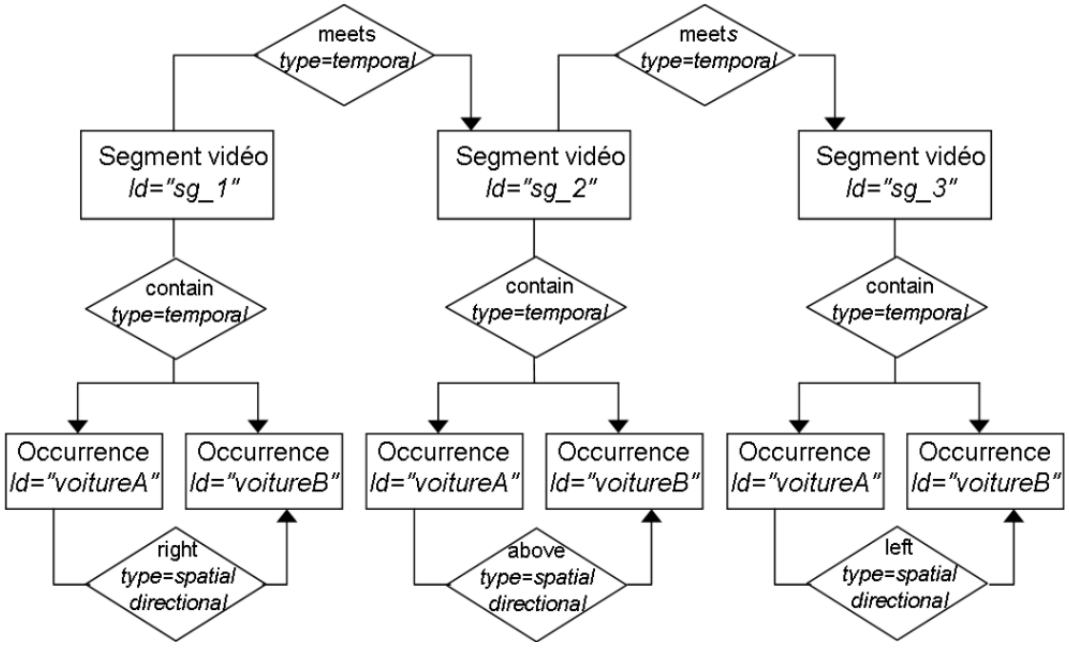

Figure 52. Graphe de description de la relation spatio-temporelle entre voitures A et B.

Avec ce modèle, la description concrète de l'exemple peut être la suivante :

```
<VideoSegment id="plan_1" xsi:type="VideoShotType">
        <MovingRegion id="VoitureA-movreg" xsi:type="OccurrenceType">
            <SpatioTemporalLocator>...</SpatioTemporalLocator>
        </MovingRegion>
        <MovingRegion id="VoitureB-movreg" xsi:type="OccurrenceType">
            <SpatioTemporalLocator>...</SpatioTemporalLocator>
        </MovingRegion>
```

```
        <Segment id="sg_1" xsi:type="VideoSegmentType">
            <MediaTime> ... </MediaTime>
        </Segment>
        <Segment id="sg_2" xsi:type="VideoSegmentType">
            <MediaTime> ... </MediaTime>
        </Segment>
        <Segment id="sg_3" xsi:type="VideoSegmentType">
            <MediaTime> ... </MediaTime>
        </Segment>

        <Graph >
            <Node id=" nodeA_1" href="#VoitureA-movreg "/>
            <Node id=" nodeA_2" href="#VoitureA-movreg "/>
            <Node id=" nodeA_3" href="#VoitureA-movreg "/>
            <Node id="nodeB_1" href="#VoitureB-movreg "/>
            <Node id="nodeB_2" href="#VoitureB-movreg "/>
            <Node id="nodeB_3" href="#VoitureB-movreg "/>
            <Node id="nodeSg_1" href="# sg_1 "/>
            <Node id="nodeSg_2" href="# sg_2 "/>
            <Node id="nodeSg_3" href="# sg_3 "/>

            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contain s"
                source="#nodeSg_1" target="#nodeA_1 "/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contains"
                source="#nodeSg_1" target="#nodeB_1"/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contains"
                source="#nodeSg_2 " target="#nodeA _2"/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contains"
                source="#nodeSg_2" target="#nodeB_2"/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contains"
                source="#nodeSg_3" target="#nodeA _3"/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="contains"
                source="#nodeSg_3" target="#nodeB_3"/>

            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="meet s"
                source="#nodeSg_1" target="#nodeSg_2"/>
            <Relation xsi:type="BinaryTemporalSegmentRelationType" name="meets"
                 source="#nodeSg-2" target="#nodeSg_3"/>

            <Relation xsi:type="DirectionalSpatialSegmentRelationType" name="right"
                source="#nodeA_1" target="#nodeB_1"/>
            <Relation xsi:type="DirectionalSpatialSegmentRelationType" name="above"
                 source="#nodeA _2" target="#nodeB_2"/>
            <Relation xsi:type="DirectionalSpatialSegmentRelationType" name="left "
                source="#nodeA_3 " target="#nodeB_3"/>
        </Graph>
</VideoSegment>
```

V.2.4.2.5 Le schéma de description du contenu de l'audio

Le schéma *AudioSegment DS* de la norme MPEG-7 permet de décrire un intervalle/segment temporel de données audio. Ce schéma est suffisamment générique pour décrire à la fois une séquence quelconque d'échantillons audio, un échantillon simple, ou même une audio entière. Il convient tout à fait à nos besoins de description de segments audio selon des granularités variables pour en permettre la composition dans des documents multimédias. La syntaxe du schéma est la suivante (cf. [Beek et al. 01]) :

```
<complexType name="AudioSegmentType">
    <complexContent>
        <extension base="mpeg7:SegmentType">
            <sequence>
                <element name="MediaTime" type="mpeg7:MediaTimeType"
                         minOccurs="0"/>
                <element name="TemporalMask" type="mpeg7:TemporalMaskType"
                         minOccurs="0"/>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="AudioDescriptor" type="mpeg7:AudioDType"/>
                    <element name="AudioDescriptionScheme" type="mpeg7:AudioDSType"/>
                </choice>
                <choice minOccurs="0" maxOccurs="unbounded">
                    <element name="TemporalDecomposition"
                             type="mpeg7:AudioSegmentTemporalDecompositionType"/>
                    <element name="MediaSourceDecomposition"
                             type="mpeg7:AudioSegmentMediaSourceDecompositionType"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Le schéma est également une extension du type *Segment DS* présenté dans la section V.2.4.2.1 pour pouvoir représenter : 1) le temps dans le segment audio ; 2) les caractéristiques audio ; et 3) la décomposition du segment audio.

La nature du contenu de l'audio est plus riche qu'un segment générique : parole, musique, bruit, locuteur, instrument de musique, silence, etc. Bien que l'utilisation du schéma *AudioSegment DS* puisse représenter génériquement tous ces types audio, une telle représentation générale des segments audio n'est pas toujours suffisante pour l'utiliser lors de l'édition du document multimédia. Par exemple, si l'auteur veut synchroniser des paroles d'un chanteur dans une chanson avec les phrases de texte de cette chanson, il lui sera très difficile d'extraire les segments audio des paroles de chanteur parmi l'ensembles des segments musiques, des bruits, etc. si tous ces segments sont représentés identiquement par le même type de segment audio ; au contraire si les paroles sont décrites par un schéma spécifique, ils peuvent être retrouvés facilement et même leur synchronisation avec le texte peut être réalisée automatiquement.

Toutefois l'utilisation directe de l'outil générique *AudioSegment DS* a des avantages. Elle diminue le coût d'implémentation (on doit implémenter un seul élément au lieu de plusieurs éléments pour chaque type d'audio). Il est de plus intéressant de garder la compatibilité avec le standard pour intégrer directement des descriptions d'autres origines sans traitement particulier.

Enfin le schéma de description du contenu de l'audio se base sur le schéma abstrait *MediaContentType* et rajoute l'élément de la structure audio *AudioStructureElement* :

```
<complexType name="AudioContentType">
    <complexContent>
        <extension base="mdefi:MediaContentType">
            <sequence>
                <element name="AudioStructureElement" type="mpeg7:AudioSegmentType"/>
                </sequence>
        </extension>
    </complexContent>
```

```
</complexType>
```

V.2.4.2.6 Le schéma de description du contenu de l'image

De la même façon que pour la description du contenu de l'audio, nous proposons d'utiliser directement *StillRegion DS* de la norme MPEG-7 pour décrire le contenu de l'image. Le schéma *StillRegion DS* permet de décrire à la fois toute région formée par un ensemble de pixels, ou même une image entière. Sa syntaxe est définie de la manière suivante :

```
<complexType name="StillRegionType">
        <complexContent>
            <extension base="mpeg7:SegmentType">
                <sequence>
                    <element name="SpatialLocator" type="mpeg7:RegionLocatorType"
                                minOccurs="0"/>
                    <element name="SpatialMask" type="mpeg7:SpatialMaskType"
                                minOccurs="0"/>
                    <choice minOccurs="0">
                        <element name="MediaTimePoint" type="mpeg7:mediaTimePointType"/>
                        <element name="MediaRelTimePoint"
                                type="mpeg7:MediaRelTimePointType"/>
                        <element name="MediaRelIncrTimePoint"
                                type="mpeg7:MediaRelIncrTimePointType"/>
                    </choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element name="VisualDescriptor" type="mpeg7:VisualDType"/>
                        <element name="VisualDescriptionScheme" type="mpeg7:VisualDSType"/>
                        <element name="GridLayoutDescriptors" type="mpeg7:GridLayoutType"/>
                    </choice>
                    <element name="SpatialDecomposition"
                                type="mpeg7:StillRegionSpatialDecompositionType"
                                minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
</complexType>
```

Le schéma est une extension du type *Segment DS* présenté dans la section V.2.4.2.1 pour pouvoir représenter : les coordonnées spatiales de la région, éventuellement, sur coordonnées temporelles (région dans une trame), les caractéristiques visuelles de la région, mais aussi la décomposition spatiale de la région.

En utilisant *StillRegion DS* la structure de l'image présentée dans la Figure 10 peut être décrite de la manière suivante :

```
<StillRegion id="Image_de_mariee">
        <ContourShape> ...</ContourShape>
        <SpatialDecomposition gap="true" overlap="true">
            <StillRegion id="PierreSR">...</StillRegion>
            <StillRegion id="FeuilleSR">
                <TextAnnotation>
                    <FreeTextAnnotation> Feuille de bananier </FreeTextAnnotation>
                </TextAnnotation>
                <ContourShape> ... </ContourShape>
            </StillRegion>
            <StillRegion id="VetementSR">
                <TextAnnotation>
                    <FreeTextAnnotation> Vêtement traditionel du Vietnam, le Ao Dai</FreeTextAnnotation>
                </TextAnnotation>
```

```
            <ContourShape> ... </ContourShape>
            <SpatialDecomposition gap="true" overlap="true">
                <StillRegion id="ChapeauSR"> ... </StillRegion>
                <StillRegion id="AvantSR"> ... </StillRegion>
                <StillRegion id="DerriereSR"> ... </StillRegion>
                <StillRegion id="PhenixSR"> ... </StillRegion>
            </SpatialDecomposition>
        </StillRegion>
        <StillRegion id="BambouSR"> ... </StillRegion>
        <StillRegion id="HerbeSR">...</StillRegion>
    </SpatialDecomposition>
</StillRegion>
```

Comme la description du contenu de l'audio, l'utilisation directe de *StillRegion DS* permet d'expérimenter rapidement et facilement l'application de description du contenu de l'image à l'édition du document multimédia. Une telle expérimentation directe a donné des résultats intéressants, par exemple, mettre un hyperlien sur une région d'image, placer un alignement spatial entre un objet et une région d'image, extraire une région l'intégrer dans un scénario, etc. Par contre, pour des compositions plus sophistiquées ou même automatiques (un des objectifs de troisième génération de document multimédia, voir la section II.3.4), un tel niveau de description du contenu des images n'est pas suffisant. Par exemple, je souhaite enrichir l'album photo de mon mariage avec des hyperliens vers la page Web de ma femme ainsi que vers ma page Web, les ancres source étant attachées à chaque région d'images où ma femme (resp. moi-même) apparaît. Si toutes les régions sont décrites identiquement par *StillRegion DS*, il sera difficile de retrouver quelles régions correspondent à des personnes. Par contre, si les régions où ma femme et moi-même apparaissent sont décrites par un schéma plus spécifique, par exemple *PeopleRegion*, la composition devient plus simple et même peut être en partie automatisée.

Enfin le schéma de description du contenu de l'image se base sur le schéma abstrait *MediaContentType* et rajoute l'élément de la structure image *ImageStructureElement* comme suit :

```
<complexType name="ImageContentType">
    <complexContent>
        <extension base="mdefi:MediaContentType">
            <sequence>
                <element name="ImageStructureElement" type="mpeg7:StillRegionType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

V.2.4.2.7 Le schéma de description du contenu du texte

La norme MPEG-7 ne fournit pas de schémas de description du contenu de texte. Nous devons donc les créer, en nous inspirant des nombreux modèles de description du texte, par exemple la TEI [TEI 01] permet de représenter l'ensemble des caractéristiques d'un texte ; ou Docbook [Docbook 01] permet de représenter des livres et des articles en utilisant le format SGML ou le format XML. Pour bénéficier des éléments de base afin de décrire un segment média quelconque *SegmentType* de MPEG-7, nous proposons de construire les schémas de description du contenu du texte en étendant le type *SegmentType* de MPEG-7.

Nous créons d'abord un type générique *TextSegmentType* qui peut être utilisé pour représenter tous les types de segment de texte (*chapitre*, *section*, *paragraphe*, *liste*, etc.). La syntaxe de ce type est la suivante :

```
<complexType name="TextSegmentType ">
        <complexContent>
            <extension base="mpeg7:SegmentType">
                <attribute name="BeginChar" type="ID" type="string" use="required"/>
                <attribute name="EndChar" type="ID" type="string" use="required"/>
            </extension>
        </complexContent>
</complexType>
```

Les segments spécifiques du texte (*chapitre*, *section*, *paragraphe*, *phrase*, *expression*, *mot* et *caractère*) sont des extensions de ce type générique. A titre d'exemple, nous donnons la définition des types *chapitre* et *phrase* (les autres types ont une définition similaire) :

```
<complexType name="ChapterType ">
        <complexContent>
            <extension base="mdefi:TextSegmentType">
                <sequence>
                    <element name="section" type ="mdefi:SectionType" minOccurs="0"
                            maxOccurs="unbounded">
                </sequence>
            </extension>
        </complexContent>
</complexType>
...
<complexType name="PhraseType ">
        <complexContent>
            <extension base="mdefi:TextSegmentType">
                <sequence>
                    <element name="word" type ="mdefi:WordType" minOccurs="0"
                            maxOccurs="unbounded">
                </sequence>
            </extension>
        </complexContent>
</complexType>
...
```

Le type *TextStructureType* permet de décrire la structure du texte. Il contient des descriptions de segment du texte qui peuvent être un segment générique ou n'importe quel segment spécifique ci-dessus. Sa syntaxe est :

```
<complexType name="TextStructureType ">
        <sequence>
            <element name="textSegment" type="mdefi:TextSegmentType" maxOccurs="unbounded"/>
        </sequence>
</complexType>
```

Enfin le schéma de description du contenu du texte se basant sur le schéma abstrait *MediaContentType* et rajoutant l'élément de la structure texte *TextStructureElement* est le suivant :

```
<complexType name="TextContentType">
        <complexContent>
            <extension base="mdefi:MediaContentType">
                <sequence>
                    <element name="TextStructureElement" type="mdefi:TextStructureType"/>
                </sequence>
            </extension>
        </complexContent>
</complexType>
```

V.2.5 Synthèse des modèles de descriptions du contenu des médias

Nous avons présenté ci-dessus des modèles de description du contenu des médias basiques : la vidéo, l'audio, l'image et le texte. Ces modèles visent à accéder plus finement au contenu des médias pour mieux les synchroniser lors de l'édition du document multimédia. Les modèles fournissent le moyen de définir le contenu des média selon différents aspects (métadonnées, données de gestion de média, sémantique du contenu, thésaurus, résumé du contenu et structure du contenu). Bien que toutes les descriptions du contenu des médias soient exploitables pour éditer des documents multimédias, nous avons fait le choix de restreindre notre modèle aux descriptions structurelles du contenu (comme première étape dans ce travail). En effet, nous considérons que ces dernières sont les plus fondamentales et les plus importantes pour l'édition. Elles permettent d'identifier directement des segments du contenu des médias. Dans ce but, nous avons choisi une approche qui permet de décrire le plus complètement possible le contenu du média, de la structure logique générale (séquence, scène et plan pour la vidéo) jusqu'aux éléments plus libres (occurrence, segment, relation spatio-temporelle). Ainsi, notre contribution en modélisation de média a principalement consisté à intégrer judicieusement plusieurs approches de l'état de l'art. De plus, nous proposons d'ajouter des contraintes temporelles pour les éléments dans la structure qui aident à éditer la description et à l'intégrer dans le document multimédia.

Un autre résultat intéressant est la représentation des modèles utilisant la norme MPEG-7. Nous contribuons en cela aux travaux de validation de ce nouveau standard et, conséquence plus intéressante, notre système gagne en interopérabilité par l'utilisation et la production de descriptions en format standard MPEG-7.

Enfin, nous allons pouvoir évaluer réellement ce travail d'une part à travers la qualité d'intégration de ces modélisations de média dans le modèle de document multimédia et d'autre part à travers la qualité des présentations multimédia résultantes. En effet notre objectif vise en tout premier lien à permettre d'accéder au contenu des médias pour réaliser des présentations multimédias plus riches. La section qui suit est consacrée à notre proposition de document multimédia, la section quatre au modèle d'animation et finalement le bilan global est effectué en section 5.

V.3 Modèle de document multimédia basé sur les sous-éléments de médias

Cette section propose d'étendre un modèle de document multimédia pour permettre d'utiliser des descriptions du contenu des médias en édition des documents multimédias. Le modèle de document multimédia utilisé pour cette extension est le

modèle Madeus qui est décrit dans la section III.4.8 comme un modèle flexible de document multimédia. Les descriptions du contenu des médias utilisées sont représentées selon les modèles présentés dans les sections précédentes.

Bien que le modèle Madeus soit basé sur une structure hiérarchique d'intervalles et de régions, qui offre un des modèles les plus expressifs [Wahl et al. 94], la limite de l'approche est principalement due au niveau insuffisant de granularité des feuilles de la structure. Avec la granularité des modèles actuels, l'auteur ne peut pas produire aisément les synchronisations fines nécessaires à la création de scénarios de présentation sophistiqués. Le problème ne peut pas être résolu en employant simplement les éléments *anchor* et *area* comme dans les modèles *CMIF* et *SMIL*. Comme on l'a vu dans la section III.4 du chapitre III leur mode de désignation des fragments et de bas niveau (décalage par rapport au début du média) et n'exploite pas des informations plus riches sur le contenu du média. Or, les fragments de média peuvent correspondre aux objets du média qui ont leur propre sémantique, leur organisation temporelle et spatiale (voir les sections III.4.4 et III.4.5). Le modèle multimédia doit considérer ces informations pour composer finement un document multimédia à partir de ces objets de média. L'utilisation des descriptions du contenu des médias est une bonne façon pour obtenir les informations d'un objet ou même un fragment de média.

Dans cette démarche pour intégrer un modèle de description de média dans un modèle de document multimédia, la principale difficulté à résoudre est la différence entre les deux modèles dans leur approche de description des structures de même nature (spatiale, temporelle et sémantique de contenu). Nous proposons donc une structure de **sous-éléments** qui permette d'adapter les descriptions du contenu des médias à la structure de présentation d'un document multimédia. Lorsque les descriptions du contenu d'un média sont présentées selon la structure du document, l'auteur peut facilement les utiliser pour composer des documents multimédias. La structure des sous-éléments est considérée comme l'extension du modèle de document multimédia.

Grâce à son modèle à base de structures hiérarchiques, il est facile avec Madeus d'introduire de nouvelles structures hiérarchiques pour les sous- éléments (voir la Figure 10). Les extensions sont faites dans chaque axe de décomposition du modèle Madeus (le *Contenu*, *Acteur*, *Temporel* et *Spatial*). C'est pourquoi le mot "sous-éléments" dans ce travail représente un ensemble de sous-éléments dans chaque axe : le *sous-acteur*, le *sous-intervalle* et la *sous-région*. Ces sous-éléments ne concernent que les trois derniers axes. L'extension dans l'axe *Contenu* est en fait l'ensemble des structures de description des médias vidéo, audio, image et texte. La suite de cette section détaille ces éléments d'extension.

V.3.1 Les médias structurés, extension de la partie de contenu

Dans la partie *Contenu* du modèle Madeus nous introduisons le concept de média structuré du moyen de quatre nouveaux types d'éléments : *StructuredVideo*, *StructuredAudio*, *StructuredImage* et *StructuredText*. Plus précisément, ces quatre types enrichissent l'élément *C-Group* du modèle Madeus défini dans [Villard 02] et dans la section III.4.8. Les médias structurés peuvent être ainsi distingués des éléments atomiques classiques comme *Video*, *Audio*, *Image*, *Text*, *Html*, etc. qui sont simplement des flux de données à présenter. Les éléments structurés

permettent de capturer à la fois des flux médias et la structuration interne de ces médias pour non seulement présenter le contenu des médias mais aussi exploiter ce contenu pour faire des compositions et des synchronisations plus fines.

Ces éléments structurés emploient les modèles de description du contenu des médias présentés dans la partie IV.2 ci-dessus. Leur syntaxe au sein de l'élément *C-Group* est la suivante :

```
...
<element name="C-Group"> <!— group de contenu -->
        <complexType>
            <sequence>
                < !—les quatre éléments structurés -->

                <element name="StructuredVideo" type="VideoContentType" minOccurs="0"
                        maxOccurs="unbounded"/>
                <element name="StructuredAudio" type="AudioContentType" minOccurs="0"
                        maxOccurs="unbounded"/>
                <element name="StructuredImage" type="ImageContentType" minOccurs="0"
                        maxOccurs="unbounded"/>
                <element name="StructuredText" type="TextContentType" minOccurs="0"
                        maxOccurs="unbounded"/>


                < !— les autres éléments atomiques classiques comme : image, text, video, audio, html, svg, etc.
sont définis ici –>
            </sequence>
        </complexType>
</element>
```

Par exemple, l'auteur veut composer un document multimédia à partir d'un ensemble de trois médias :

♦ un extrait de texte de la chanson Tarzan :

```
"Come stop your crying
It will be alright
Just take my hand
Hold it tight

I will protect you
from all around you
I will be here
Don't you cry ..."
```

♦ une audio de la chanson Tarzan : *TarzanYouBeInMyHeart.mp3*,

♦ une vidéo de mariage : *TienWeddingVideo.mov*.

Au lieu de déclarer ces ressources à travers des éléments atomiques classiques comme :

```
<C-Group> ...
        <Text ID="Text_2721">

            Come stop your crying
            It will be alright
            Just take my hand
            Hold it tight ...</Text>


        <Audio ID="Audio_3237" FileName="TienWedding/audios/TarzanYouBeInMyHeart.mp3" />
```

```
        <Video ID="VideoContent_17664"   FileName="TienWedding/videos/TienWeddingVideo.mov"/>
</C-Group>
```

l'auteur peut utiliser les éléments structurés de contenu pour spécifier de plus de la structure du contenu des médias comme suit :

```
<C-Group> ...
        <StructuredText  ID="Text_2721">

            Come stop your crying
            It will be alright
            Just take my hand
            Hold it tight
            ...

            <TexteStructure>
                <section ID="section_1 Begin_Char="0" End_Char="140">
                    <Paragraph ID="C_SonTarzanPara1" Begin_Char="0" End_Char="71">
                        ...
                        <Sentence ID="C_Sentence2" Begin_Char="41" End_Char="57"/>
                        ...
                    </Paragraph>...
                </section>...
            <TexteStructure>   </StructuredText>

        <StructuredAudio ID="Audio_3237" >
            <MediaManagement>
                <MediaLocator><MediaUri> TienWedding/audios/TarzanYouBeInMyHeart.mp3 </MediaUri>
                </MediaLocator>...
            <MediaManagement>
            ...
            <AudioStructure ID="C_TarzanAudioComplet" xsi:type="mpeg7:AudioSegmentType" >
                <TimeMedia>...</TimeMedia>
                <TemporalDecomposition>
                    ...
                    <AudioSegment ID="C_Extract1" xsi:Type="mpeg7:AudioSegmentType">
                        <TimeMedia>...</TimeMedia>
                        <TemporalDecomposition>...
                            <AudioSegment ID="DesAS2"> ... </AudioSegment>...
                        </TemporalDecomposition>
                    </AudioSegment>
                    ...
                <TemporalDecomposition> ... </AudioStructure> </StructuredAudio>

        <StructuredVideo ID="VideoContent_17664" >
            <MediaManagement>
                <MediaLocator><MediaUri> TienWedding/videos/TienWeddingVideo.mov</MediaUri>
                </MediaLocator>   ...
            <MediaManagement>
            ...
            <VideoSegment ID="WeddingVideoSTRUCTURE" xsi :type= "VideoStructureType" >
                <TimeMedia>...</TimeMedia>
                <TemporalSegmentDecomposition xsi:type="VideoStructureDecompositionType">
                    <VideoSegment ID="WeddingVideoSEQUENCE" ...> ...
                        <Tem poralSegmentDecomposition xsi:type="VideoSequenceDecompositionType">
                            <VideoSegment ID="WeddingVideoSCENE" ...>
                                <TemporalSegmentDecomposition xsi:type="VideoSceneDecompositionType">
                                    ...
                                    <VideoSegment ID="RockingChairSHOT" ...>
                                        <TemporalSegmentDecomposition
                                            xsi:type="VideoShotDecompositionType">
                                        <MovingRegion ID="Mariée" xsi:type="OccurrenceType"> ...
                                        </MovingRegion>
                                        <MovingRegion ID="Marié" xsi:type="OccurrenceType"> ...
```

```
                                    </MovingRegion>
                                </TemporalSegmentDecomposition>
                            </VideoSegment> </TemporalSegmentDecomposition>
                        </VideoSegment> </TemporalSegmentDecomposition>
                    </VideoSegment></TemporalSegmentDecomposition></VideoSegment>
        </StructuredVideo>
</C-Group>
```

Dans cet exemple (l'exemple complet est donné en annexe B), le texte est structuré en sections, paragraphes et phrases ; l'audio est décrite via des segments correspondant aux sections, paragraphes et phrases du texte ; la vidéo du mariage est décomposée logiquement en une séquence qui contient une scène se décomposant en quatre plans et, dans chaque plan, des occurrences de personnages.

Les descriptions de contenu peuvent être résultat d'une analyse automatique de média comme savent produire les outils vus au chapitre IV aussi faire l'objet d'une édition manuelle (cf. chapitre VI) par l'auteur, qui ne va dans ce cas, spécifier que les descriptions utiles pour la composition du document qu'il est en train de réaliser.

V.3.2 Le sous-acteur (SubActor), extension de la partie d'acteur (A-Group)

Lors de la rédaction d'un document multimédia, l'auteur doit avoir la capacité de spécifier des actions ou des informations de style de présentation sur des fragments de média, comme mettre en valeur (*highlight*) un segment de texte, mettre un suivi d'objet (*tracking*) dépistage, un hyperlien sur une région mobile d'un segment vidéo, etc. Un sous-élément de l'élément *DefActor* appelé sous-acteur (*DefSubActor*) est prévu à cette fin. Ce sous-élément est chargé de représenter un fragment média comme un objet sur lequel l'auteur peut facilement spécifier des actions ou des styles de présentation. L'élément *DefSubActor* peut représenter logiquement un segment significatif décrit dans la partie *Content* ci-dessus en utilisant l'attribut *Content* pour référencer ce segment. L'attribut *Content* d'un sous-acteur peut prendre une valeur qui est un identificateur d'une description d'un segment média ou une expression *XPath* qui pointe sur une description d'un fragment média. Dans le cas où l'attribut *Content* n'est pas spécifié, le sous-acteur contient directement la spécification du segment de média sous forme de coordonnées absolues. Plus précisément, le sous-élément *SubActor* enrichi l'élément *DefActor* de la partie *A-Group* du modèle Madeus selon la façon suivante :

```
...
<element name="A-Group"> <!—group d'acteur -->
        <complexType>
            <sequence>
                ...
                <element name="DefActor"> ...
                    <complexType>
                        <sequence>
                            <element name="DefSubActor">

                                <complexType>
                                    <choice maxOccurs="unbounded">
                                        <element name="HighLight" minOccurs="0" ...>... </element>
                                        <element name="Tracking" minOccurs="0" ...>... </element>
```

```
                                    <element name="Hidden" minOccurs="0" ...> ... </element>
                                    <element name="toolTip" minOccurs="0" ...> ... </element>
                                    <element name="Link" minOccurs="0" ...> ... </element>
                                </choice>
                                <attribute name="ID" type="string" use="required" />
                                <attribute name="Content" type="string" use="optional" />
                            </complexType>
                        </element>

                    </sequence>
                </complexType>
            </element>
        </sequence>
    </complexType>
</element>
```

Par exemple, pour marquer (*highlight*) un segment du texte et lui associer un hyperlien, l'auteur peut spécifier :

```
< A-Group> ...
    <DefActor ID="DefActor_2725" LinePaint="rgb(128,0,255)" FontFamily="Comic Sans MS" FontSize="16"
Content="#Text_2721" VScroll="true"> ...
        <DefSubActor ID=" SubActorSentence2 " Content="#C_Sentence2">
            <Highlight color=" rgb(28,100,55)"/>
            <Link timeref="TarzanSongAudio.DesAS2@begin"/>
        </DefSubActor>
    </DefActor>
    ...
<A-Group>
```

Lorsque le lien sur le segment du texte défini ci-dessus est activé, la présentation courante du document multimédia est interrompue pour lancer la présentation à l'instant correspondant au début (*begin*) du segment audio *#DesAS2* de l'intervalle *TarzanSongAudio*.

Les descriptions suivantes définissent un acteur (*DefActor_25101*) qui représente une extraction de la vidéo : c'est la scène *#WeddingVideoSCENE*. Cet acteur contient un sous-acteur (*SubActorWeddingSceneEx*) qui représente l'occurrence *#Mariée* au sein de la scène. De ce fait, l'auteur spécifie des actions (*hyperlien* et *suivi d'objet*) sur l'occurrence de la scène.

```
...
<DefActor ID="DefActor_VidéoMariage" Content="#WeddingVideoSCENE">
    <DefSubActor ID="Follow Mariée" Content="#Mariée">
        <Stracking color="red" stroke="1"/>
        <Link href="Mdefi/StructuredVideo/MariéePresentation.madeus "/>
    </DefSubActor>
</DefActor>
```

V.3.3 Le sous-intervalle (*SubInterval*), extension de la partie temporelle

Le modèle Madeus présente des acteurs dans le temps et dans l'espace au travers d'éléments intervalle et région au sein de la structure temporelle et spatiale du document. Nous proposons d'étendre la structure temporelle au moyen de l'élément sous-intervalle (*SubInterval*) afin de représenter des sous-acteurs dans le temps.

Un sous-intervalle est d'abord un intervalle temporel dont la définition a été précisément établie depuis longtemps [Allen 83], [Little et al. 93]. Un sous-intervalle possède donc les attributs temporels "classiques" : *début*, *durée* et *fin* ; il peut être utilisé dans des relations temporelles avec d'autres intervalles déployé pour composer un scénario temporel. Par contre, à la différence d'un intervalles, un sous-intervalle est toujours un composant d'un intervalle père : il doit respecter la relation ***during*** [Allen 83] avec l'intervalle qui le contient (Si *A* ***during*** *B*, *A*.début > *B*.début AND *A*.fin =< *B*.fin). La syntaxe d'un sous-intervalle est :

```
...
<element name="T-Group"> <!— group de temporel –>
        <complexType>
            <sequence>     ...
                <element name="Interval">
                    <complexType>
                        <sequence>
                            <element name="SubInterval">
                                <complexType>
                                    <attribute name="ID" type="string" use=" required" />

                                    <attribute name="SubActors" use="optional" >...</attribute>
                                    <attribute name="Animations" use="optional" >...</attribute>
                                    <attribute name="ActOn" use="optional" >...</attribute>

                                    <attribute name="Begin" use="default" valu e="pref:0ms" >...</attribute>
                                    <attribute name=" Duration " use="default" value="pref:10m s">...</attribute>
                                </complexType>

                            </element>

                        </sequence> ...
                    </complexType>
                </element>
            </sequence>    ...
        </complexType>
</element>
```

Un sous-intervalle peut référencer un ou plusieurs sous-acteurs ou des animations (l'animation sera détaillée dans la section suivante V.4) qui sont alors tous présentés ou activés pendant la même période. Les attributs *SubActors* et *Animations* sont utilisés pour spécifier les sous-acteurs et animations animés par le sous-intervalle.

Nous identifions deux types de sous-intervalles temporels : les sous-intervalles **passifs** et les sous-intervalles **actifs**. En fait, comme les médias, les segments de média peuvent être soit des segments statiques comme une phrase de texte, une région d'image, etc. dont le contenu n'est pas changé dans le temps, soit des segments continus comme un segment de vidéo ou d'audio, dont le contenu évolue dans le temps. Un sous-intervalle qui représente un segment non temporel est un sous-intervalle passif et celui qui représente un segment temporel est un sous-intervalle actif.

Les informations temporelles d'un **sous-intervalle passif** doivent être spécifiées explicitement par l'auteur à travers des attributs temporels (*Begin* et *Duration*) ou une relation temporelle. Par exemple, dans l'extrait ci-dessous, la ligne numérotée 5 définit un sous-intervalle passif représentant une phrase marquée

(*highlighted*) du texte. Dans ce cas, l'auteur spécifie les informations temporelles pour ce sous-intervalle par la relation temporelle *Equals* présentée ligne10.

```
...
1  <T-Group ID="Temporal Root" Duration="pref:153121ms">
2        <T-Group ID="FirstPart" Duration="pref:51121ms" ... >
3             <Interval ID="TarzanSongText" Actor="#DefActor_2725">
4                  ...
5                  <SubInterval ID="HighLightSentence2" SubActor="#SubActorSentence2 "/>
6                  ...
7             </Interval>
8             <Interval ID="TarzanSongAudio" Actor="#DefActor_3240"/>
9             ...
10            <Equals Interval1="#TarzanSongAudio.DesAS2" Interval2="# HighLightSentence2"/>
11            ...
12       </T-Group>
13       <T-Group ID="SecondPart">
14            <Interval ID="Interval_VidéoMariage " Actor="#DefActor_VidéoMariage" Begin="pref:1000ms">
15                 ...
16                 <SubInterval ID="SubIntervalFollowMariée" SubActor="#FollowMariée"/>
17            </Interval>
18            <Interval ID="IntervalTxtMotion" Actor="#TxtMotion">...</Interval>
19            ...
20            <Equals Interval1="# SubIntervalFollowMariée " Interval2="#IntervalTxtMotion "/>
21            ...
22       </T-Group>
23       ...
24 </T-Groug>
...
```

A l'inverse, **le sous-intervalle actif** n'a besoin d'aucune spécification explicite des informations temporelles ; plus précisément l'auteur ne peut pas spécifier des informations temporelles pour le sous-intervalle actif au moyen d'attributs ou de relations temporelles car il hérite automatiquement des informations temporelles intrinsèques du segment média qu'il représente. Le sous-intervalle est utilisé comme un repère dans des relations temporelles pour propager les informations aux autres intervalles et sous-intervalles passifs. Dans l'extrait ci-dessous, la ligne 16 définit un sous-intervalle actif *#SubIntervalFollowMariée* il référence le sous-acteur *#FollowMariée* qui représente l'occurrence *#Mariée* de la vidéo. Les lignes 10 et 20 (ci-dessous) montrent l'utilisation de sous-intervalles actifs (*#TarzanSongAudio.DesAS2* et *#SubIntervalFollowMariée*) pour synchroniser la présentation de *#HighLightSentence2* et *#IntervalTxtMotion*.

Le résultat des compositions ci-dessus est : la deuxième phrase *#C_Sentence2* (temporalisé par le sous-intervalle *#HighLightSentence2*) du texte de la chanson Tarzan sera coloré (*highlighted*) pendant le temps où le segment audio *#DesAS2* (temporalisé par le sous-intervalle *#TarzanSongAudio.DesAS2*) de la chanson est joué ; et un texte (intervalle *#IntervalTxtMotion*) sera présenté pendant l'apparition de l'occurrence *#Mariée* (le sous-intervalle *#SubIntervalFollowMariée*) de la vidéo.

Nommage par combinaison des noms. Notre modèle fournit le moyen de créer dynamiquement des sous-intervalles actifs sans nécessiter de spécification explicite pour synchroniser des intervalles ou des sous-intervalles passifs. Par exemple, la ligne 10 spécifie une synchronisation *Equals* entre *#TarzanSongAudio.DesAS2* (actif) et *#HighLightSentence2* (passif). L'intérêt de cet exemple est que le sous-intervalle actif *#TarzanSongAudio.DesAS2* est un sous-intervalle de l'intervalle

#TarzanSongAudio défini en ligne 8, mais il n'est pas défini explicitement dans le contenu de *#TarzanSongAudio* (le contenu de cet intervalle n'est pas spécifié). Le sous-intervalle est créé et inséré implicitement dans le contenu de l'intervalle *#TarzanSongAudio* lorsque l'analyseur du document rencontre pour la première fois le nom *#TarzanSongAudio.DesAS2*. Pour former dynamiquement un sous-intervalle actif, l'auteur combine **l'identificateur d'un intervalle** (*#TarzanSongAudio* dans l'exemple) représentant un média continu structuré ou un segment décrivant un média continu avec **l'identificateur d'un sous-segment temporel** (*#DesAS2* dans l'exemple) décrit dans le média structuré ou dans la description du segment. Une telle spécification permet de créer automatiquement des sous-intervalles. En fait l'identificateur du sous-intervalle créé est la combinaison de deux identificateurs ; le père de ce sous-intervalle est déterminé facilement par la première partie de la combinaison ; les informations temporelles de ce sous-intervalle sont reprises de la description du segment qui est identifié facilement grâce à la deuxième partie de la combinaison.

V.3.4 La sous-région (*SubRégion*), extension de la partie spatiale

En suivant le même principe, l'élément sous-région (*SubRegion*) est créé dans la structure spatiale du modèle Madeus pour représenter une portion spatiale d'un média visuel. L'identification de l'élément de sous-région fournit les moyens de spécifier des relations spatiales plus sophistiquées et plus fines avec d'autres régions des autres médias visuels.

Précisément la sous-région est utilisée pour spécifier la présentation spatiale d'une portion du document multimédia. Elle possède donc tout les attributs particuliers de la région comme l'attribut de positionnement (*left*, *top*) et l'attribut de dimensionnement (*width*, *height*). De plus, dans la plupart des modèles actuels, la région est limitée à une forme de rectangle. En réalité, la forme d'un objet est plus complexe ; la sous-région est donc étendue avec l'attribut *Paths* (de la description spatiale de média) qui permet de décrire une forme complexe permettant de représenter richement les objets spatiaux. La sous-région doit être définie au sein du contenu d'une région et doit respecter la contrainte d'encapsulation (*inside*) avec cette région (Si $région_A$ ***inside*** $région_B$, $région_A$ Í $région_B$). La syntaxe étendue de la sous-région dune partie spatiale est la suivante :

```
...
<element name="S-Group"> <!—group de spatial -->
        <complexType>
            <sequence>     ...
                <element name="Region">
                    <complexType>
                        <sequence>
                            <element name="SubRegion">
                                <complexType>
                                    <attribute name="ID" type="string" use="required" />
                                    <attribute name="Left" use="default" value="0">...</attribute>
                                    <attribute name="Top" use="default" value="0">...</attribute>
                                    <attribute name="Width" use="default" value="10">...</attribute>
                                    <attribute name="Height" use="default" value="10">...</attribute>

                                    <attribute name="SubActors" use="required">...</attribute>
                                    <attribute name="Paths" use="optional">...</attribute>
```

```
                    </complexType>

                </element>

            </sequence> ...
        </complexType>
    </element>
  </sequence>   ...
</complexType>
</element>
```

Une sous-région peut représenter logiquement un objet de média à travers la référence vers une description de cet objet (*Occurrence*/*MovingRegion* ou *StillRegion*). Ce type de sous-région prend automatiquement les informations de la description du segment spatial qu'il représente. La référence à un segment décrit est réalisée via un sous-acteur par l'attribut *SubActors* ; par exemple, la ligne 6 de l'extrait ci-dessous présente la définition d'une sous-région par référence au sous-acteur *#FollowMariée* ce sous-acteur porte une action de hyperlien et une action de suivi d'objet *#Mariée* (voir la définition du sous-acteur *#FollowMariée* dans la section V.3.2).

Une sous-région doit être définie de façon plus complète lorsque le sous-acteur qu'il référence est défini sans référence à un segment de contenu. L'auteur doit alors spécifier explicitement les informations spatiales nécessaires à sa représentation dans l'espace.

Dans l'exemple ci-dessous, la sous-région *#FollowMariéeRegion* est utilisée pour spécifier deux synchronisations spatiales fines avec la région du média texte *#TxtMotionRegion* (lignes 9 et 10). Ces relations spatiales alignent la présentation du texte avec le coin en haut et à gauche de la sous-région *#FollowMariéeRegion*. L'occurrence *#Mariée* étant une région mobile, les coordonnées spatiales de la sous-région *#FollowMariéeRegion* évoluent dans le temps. Grâce aux deux relations spatiales (lignes 9 et 10), la position de la région du texte *#TxtMotionRegion* est aussi mise à jour, ce qui donne un effet intéressant : le texte suit dynamiquement l'occurrence.

```
1 ...
2 <S-Group ID="Spatial Root" Height="564" Width="999">
3       ...
4       <Region Actor="#DefActor_VidéoMariage" Left="238" Top="169">
5           ...
6           <SubRegion SubActors="# Follow Mariée" ID="Follow MariéeRegion"/>
7       </Region>
8       <Region Actor="#TxtMotion" ID="TxtMotionRegion" />
9       <Top _align Region1="#Follow MariéeRegion" Region2="#TxtMotionRegion "/>
10      <Left_align Region1="#Follow MariéeRegion" Region2="#TxtMotionRegion "/>
11      ...
12 </S-Group>
```

V.3.5 Modèle complet

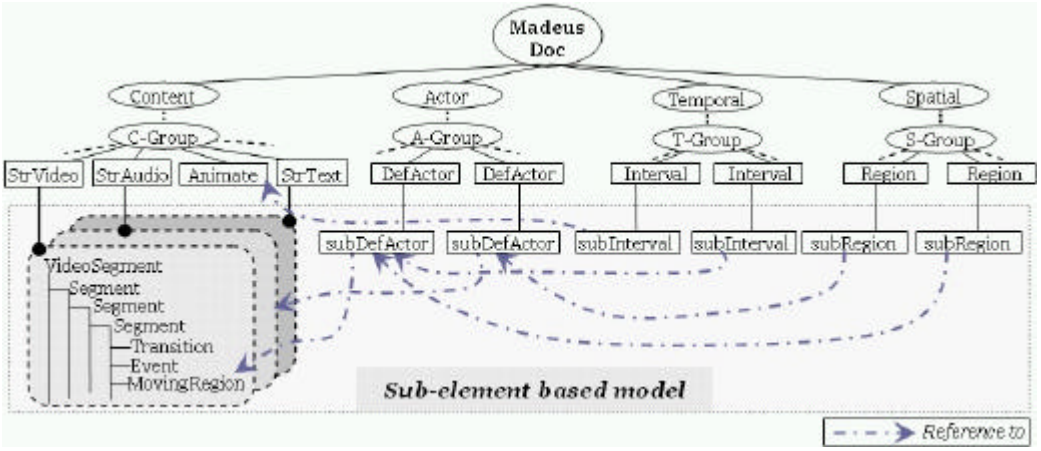

Figure 53. Structure des sous-éléments dans le modèle de Madeus.

La figure ci-dessus résume la définition de la structure des sous-éléments et la liaison entre eux. Pour conclure, un sous-élément appartient toujours à un élément et possède une relation contrainte avec cet élément pour exprimer sa dépendance sémantique dans la dimension correspondante.

V.3.6 Evaluation

La plupart des modèles multimédia existants ne manipulent pas sémantiquement les portions de média. Lorsqu'ils permettent leur accès, c'est seulement au moyen d'attributs de bas niveau comme des instants temporels, des coordonnées spatiales (voir les sections III.4.4, III.4.5, III.4.6). Notre approche peut être comparée d'une part avec la technique de l'ancre (*anchor*) et de région (*area*) des langages existants (par exemple SMIL) en ce sens qu'elle autorise la spécification absolue des segments médias. Cependant, notre approche permet aussi la spécification de fragments sémantiques en utilisant des descriptions de contenu des médias. De plus, avec la décomposition en sous-éléments (médias structurés, *DefSubActor*, *SubInterval* et *SubRegion*), notre approche est plus puissante car elle permet de multiples usages : mettre des hyperliens, des actions et des styles de présentation sur un segment de média ; créer des synchronisations spatiales, temporelles et spatio-temporelles fines entre des segments de média ou entre des segments de média et des médias entiers.

De plus, par l'utilisation des descriptions du contenu, le modèle crée un pont avec des applications d'indexation multimédia. Un tel lien avec les applications multimédias permet dans un premier temps de renforcer la capacité de spécification des synchronisations fines, tout en assurant le maintien de structures sémantiques dans ces synchronisations, ce qui ne peut pas être réalisé par l'approche absolue. L'exploitation plus profonde des techniques propres à ce domaine, comme l'indexation sémantique du contenu, les liens entre différentes indexations, ou bien à un plus haut niveau, les liens entre les modèles, pourrait ouvrir de nombreuses voies dans le domaine de l'édition et de la présentation du document multimédia.

La structure des sous-éléments est aussi utilisée pour représenter un modèle abstrait d'animation. Nous présentons ce modèle d'animation dans la section V.4 suivante.

V.4 Modèle d'animation

Bien que l'animation soit prise en compte dans la plupart des modèles de document multimédia existants, sa puissance est encore limitée (voir la section III.4.7). Les objectifs du modèle d'animation présenté dans cette section sont non seulement la *spécification* des animations mais aussi la *réutilisation* des animations et la *simplification* de la maintenance des documents multimédias. Pour ces objectifs nous choisissons l'approche qui consiste à définir des *animations abstraites, indépendantes* de l'*objet* animé et du *temps* concret.

Le langage d'animation proposé ici se veut un langage hôte pour l'animation de SMIL qui offre un niveau plus haut d'abstraction et donc plus de flexibilité dans la spécification, la réutilisation et la modification. Vodislav a montré [Vodislav 95] comme obtenir une telle flexibilité en créant des animations abstraites sur un objet graphique. L'animation concrète de l'objet est produite quand le point initial de la trajectoire réelle est déterminé. Dans VRML [VRML], un interpolateur est défini indépendamment sur un intervalle abstrait [0,1]. Une animation sur un objet est donnée lorsque l'interpolateur appliqué à un intervalle concret et les valeurs interpolées affectées à un attribut de l'objet animé. Cependant, nous allons plus loin dans cette approche de la création d'animation abstraite. Une animation dans notre modèle peut être appliquée sur différents objets médias concrets au lieu d'être appliquée seulement sur un objet comme dans le modèle de Vodislav, ou sur différents intervalles au lieu d'être appliquée seulement sur un intervalle comme dans l'interpolateur VRML.

V.4.1 Modèle d'animation abstrait

Notre approche pour l'animation abstraite suit la même idée que les feuilles d'animation (*cascading animation sheets*[17]). Une animation abstraite est un raffinement des animations de base de SMIL (voir la section III.4.7). Ce raffinement consiste à détacher l'objet animé et le temps concret des animations de base.

Le principe du modèle est présenté dans la Figure 54.

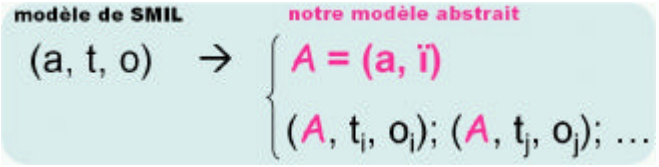

Figure 54. Modèle d'animation abstrait à partir du modèle d'animation de SMIL

Où le ***a*** représente un des animateurs basiques de SMIL {*set*, *animate*, *animateColor* et *animateMotion*}. Cependant au lieu d'intégrer l'animateur (*a*), l'objet animé (o) et le temps (t) dans une seule animation, nous proposons de définir une abstraction A composée d'un animateur et d'un intervalle temporel abstrait ï=[0,1] ; dans un temps concret (les intervalles t_i ou t_j), une animation réelle sur un objet (o_i ou o_j) peut être spécifiée par référence à l'animation abstraite et à l'objet animé (voir la Figure 55). Ainsi plusieurs animations concrètes peuvent être

[17] Proposals for additions to the SVG-specification, pro http://www.pinkjuice.com/SVG/spec-prop.xhtml

créées par réutilisation d'une animation abstraite. La Figure 55 résume les composants de ce modèle. Lorsque A abstraite est utilisée dans un intervalle concret, l'intervalle ï est projeté sur cet intervalle. La Figure 56 illustre une telle projection.

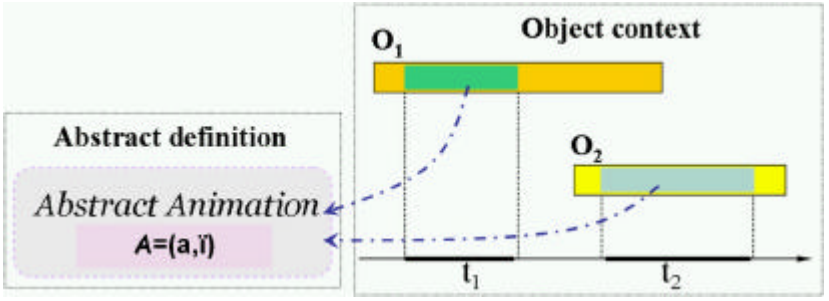

Figure 55. Représentation graphique de notre modèle d'animation abstraite.

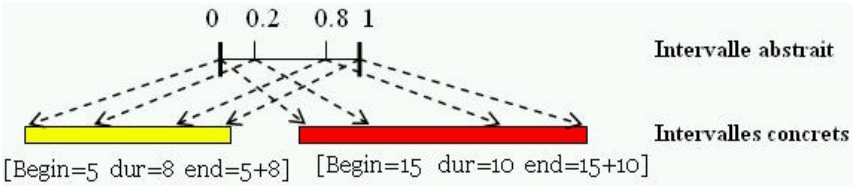

Figure 56. Intervalle abstrait projeté sur deux intervalles concrets.

On peut noter que l'intervalle abstrait [0,1] est également utilisé dans le modèle d'interpolation de VRML. Cependant, l'interpolateur VRML reste lui-même un objet concret qui ne peut pas être utilisé à la fois sur plusieurs intervalles concrets. Dans notre modèle, il est possible de réutiliser une animation abstraite sur plusieurs intervalles et pour animer plusieurs objets. De plus, notre modèle hérite partiellement du modèle d'animation de SMIL qui permet de spécifier richement des animations, par exemple, le mode de calcul (*calcMode*) peut être *discrete*, *linear*, *paced* ou *spline* au lieu d'être seulement linéaire comme dans VRML ; de même, les valeurs d'interpolation peuvent être relatives ou absolues (*additive*={*sum*, *replace*}).

V.4.2 Représentation du modèle dans Madeus

On peut considérer la définition indépendante des animations dans notre modèle comme reflétant la distinction des deux axes : *Content* et *Temporel* du modèle Madeus. Ainsi notre modèle d'animation peut être facilement traduit en Madeus. La spécification d'animation dans un document Madeus selon notre modèle est donc effectuée au travers de deux étapes : la spécification d'animations abstraites et la spécification concrète de leur application (voir la Figure 58).

- La *spécification abstraite* définit les animations abstraites qui sont indépendantes de la présentation. Cette définition est un raffinement de l'animation SMIL dans lequel les attributs temporels (*begin*, *dur* et *end*) et l'attribut *targetElement* sont éliminés. Le plus important dans cette partie est la spécification d'un intervalle [0, 1] sur lequel des points temporels abstraits correspondant aux positions sur la trajectoire, aux échelles, aux états de

transition, etc. sont définis. Puisqu'une telle définition abstracte d'animation est indépendante de tout espace de présentation (spatial et temporel), il est naturel qu'elle apparaisse dans la partie *Content* du modèle de Madeus. Les extraits de code ci-dessous définissent deux animations abstraites pour augmenter (resp. diminuer) linéairement la valeur de l'attribut *AlphaSource* de 0 (1.0) à 1.0 (0). L'attribut *AlphaSource* définit la transparence d'un objet visible (texte, image, vidéo, etc.) ; des telles animations permettent de faire apparaître (resp. disparaître) graduellement la présentation d'un objet visuel.

```
...
<C-Group>
    ...
    <Animate ID="UpAlphaAnimation"attributeName="AlphaSource" values="0;1.0" keyTimes="0;1"
calcMode="linear"
            additive="replace"/>
    <Animate ID="DownAlphaAnimation" attributeName="AlphaSource" values="1.0;0" keyTimes="0;1"
calcMode="linear"
            additive="replace"/>
    ...
</C-Group>
```

♦ La *spécification concrète* instancie l'animation abstraite par la spécification du temps concret dans lequel l'animation est activée ainsi que les objets cibles qui sont concernés. En fait, selon le modèle Madeus une ressource d'information déclarée dans la partie *Content* peut être utilisée plusieurs fois dans le document multimédia, une animation abstraite peut donc être réutilisée dans plusieurs animations spécifiques pour animer plusieurs objets à des instants différents. Une animation sur un objet est toujours activée **pendant** la présentation de cet objet. C'est pourquoi un **sous-intervalle** est approprié à représenter l'animation sur l'objet. L'attribut *targetElement* est employé pour explicitement identifier l'élément de média cible qui est animé. Par exemple, les extraits de code ci-dessous spécifient les transitions entre deux présentations successives d'une liste d'images, réalisées par l'utilisation de deux animations augmentant diminuant la transparence au début et à la fin de la présentation de chaque image. Pour cet exemple, l'auteur définit dans chaque intervalle de représentation de l'image deux sous-intervalles (*Interval_UpAlphaAniImage*N et *Interval_DownAlphaAniImage*N) qui correspondent aux deux phases de l'animation complète. La liste d'images peut être longue, mais le nombre d'animations abstraites utilisées reste seulement de deux (*UpAlphaAnimation* et *DownAlphaAnimation*). Cet exemple illustre l'intérêt de notre approche au niveau de la réutilisation. L'avantage est encore plus net dans le cas où l'auteur veut changer, par exemple, la valeur de début de l'intervalle de l'augmentation de la transparence "0.3 ;1.0" au lieu de "0 ;1.0", dans ce cas, l'auteur doit changer simplement la valeur de l'attribut *values* de l'élément *UpAlphaAnimation* défini dans la partie *Content*.

```
...
</T-Group>
    ...
    <Interval ID="Interval_Image1" Actor="DefActor_12093" Duration="pref:2914ms"
Begin="pref:43130ms">
```

```
        <SubInterval ID="Interval_UpAlphaAniImage1" Animate="UpAlphaAnimation"
Duration="pref:1000ms"/>
        <SubInterval ID="Interval_DownAlphaAniImage1" Animate="DownAlphaAnimation"
Begin="pref:1914ms" Duration="pref:1000ms"/>
    </Interval>
    ...
    <Interval ID="Interval_Image12" Actor="DefActor_12094" Duration="pref:3039ms"
Begin="pref:46044ms">
        <SubInterval ID="Interval_UpAlphaAniIm age12" Animate="UpAlphaAnimation"
Duration="pref:1000ms"/>
        <SubInterval ID="Interval_DownAlphaAniImage12" Animate="DownAlphaAnimation"
Begin="pref:2039ms" Duration="pref:1000ms"/>
    </Interval>
    ...
</T-Group>
```

Figure 57. La vue temporelle graphique de la spécification de l'exemple ci-dessus.

A noter que le sous-intervalle qui représente une animation est un sous-intervalle passif, i.e., l'auteur doit spécifier les informations temporelles pour ce sous-intervalle, soit par la spécification directe des attributs temporels, soit par les relations temporelles.

Notre approche peut être comparée à l'abstraction de construction de médias (MCA, Media Construction Abstraction) proposé par Nanard [Nanard 01]. MCA vise à aider des utilisateurs dans leur processus de conception par des diagrammes de MCF où la composition de médias permet des définitions à base d'événement. Pour cela, il emploie le paradigme de connecteur et de boîte. Au contraire, notre définition d'animation repose sur une spécification à base d'intervalles où la composition résulte de relations entre des intervalles. Elle permet la spécification de synchronisations par relations et non par événement. Par contre notre modèle n'offre qu'un niveau d'abstraction tandis que MCA offre un mécanisme d'encapsulation et de construction beaucoup plus riche.

La Figure 58 présente une synthèse structurelle du modèle d'animation abstrait implantée dans le modèle Madeus.

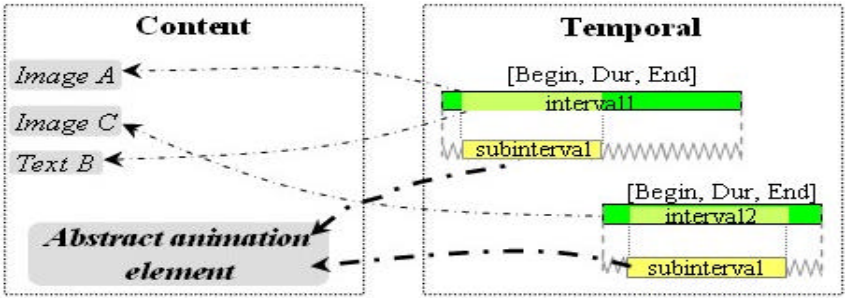

Figure 58. Structure d'animation implantée dans le modèle Madeus.

V.5 Synthèse

Dans ce chapitre nous proposons un modèle qui permet d'accéder plus finement et plus sémantiquement au contenu de médias dans l'environnement d'édition et présentation de documents multimédias. Le modèle est un mariage de deux domaines de modélisation multimédia : indexation et intégration. L'application d'indexation est donc étendue pour non seulement de la recherche d'information multimédia mais aussi pour son intégration fine. Au niveau de l'intégration, notre modèle permet non seulement de surmonter les limites de la synchronisation entre les médias mais aussi apporte la promesse de nombreuses ouvertures, comme former automatiquement une présentation multimédia à partir d'une requête.

Dans notre l'approche de description du contenu des médias, nous proposons un modèle approprié à la composition du document multimédia et qui prend en considération les besoins de ce domaine. Nous proposons par ailleurs l'emploi de la norme MPEG-7 pour exprimer ces modèles de descriptions. De ce fait, nos spécifications sont non seulement standardisées mais aussi interopérables. Une extension "souple" du modèle Madeus est également décrite afin d'utiliser la description du contenu de média dans ce modèle particulier. Les parties *Content*, *Actor*, *Temporal et Spatial* du modèle sont donc enrichies par des structures de sous-éléments. Plus précisément, dans la partie *Content*, nous proposons d'inclure la description du contenu des médias dans la déclaration des ressources. De ce fait, nous créons de nouveaux éléments de médias structurés qui contiennent non seulement le flux de données mais aussi la structure de ces données. Nous avons montré que cette approche permet de dépasser ce que nous considérons comme la cause première du traitement limité du contenu des médias dans les documents. Les autres sous-éléments dans la structure étendue (le *sous-acteur*, le *sous-intervalle* et la *sous-région*) permettent de représenter les descriptions du contenu de façon à adapter la structure du document aux besoins de la composition fine.

Cependant, la modélisation du contenu de média, dans le cadre de cette thèse, s'est concentrée seulement sur le premier niveau, la description de la structure logique. Bien que celle-ci soit la plus importante pour la composition du document multimédia, des expérimentations plus poussées avec des modélisations de plus haut niveau (sémantique, thésaurus, etc.) pourraient donner plus de puissance à l'intégration multimédia.

Il est intéressant de remarquer que la puissance d'abstraction et la souplesse de la structure des sous-éléments est aussi employée pour décrire le modèle d'animation abstrait dans lequel une animation est décrite en deux parties :

spécification abstraite et *spécification concrète* qui correspondent aux deux parties *Content* et *Temporal* du modèle Madeus. Ce travail a prouvé que la spécification abstraite d'animation est possible et réaliste pour des documents déclaratifs sous forme XML. Le travail raffine l'ensemble des animations de base de SMIL pour les rendre plus abstraites. De plus, l'emploi de l'élément sous-intervalle pour exprimer l'animation spécifique permet de définir l'organisation temporelle de façon soit absolue soit relative.

Dans le chapitre suivant nous décrivons un environnement auteur qui est implémenté expérimentalement à partir de ce modèle.

Chapitre VI. Mdéfi : un Environnement auteur pour l'intégration fine de média

VI.1 Introduction

Dans le Chapitre I. nous avons mentionné une architecture de l'environnement auteur plus confortable qui propose de lier l'indexation multimédia avec l'intégration multimédia. L'architecture doit être basée sur un modèle étendu présenté dans le Chapitre V qui permet d'employer des descriptions du contenu des médias dans la composition du document multimédia. Dans ce chapitre nous présentons un prototype d'environnement auteur selon une telle architecture.

Dans ce mémoire, nous appuyons sur les travaux réalisés avant nous sur les outils MADUES et dont les descriptions se trouvent dans [Layaïda 97] [Sabry 99] [Tardif 00] [Villard 02].

Nous ne discutons donc que des extensions du prototype Madeus que nous avons réalisées pour adapter ce système à notre nouveau modèle. L'organisation du chapitre est la suivante : nous présentons premièrement le principe de l'outil expérimental. Puis pour faciliter la description de nouveau système, nous présentons rapidement l'architecture principale de l'outil sous jacent Madeus. Le cœur du chapitre est consacrée à la description détaillée du prototype Mdéfi : modèle d'objet interne, gestionnaires de présentation, édition dans les différentes vues et édition des médias structuré.

VI.2 Principes du système

La réalisation d'une présentation multimédia utilise souvent directement des médias (la flèche *a* dans la Figure 59). De ce fait la composition est limitée à l'intégration à gros grains entre des médias. Nous avons intégré dans le système d'édition deux nouveaux modules (voir la Figure 59) qui permettent d'améliorer la composition au niveau de fragment de média.

Plus précisément la Figure 59 présente le principe de l'environnement auteur pour l'intégration fine de média. Le premier module permet d'analyser automatiquement le contenu des médias et produit ensuite une première version de description de contenu de médias. Dans ce module, un processus d'interprétation produit des descriptions conformément au modèle de description du contenu de média. Le deuxième module comprend des outils d'édition des descriptions du contenu de média. Ce module prend des descriptions produites automatiquement ou

des descriptions existantes en entrée et permet à l'auteur de les compléter ou de les modifier manuellement. Finalement, les médias structurés (la flèche *b* dans la Figure 59), qui synthétisent non seulement le flux du contenu de média mais aussi des descriptions de ce contenu, sont disponibles pour des compositions dans le document multimédia. Ce schéma de création de documents multimédias présente non seulement la capacité de générer en temps réel des descriptions de contenu de médias mais aussi la capacité de réutiliser des descriptions existantes décrites par des standards comme MPEG 7.

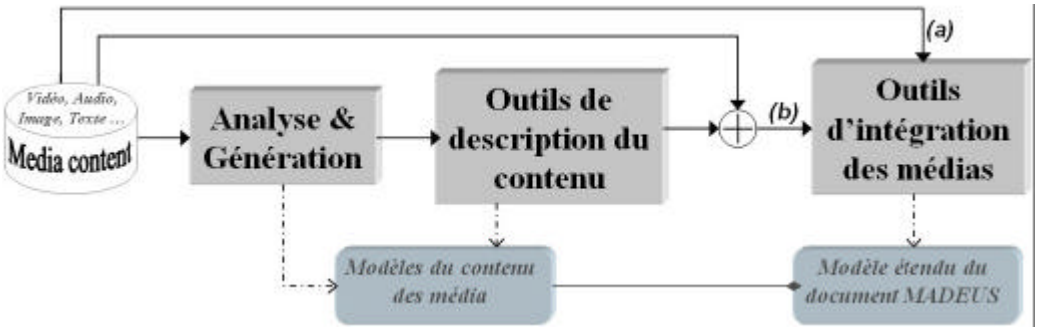

Figure 59. Schéma d'édition pour document multimédia avec médias structurés.

Dans la suite de ce chapitre nous présentons l'application basée sur ce schéma. L'environnement expérimental, appelé Mdéfi (*Multimédia Description and Fine-grained Intégration*), est un environnement auteur étendu de l'environnement auteur Madeus.

VI.3 Principe de l'architecture de l'outil sous jacent Madeus

La Figure 60 présente une architecture globale de l'outil auteur Madeus. L'outil peut charger un document Madeus, l'analyser et créer un document interne correspondant au document entré. Dès que le document interne est créé, la gestion des vues de l'outil peut l'utiliser pour créer des vues de ce document Madeus. Par défaut, la vue de présentation du document est créée. Lorsque la vue de présentation est disponible l'utilisateur peut jouer le document. A travers la vue de présentation l'auteur peut aussi éditer la disposition spatiale des médias par manipulation directe.

L'auteur peut choisir d'ouvrir d'autres vues pour découvrir le document dans les dimensions différentes que la présentation. La vue temporelle représente graphiquement le scénario temporel du document Madeus. Elle permet aussi d'éditer le scénario temporel par manipulation directe des objets graphiques. La vue hiérarchique représente le document sous forme arborescente, qui permet à l'utilisateur de naviguer facilement dans le contenu du document. L'utilisateur peut modifier, insérer ou supprimer un nœud dans l'arbre. Les trois vues : présentation, temporelle et hiérarchique sont les trois vues principales les plus importantes de l'outil auteur. Dans le cadre d'autres prototypes, d'autres vues ont été développées expérimentalement comme : la vue textuelle qui présente directement le contenu textuel du document ; la vue de contenu qui présente tous les médias utilisés dans une table des médias ; ou bien la vue de source XML et la vue de feuille de transformation XSLT pour éditer la transformation d'un document XML vers un

document Madeus [Villard 02] ; etc. En principe l'outil propose une plate-forme ouverte multi vues.

Les vues de l'outil sont synchronisées grâce à un système d'événement. Toutes les vues s'abonnent à des événements de la gestion du document interne. Lorsque l'utilisateur manipule sur une vue, un événement est notifié au gestionnaire du document interne à travers la gestion des vues. Si l'événement est reconnu au niveau de la gestion des vues, il peut être stoppé et traité tout de suite. Un exemple d'événement traité par une vue est le double clic sur un élément composite dans la vue temporelle ou la vue hiérarchique pour ouvrir ou fermer le contenu de cet élément. Un événement aussi simple n'a pas besoin d'être diffusé plus largement. Sinon, l'événement arrive au gestionnaire du document interne où il est analysé pour trouver ses abonnements, puis il est rediffusé à tous les abonnements. Lorsque les abonnements reçoivent l'événement, ils mettent à jour leurs structures puis leurs interfaces. Ainsi les différentes parties de l'outil sont toujours synchronisées.

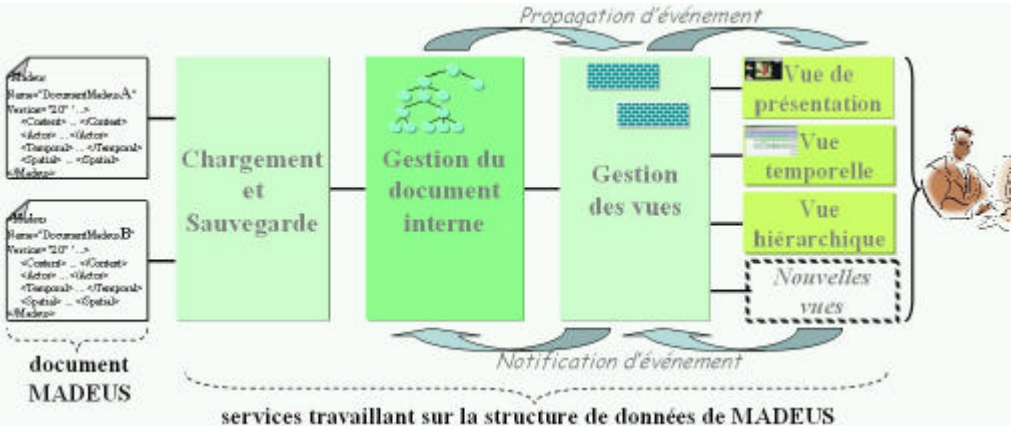

Figure 60. Principe de l'outil Madeus

VI.3.1 Modèle d'objet du document interne

L'outil Madeus utilise une structure de document interne, appelé *MadeusDocument*, qui est basée sur l'interface *DOM*. Cette structure consiste en un *document* et des *éléments*, mais est développée de façon beaucoup élargie avec un ensemble de types d'élément plus riches. Les descriptions détaillées de cette structure interne de Madeus sont données dans [Tardif 00] [Navarros 01] [Villard 02]. Dans cette section, nous présentons une vue synthétique de cette structure à travers le schéma de modèle d'objet ci-dessous (voir la Figure 61). En général, la structure interne est très semblable à la structure du document Madeus externe (voir le chapitre V précédent et la section III.4.8). de ce fait, cette structure interne présente la même limite dans la granularité de la décomposition puisque qu'elle ne permet de manipuler que les médias de façon globale.

Le document interne est un document central du système sur lequel plusieurs services d'édition sont fournis comme le formatage, la vérification de la cohérence, l'édition directe. Il fournit aussi le service de gestion d'événements comme décrit précédemment.

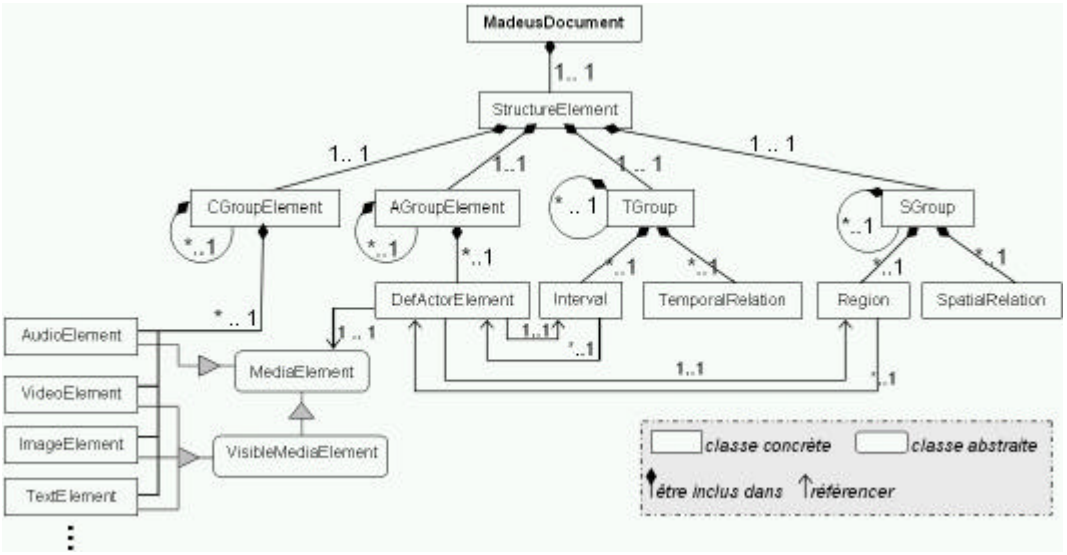

Figure 61. Modèle général d'objet du document interne de Madeus

VI.3.2 Formatage

Le formatage est une machine qui permet de construite et maintenir un document interne valide. Cette machine peut travailler en deux modes : le mode global ou le mode incrémental. En mode global, le formatage est utilisé dans le module de chargement de document externe. Son rôle est de calculer tous les placements des composants à partir des spécifications relatives exprimées dans le document externe. Dans le cas des spécifications invalides dans le document, le formatage peut aussi chercher une solution la plus appropriée. En mode incrémental le formatage permet de maintenir le document interne valide lorsqu'il y a des modifications apportées au cours de l'édition.

Le système Madeus fournit trois types principaux de formatage : formatage d'attributs, formatage temporel et formatage spatial.

Le formatage d'attributs permet de déterminer le type, l'unité ou le format d'une spécification pour trouver une bonne valeur interne d'un attribut. Par exemple, si la spécification de l'attribut *begin* est : *begin="pref:10s"*, le formatage d'attributs produira un attribut interne *beginPref* avec sa valeur 10000 millisecondes.

Les deux types de formatage suivants sont très importants car, ils permettent de construire et valider la structure temporelle interne et la structure spatiale interne du document. Puisque les modèles temporel et spatial de Madeus sont hiérarchiques, Madeus fournit un système de formatage hiérarchique pour partager la charge de formatage à chaque niveau hiérarchique. De ce fait, chaque nœud composite de la structure possède un solveur de formatage. La Figure 62 présente le modèle d'objet d'une telle structure dans le système Madeus où chaque objet temporel composite *TGroup* et chaque objet spatial composite *SGroup* possède un résolveur temporel *TSolver* et un résolveur spatial *SSolver*.

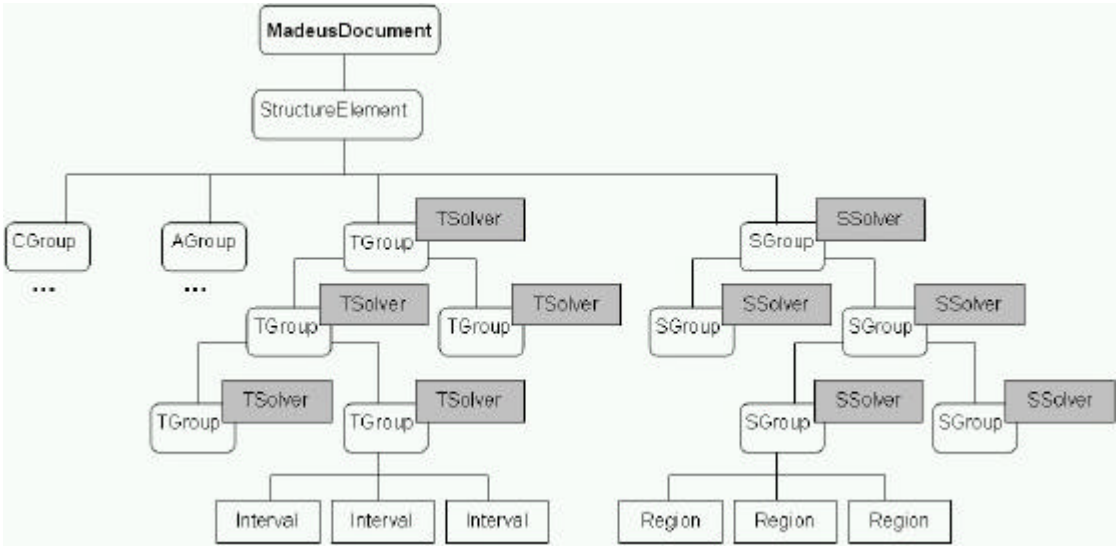

Figure 62. Formatage hiérarchique

Le résolveur temporel utilise le résolveur linéaire *Cassowary* [Badros et al. 98] et le résolveur spatial utilise le résolveur linéaire *DeltaBlue* [Carcone 97]. Une étude approfondie de l'utilisation de différents résolveurs et la comparaison entre eux se trouve dans [Tardif 00].

VI.3.3 Graphe temporel

Bien que la structure hiérarchique permette de représenter richement le contenu d'un document, elle ne représente pas complètement le scénario de synchronisation temporelle du document. Chaque objet temporel composite possède en plus une structure de graphe pour représenter les synchronisations temporelles dans cet objet composite : chaque objet temporel (*Interval* ou *TGroup*) dans l'objet temporel composite est représenté par un arc et deux nœuds (voir la Figure 63). En principe, le graphe utilise les nœuds de début et fin d'un arc et des arcs de délai pour représenter la synchronisation entre des objets. Dans l'exemple ci-dessous, le début de l'objet *A* et les fins des objets *B* et *C* sont représentés par les arcs de délai notés "*d*". La synchronisation de co-démarrage entre l'objet *A* et l'objet *B* (*Starts*) est représentée par leur nœud de début commun. La relation de séquence entre l'objet *A* et l'objet *C* est représentée par le nœud fusionné entre la fin de l'objet *A* et le début de l'objet *C*.

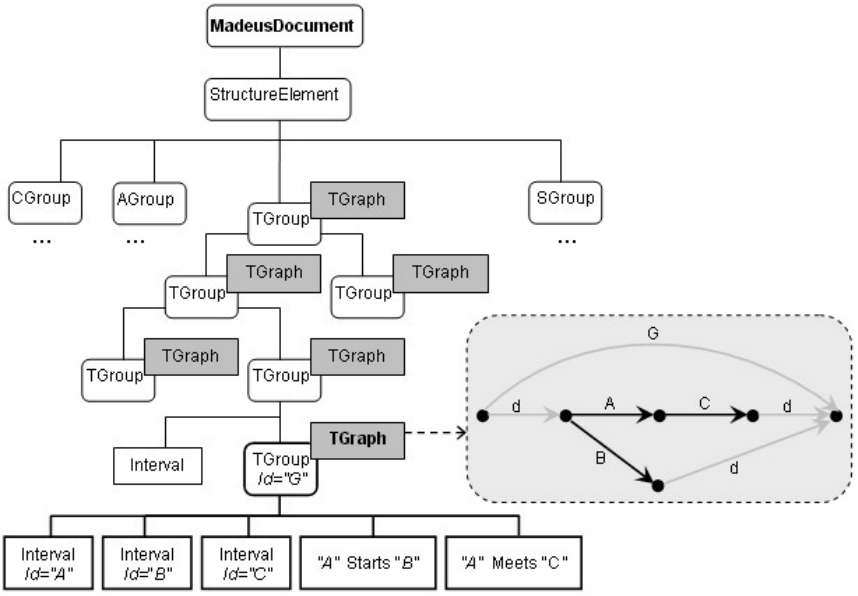

Figure 63. Un exemple d'un graphe et la structure hiérarchique des graphes.

VI.3.4 Principes de construction d'une vue

Chaque vue dans l'outil auteur Madeus possède son propre document dont la structure est dépendante de la caractéristique de la vue spécifique. Par exemple, la vue de présentation possède la structure de document *ExecutionDocument* ; la vue temporelle travaille sur le document *TimeLineDocument* ; la vue hiérarchique utilise son document *HierarchicalDocument* ; etc. Toutefois ces documents de vue ont comme point commun qu'ils sont tous construits en utilisant le document interne du système *MadeusDocument*. Pour créer une nouvelle vue, le document interne central *MadeusDocument* est parcouru selon un algorithme de profondeur d'abord, chaque élément visité est testé par un filtre, et si l'élément est accepté, un élément de document de la vue correspondante est créé. De ce fait on peut construire plusieurs documents de vue dont chaque élément contient un pointeur vers un élément d'origine du document *MadeusDocument*.

En utilisant un filtre dans pour chaque vue spécifique, la construction d'un document de cette vue ne prend en compte que des parties intéressées dans le document central *MadeusDocument*. Par exemple, la vue de présentation ne s'intéresse que des informations de style (la partie *Actor*), de temps (la partie *Tempral*) et d'espace (la partie *Spatial*), il laisse passer la partie *Content* ; la vue temporelle n'utilise que la partie *Temporal* ; la vue hiérarchique utilise tout le document *MadeusDocument*.

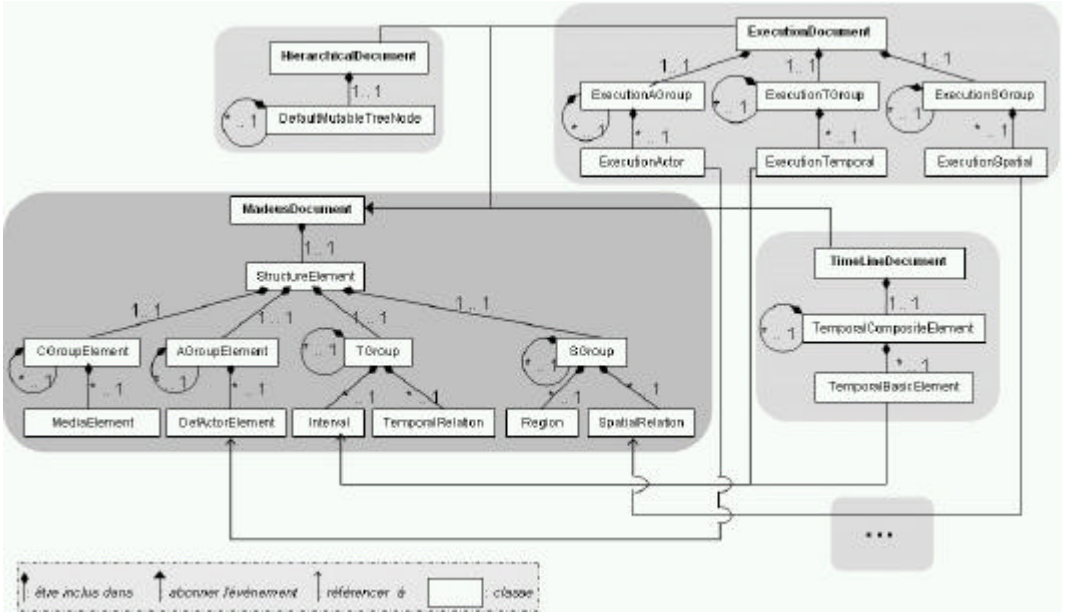

Figure 64. Modèle d'objet des documents de vue.

En fait, il est très facile de développer une nouvelle vue dans l'outil auteur Madeus.

VI.4 Mdéfi : environnement auteur expérimental de composition fine

L'environnement auteur expérimental Mdéfi présenté dans cette section est une extension de l'outil auteur Madeus pour prendre en compte le modèle étendu présenté dans le chapitre V. De plus, en se basant sur le modèle étendu, nous pouvons expérimenter des services d'édition plus fins et plus confortables, qui permettent de composer des documents multimédias plus intéressants.

Toutes les parties de l'outil Madeus sont étendues pour prendre en compte les besoins ci-dessus. La Figure 65 représente les extensions (les parties rouges) dans l'architecture de l'outil Madeus. Précisément, le chargement et la sauvegarde du système doivent être mis à jour pour reconnaître les nouveaux éléments des documents Madeus ouverts. La structure du document interne doit être enrichi pour représenter des nouveaux éléments dans le système interne. Dans la vue de présentation, la machine d'exécution doit être améliorée pour présenter et synchroniser les nouveaux types d'objet média comme l'animation, les médias structurés, les objets ou les segments de média, etc. La vue temporelle est étendue pour permettre de représenter la structure temporelle interne d'un média. Un ensemble de nouvelles vues de média structuré a été ajoutée pour décrire semi-automatiquement le contenu des médias.

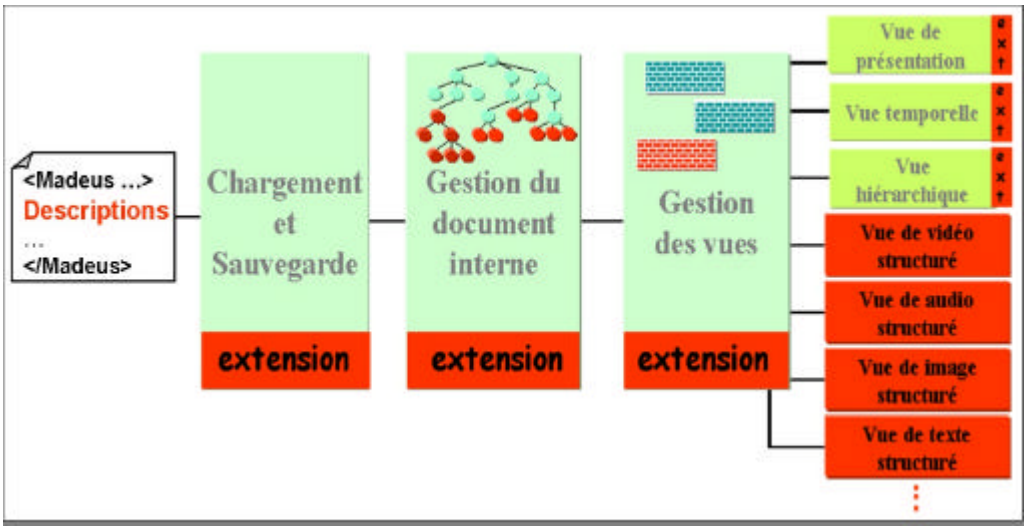

Figure 65. Extensions dans l'architecture de l'outil Madeus

La suite de la section concentre sur les extensions des parties : la structure de document interne, la machine de présentation, la vue temporelle et enfin les vues des médias structurés.

VI.4.1 Structure de document interne central

Nous avons développé une structure d'objet étendue correspondant au modèle des sous éléments présenté dans chapitre précédent.

VI.4.1.1 Modèle d'objet des médias structurés

Le modèle de document interne de Madeus fournit une classe la plus abstraite *MediaElement* pour des objets de média qui contient un minimum des caractéristiques qu'un objet de média doit avoir pour être considéré comme un média dans le système. Les autres classes de média (*AudioMediaElement*, *VisibleMediaElement* et leurs sous classes) doivent hériter de cette classe la plus abstraite. Les objets de média structuré sont aussi construits à partir de ces classes basiques de média. La Figure 66 présente le modèle d'objet étendu pour les objets de médias structurés.

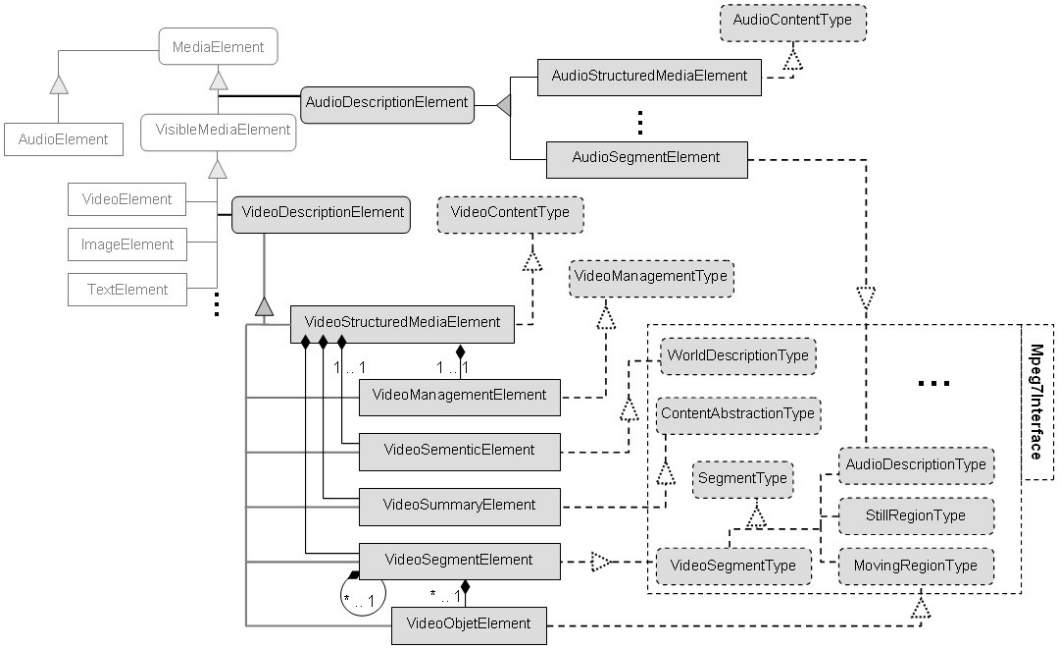

Figure 66. Modèle d'objet des descriptions du contenu des médias

Le modèle d'objet étendu pour les médias structurés hérite d'une part des classes abstraites d'objet média comme la classe *MediaElement* et la classe *VisibleMediaElement* qui permettent de traiter les objets de descriptions du contenu (média structuré) comme un média pour composer la présentation multimédia. D'autre part, les objets du modèle étendu sont implémentés avec des interfaces qui permettent d'accéder aux informations de description. Nous proposons à ce niveau un paquetage des interfaces compatible avec les spécifications de MPEG-7 qui sont construites en se basant sur les syntaxes des outils de description de MPEG-7. Par exemple, pour l'outil de descrption des segments de vidéo *mpeg7:VideoSegmentType* (cf. la section V.2.4.2.2), nous proposons l'interface *VideoSegmentType* décrite ci-dessous :

```
package FR.inria.opera.Kaomi.Document.Content.Mpeg7Interface.ContentDescription;

import java.util.*;
import FR.inria.opera.Kaomi.Document.Content.Mpeg7Interface.LinkToMediaLocation.TimeDSTools.*;
import FR.inria.opera.Kaomi.Document.Content.Mpeg7Interface.SchemaTools.BaseTypes.*;
import FR.inria.opera.Kaomi.Document.Content.Mpeg7Interface.LinkToMediaLocation.MediaLocators.*;

public interface VideoSegmentType extends SegmentType {
        public MediaTimeType getMediaTime();
        public SegmentType[] getDecompositionElements();
        public VideoSegmentType[] getTemporalDecompositionElements();
        public SegmentType[] getSpatialDecompositionElements() ;
        ...
}
```

Une telle interface permet de récupérer des informations temporelles *getMediaTime()* et les éléments de décomposition de segment vidéo. Comme ce travail ne s'intéresse dans un premier temps qu'à l'expérimentation de la composition directe avec la structure des médias, nous ne nous intéressons ici qu'à l'accès aux informations temporelles, spatiales et hiérarchiques du contenu.

VI.4.1.2 L'objet représentant l'animation

Les animations abstraites (voir la section V.4) sont des objets médias particuliers, elles ne sont ni des objets de média audio, ni des objets de média visible ou continu. Nous avons donc proposé de construire des classes qui héritent directement de la classe la plus abstraite d'objet média *MediaElement* pour représenter les animations abstraites dans notre système. La Figure 67 présente un modèle d'objet des animations correspondant aux éléments d'animations abstraites présentées dans les sections III.4.7.2 et V.4.

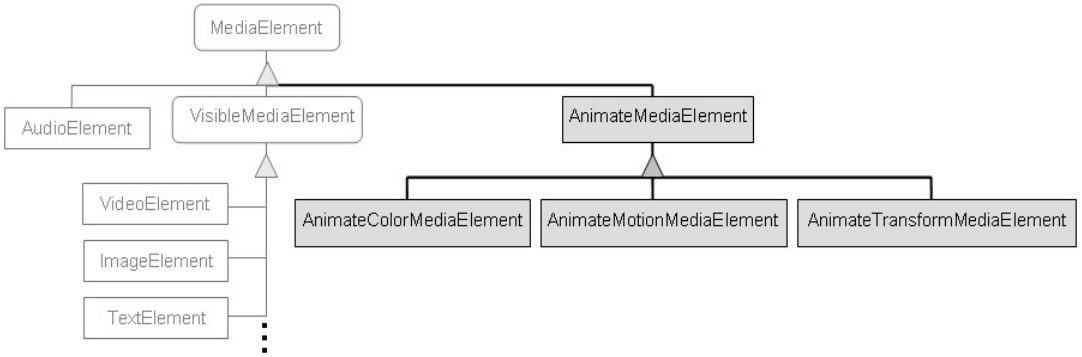

Figure 67. Le modèle d'objet des animations abstraites dans le système Mdéfi.

Ces objets d'animation abstraite seront utilisés par des *players* particulier pour animer des objets médias.

VI.4.1.3 L'objet représentant le sous-acteur

La classe *DefSubActorElment* est une extension de la classe *DefActorElement* pour représenter l'élément de sous acteur à l'intérieur du système. Comme sa super classe *DefActorElement* (cf. la Figure 61), la sous-classe *DefSubActorElment* crée des objets centraux qui intègrent toutes les informations pour un sous acteur. Elle hérite de sa super classe un pointeur vers un contenu, et possède elle-même des informations de style de présentation d'un sous acteur (cf. la Figure 68).

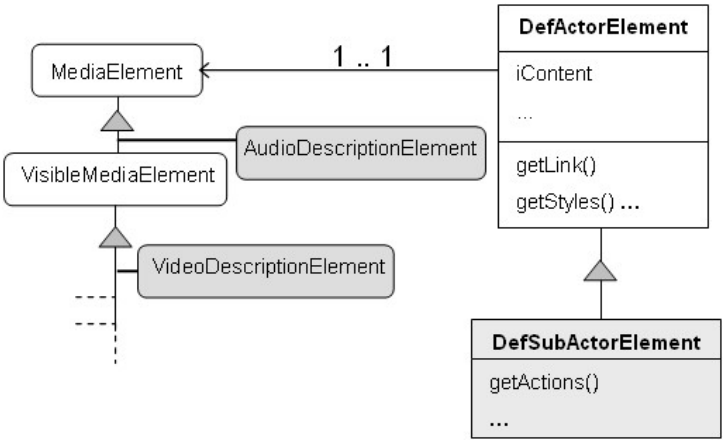

Figure 68. Modèle d'objet de sous-acteur

En fait, pour cette expérimentation, l'objet *defSubActorElement* n'est pas beaucoup étendu par rapport à sa super classe, sauf quelques méthodes comme appliquer des actions sur le sous-acteur. Il fonctionne donc principalement en utilisant des méthodes et des attributs de sa super classe. Toutefois pour une

expérimentation de plus haut niveau il peut être plus étendu, par exemple, pour représenter des sous acteurs génériques au lieu de sous-acteurs spécifiques déterminés par des identificateurs (IDs).

VI.4.1.4 L'objet représentant le sous-intervalle

La Figure 69 présente le modèle d'objet étendu de la sous-classe *SubIntervalElement* de la classe *Interval* pour représenter le sous-intervalle dans le document interne.

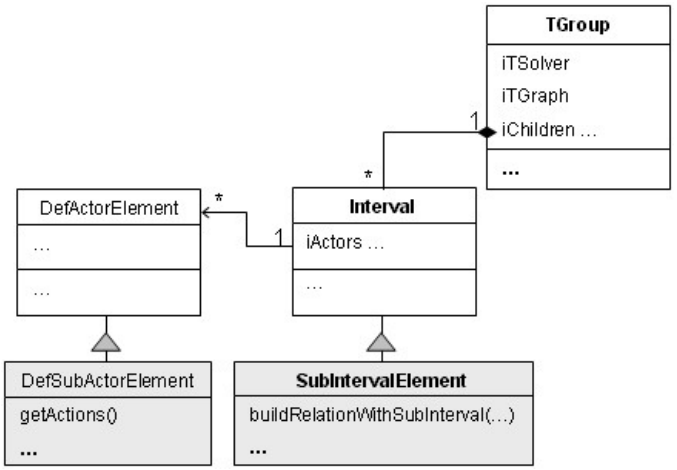

Figure 69. Modèle d'objet de l'élément de sous intervalle.

Lorsqu'un sous-intervalle est créé, il est automatiquement synchronisé avec l'intervalle qui le contient par une relation *à l'intérieur*. Deux délais par défaut sont créés avant et après le sous intervalle. De ce fait, il est assuré que le sous-intervalle est toujours présenté pendant l'intervalle qui le contient. Une telle relation intrinsèque ne peut pas être représentée sous forme d'arbre hiérarchique, mais sous forme de graphe (voir la Figure 70).

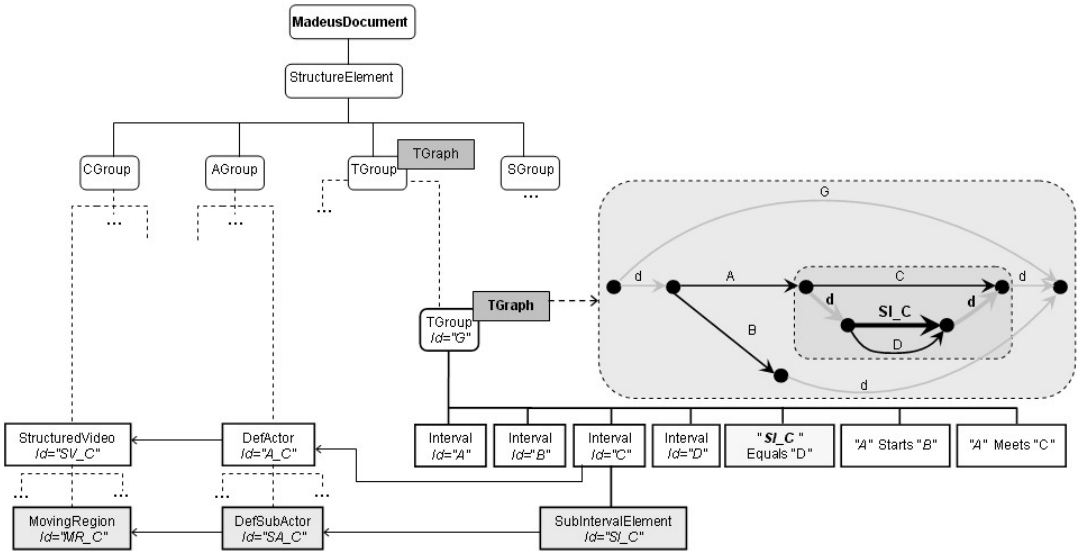

Figure 70. Exemple d'un objet de sous intervalle sous forme hiérarchique et graphique.

La disposition temporelle du sous-intervalle par rapport à l'intervalle qui le contient dépend du type de sous-intervalle (*actif* ou *passif*). S'il est actif, le

formatage prendra des informations temporelles du segment qu'il représente pour déterminer la position temporelle du sous intervalle. Dans ce cas la disposition temporelle par rapport à l'intervalle qui le contient est fixée. Ce type de sous intervalle est donc utilisé pour synchroniser les autres objets temporels. L'exemple dans la Figure 70 présente la synchronisation *Equals* entre le sous intervalle actif *SI_C* et l'intervalle *D*. Le résultat est la présentation parallèle d'un objet (ici *D*) avec une occurrence dans la vidéo (sous-intervalle *SI_C*). Si le sous intervalle est passif (dans le cas il représente un segment statique comme segment de texte ou représente une animation), la disposition est plus souple. Le formateur peut se baser sur les spécifications de sous-intervalle. Dans ce cas aucune spécification temporelle n'est donnée pour le sous intervalle, il sera formaté plus tard par les spécifications relatives à travers des relations temporelles entre le sous intervalle et un autre objet temporel ou même par des manipulations directes de l'utilisateur à travers la vue temporelle (voir la section V.3.3 pour avoir plus détail).

Le principe de l'approche est toujours basé sur des délais pour faire la synchronisation fine. Toutefois, l'innovation de l'approche est la sémantisation du sous intervalle par la référence à un acteur ou directement à une description d'un segment média. De ce fait, le sous-intervalle peut représenter un segment de média dont la signification est bien décrite selon un modèle de description du contenu standard. L'autre avantage très importante est la capacité de formatage automatique du sous intervalle. Cela surmonte la limite de spécification absolue de l'approche classique.

VI.4.1.5 L'objet représentant la sous-région

La Figure 71 présente le modèle d'objet de sous région et un exemple de document interne de sous région. Sur le même principe que celui sous-intervalle, l'objet de sous-région est une extension de l'objet de région. Lorsque un l'objet de sous région est créé, s'il représente un segment décrit dans la partie *Content* (voir la Figure 71), il est formaté automatiquement par la méthode *setIntrinsicValue(…)* qui va prendre des informations spatiales de segment décrit.

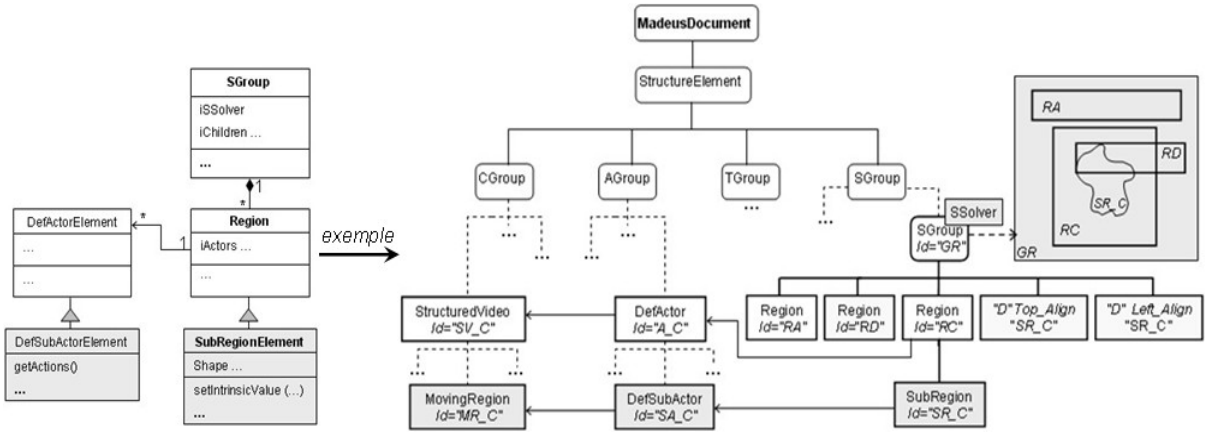

Figure 71. Le modèle d'objet de sous région et un exemple de structure des objets internes de sous région.

L'exemple dans la Figure 71 montre aussi les relations spatiales fines (*top_align* et *left_align*) entre la sous-région *SR_C* et la région *RD*. Ces relations sont enregistrées dans le résolveur spatial (*SSolver*) de la région composite

(*SGroup*) pour maintenir la disposition spatiale entre ces deux régions lorsque l'auteur fait des modifications comme le déplacement d'une des deux régions. De plus, dans le cas où une des deux régions est une région mobile, la sous-région est alors mobile (par exemple une occurrence mobile de la vidéo), les relations sont aussi maintenues par le résolveur. Dans notre exemple, la région *RD* sera alignée en haut et à gauche de la sous-région *SR*_C à tout instant de la présentation, même lorsque l'occurrence associée à la sous-région *SR*_C sera en mouvement.

Comme le sous intervalle, un tel mécanisme de représentation des sous régions est plus significatif que les mécanismes d'*ancre* ou de *secteur* (*anchor* et *area*). Une sous région représentée est décrite selon un modèle de description du contenu de média. De ce fait l'objet représentant une sous région a deux parties, une partie logique qui contient des descriptions du contenu plus sémantique (par exemple l'objet *MovingRegion*) et une partie de présentation (l'objet *SubRegion*). Alors la partie de présentation s'occupe de la présentation de la sous-région, et la partie logique s'occupe de la logique de la présentation de la sous-région. Par exemple, pour les régions mobiles l'objet de sous région *SubRegion* ne calcule pas à chaque instant les positions de ces régions. C'est la partie logique, constituée des objets de la description de contenu des régions mobiles, qui mettent à jour les valeurs de l'objet de présentation *SubRegion*. L'objet de sous région *SubRegion* ne fait seulement qu'afficher les valeurs les plus à jour de cette sous-région.

VI.4.1.6 Les objets de structure temporel/spatial d'un média structuré

Les objets présentés ci-dessus correspondent directement au modèle externe de document, i.e., ces objets de document interne ne permettent de représenter que des segments de média qui ont utilisés pour spécifier la présentation du document. Ils sont suffisants pour jouer le document. Mais pour éditer le document une telle structure d'objet a encore des limites. Bien que l'auteur puisse identifier des segments d'un média à travers les descriptions du contenu du média dans la partie *Content*, un tel mode de spécification est encore trop complexe pour l'auteur. Une représentation graphique des structures temporelles et spatiales du contenu du média permet à l'auteur d'identifier plus facilement des segments du média pour composer le document.

Dans notre environnement auteur, nous proposons de visualiser les structures temporelles et spatiales du contenu d'un média structuré dans les vues spatiales et temporelles du document. Nous décrirons ces vues de façon plus détaillées dans les sections VI.4.2 et VI.4.3 suivantes. Pour permettre aux vues d'afficher les structures internes d'un média structuré, nous proposons de créer deux types d'objet dans le modèle d'objet de document interne. Ce sont les objets *TemporelStructureComponent* et *SpatialStructureComponent* qui représentent des composants temporels et spatiaux du contenu d'un média structuré. La Figure 72 présente les modèles de ces objets dans le modèle de document interne.

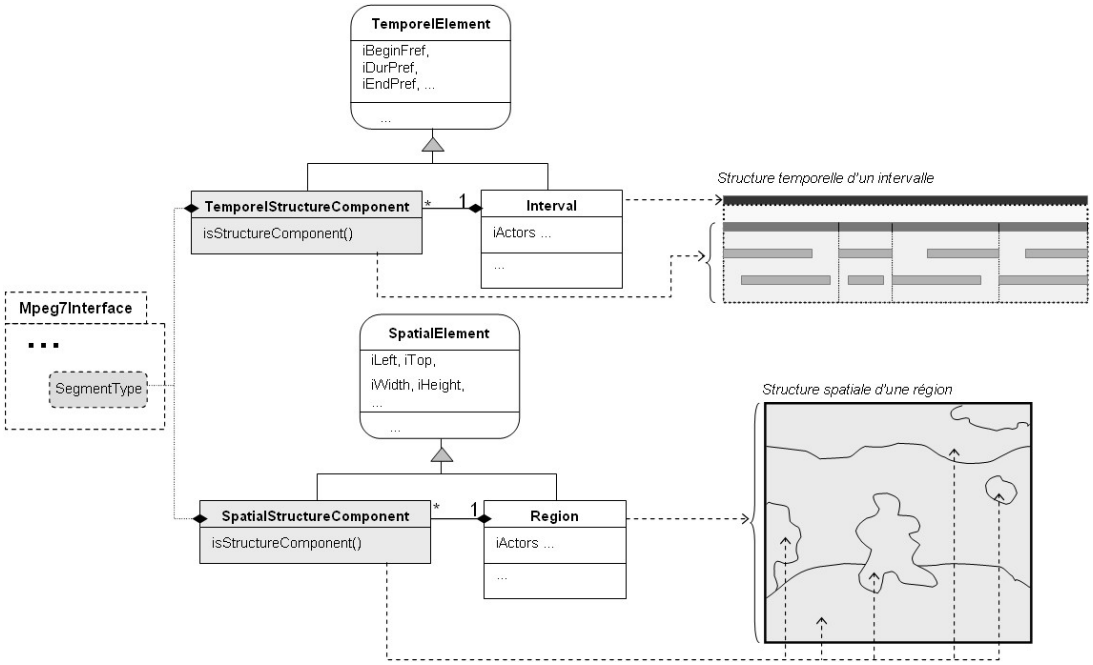

Figure 72. Le modèle d'objet et l'exemple des structures spatiales et temporelles de média structuré.

En principe, les objets composant la structure interne d'un média structuré sont des objets temporels ou spatiaux du document interne, mais ils ne sont pas utilisés pour synchroniser les présentations entre des média ou segment média. Ils sont utilisés dans la vue temporelle pour représenter la structure temporelle du contenu d'un média structuré, et dans la vue spatiale pour représenter la structure spatiale du contenu d'un média visuel structuré. De ce fait, le contenu des médias n'est plus une boîte noire, l'auteur dans la vue temporelle ou spatiale peut voir et identifier facilement des segments temporels ou spatiaux pour composer, par exemple, des synchronisations fines avec d'autres médias, ou des liens sur des sous régions.

La construction des objets composant la structure interne d'un média structuré est totalement automatique et est basée sur les descriptions du contenu de média structuré de la partie *Content* du document interne/externe.

VI.4.2 Présentation du document dans *la vue d'exécution*

La vue d'exécution présente le document multimédia à l'auditoire sur les périphériques graphique et audio de façon interactive. La machine d'exécution et les gestionnaires de présentation associés sont décrits dans [Sabry 99]. Dans cette section, nous allons présenter principalement le mécanisme l'exécution de cette vue utilisé pour faciliter l'accès aux médias structurés.

VI.4.2.1 Principes d'exécution dans Madeus

Le modèle d'exécution du système Madeus est très souple, il est structuré en deux parties : gestionnaire d'exécution de média (GEM) et gestionnaire d'exécution du document (GED) (voir la Figure 73). Cette organisation du système d'exécution correspond bien aux deux niveaux de présentation multimédia : média individuel et intégration multimédia. Le GEM gère la présentation de chaque média individuel,

tandis que le GED gère les synchronisations entre des médias individuels et les paramètres de présentation pour des médias individuels.

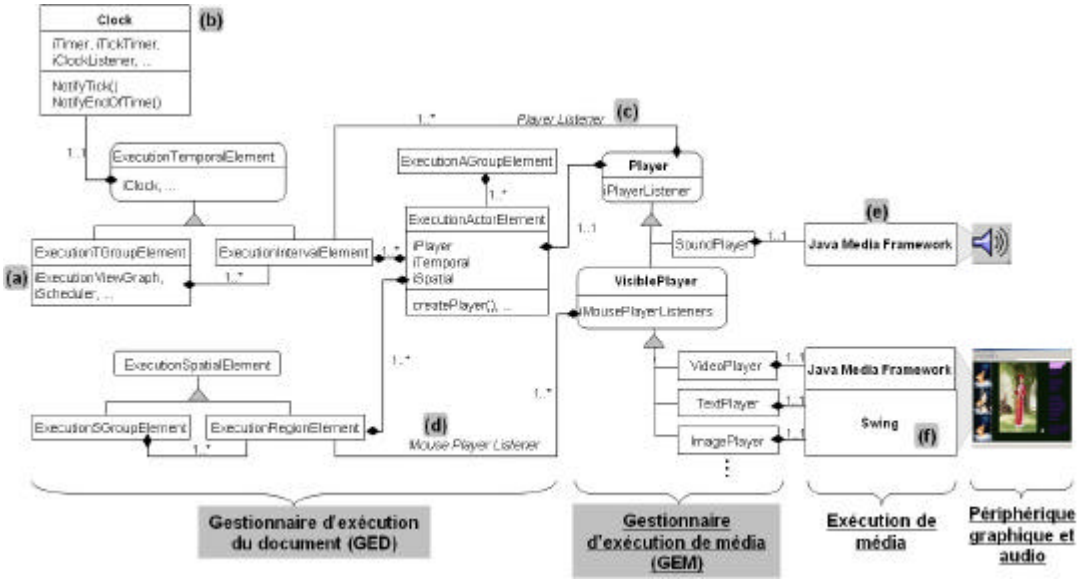

Figure 73. Modèle d'objet du système d'exécution.

Le GEM permet d'embarquer des *players* média existants comme *swing* pour afficher des médias graphiques et *JMF* (*Java Media Framework*) pour présenter la vidéo et l'audio (voir la Figure 73e et la Figure 73f). En fait chaque exécution de média est gérée par un GEM spécifique qui permet de lancer la présentation de média selon des ordres et des paramètres de présentation du GED. Le GEM gère aussi les événements issus de l'exécution de média comme *pause*, *fin*, *clic sur*, etc.. Le GEM les envoie au GED à travers des écouteurs du *player* ("*player listener*" et "*mouse player listener*", voir la Figure 73c et la Figure 73d). Nous proposons d'étendre ces GEMs pour permettre de gérer et accéder plus finement dans les exécutions de média structuré (voir la section VI.4.2.2).

Le GED est le scénario temporel et spatial du document, il est construit directement à partir des structures temporelles et spatiales du document interne central du système (voir la section VI.3.4). Le GED permet de gérer la présentation du document selon trois dimensions : temporelle, spatiale, événementielle :

- Le gestionnaire d'exécution temporelle se compose des objets d'exécution temporelle (*ExecutionTemporalElement* et ses sous classes) qui sont construits à partir de la partie temporelle du document interne central (voir la section VI.3.1). Chacun de ces objets temporels contient une horloge (voir la Figure 73b) pour gérer la durée d'une présentation temporelle. Un graphe d'exécution qui est construit directement à partir du graphe temporel du document interne (voir la section VI.3.3) gère le démarrage de chaque objet d'exécution temporelle. Le graphe d'exécution est contenu dans chaque objet d'exécution temporelle composite (*ExecutionTGroupElement*, voir la Figure 73a). En fait, la présentation temporelle du document est effectuée à travers le graphe d'exécution par un ordonnanceur : au début, les objets d'exécution temporelle liés aux arcs de sortie du premier nœud sont exécutés ; lorsque tous les objets des arcs entrants d'un nœud sont finis, des

objets des arcs sortants de ce nœud sont lancés ; le processus continue jusqu'au dernier nœud du graphe. Le parcours dans le graphe d'exécution est également hiérarchique. Lorsque un objet composite est exécuté dans un graphe, il active son graphe d'exécution. Lorsque un objet d'exécution temporelle est activé, les *players* des médias contenus dans cet objet sont lancés pour présenter des médias dans les périphériques.

- Le gestionnaire d'exécution spatiale se compose des objets d'exécution spatiale (*ExecutionSpatialElement* et ses sous classes) qui sont construits à partir de la partie spatiale du document interne central de Madeus (voir la section VI.3.1). Ces objets gèrent et fournissent des paramètres d'affichage comme la position, la dimension et la disposition des régions spatiales pour des *players* de média visuel. A noter que les gestionnaires d'exécution temporelle et spatiale communiquent avec les *players* à travers les exécutions d'acteurs (*ExecutionActorElement*). L'objet d'exécution d'acteur sert d'intermédiaire entre deux parties de gestion d'exécution et de *player* média.

- Le gestionnaire d'événement de synchronisation reconnaît deux sortes d'événements : interne et externe :

  a. Le gestionnaire d'événement synchronisation interne permet d'effectuer des synchronisations dans le système. Par exemple, lorsque la durée d'un objet d'exécution temporelle s'achève, un événement de fin d'objet est envoyé à l'ordonnanceur pour exécuter les objets suivants. Dans le système Madeus ce type d'événement est installé systématiquement pour chaque objet d'exécution temporelle et géré par l'horloge de chaque objet. Un autre type d'événement de synchronisation issu de *player* permet de synchroniser le *player* avec le gestionnaire d'exécution. Cela permet de gérer des cas asynchrones et lorsque la durée des objets est indéterminée. Par exemple, les *players* des médias continus ont leur propre temps qui n'est souvent pas synchronisé avec la présentation globale du document. Les synchronisations avec ces médias peuvent alors être cassées, et un événement de *player* informant le gestionnaire d'exécution temporelle de ce ralenti peut aider le gestionnaire temporel de gérer ce ralenti. C'est aussi le principe du gestionnaire de la synchronisation des lèvres (*lipsync*) dans [Sabry 99] (Madeus 1.0) dans lequel le gestionnaire de synchronisation désigne un objet de média comme un maître : c'est lui qui s'occupe de signaler régulièrement des événements aux autres objets considérés comme des esclaves. Ce type d'événement est aussi important pour notre implémentation des synchronisations fines qui sera détaillée dans la section VI.4.2.3. La différence entre notre synchronisation fine et la synchronisation des lèvres est l'unité de synchronisation : notre synchronisation fine est basé sur les événements des segments sémantiques, tandis que la synchronisation des lèvres est basée sur les événements réguliers dans des unités constantes du temps. Ce type d'événement permet aussi de gérer des médias indéterminés comme des flux réel de la vidéo ou de l'audio. Lorsque le flux réel est fini, un événement est envoyé au gestionnaire d'exécution temporelle pour synchroniser avec les autres média (voir la section VI.4.2.3).

b. Le gestionnaire d'événement de synchronisation externe permet de traiter des interactions de l'utilisateur avec la présentation du document. Grâce à ce gestionnaire d'événement, le système peut effectuer des navigations internes ou externes de la présentation selon la demande de l'utilisateur. Nous avons raffiné ce gestionnaire d'événement pour permettre à l'utilisateur d'interagir plus finement avec des segments d'un média (voir la section VI.4.2.4).

En bref, l'organisation de la vue présentation se compose de deux parties principales : gestionnaire d'exécution du document et gestionnaire d'exécution de média. Le gestionnaire d'exécution du document est construit en se basant sur le document interne central du système. Tandis que le gestionnaire d'exécution de média est construit en se basant sur le type de média et travaille sous la supervision et le contrôle du gestionnaire d'exécution.

VI.4.2.2 Exécution de média structuré et exécution complémentaire

Une exécution de média atomique (non structuré) peut être gérée à gros grain avec les actions de démarrage, de pause et de fin ainsi que en considérant la région du rectangle d'affichage (pour les médias visuels). Le fait de décrire le contenu de média permet de mettre en œuvre des exécutions plus sophistiquées pour ce média, appelles l'**exécution de média structuré** et l'**exécution complémentaire**, qui peuvent non seulement présenter le flux des données du contenu de média, mais aussi traiter parallèlement les métadonnées de description du contenu. L'exploitation de ces métadonnées au moment de l'exécution du média permet de gérer et traiter plus finement la présentation d'un média, par exemple, déterminer une occurrence d'un objet vidéo, sa position et sa forme et mettre un contour de suivi ou un lien sur cette occurrence.

Nous proposons un modèle d'exécution de média structuré et d'exécution de segment en étendant le modèle d'exécution média de Madeus comme présenté dans la Figure 74.

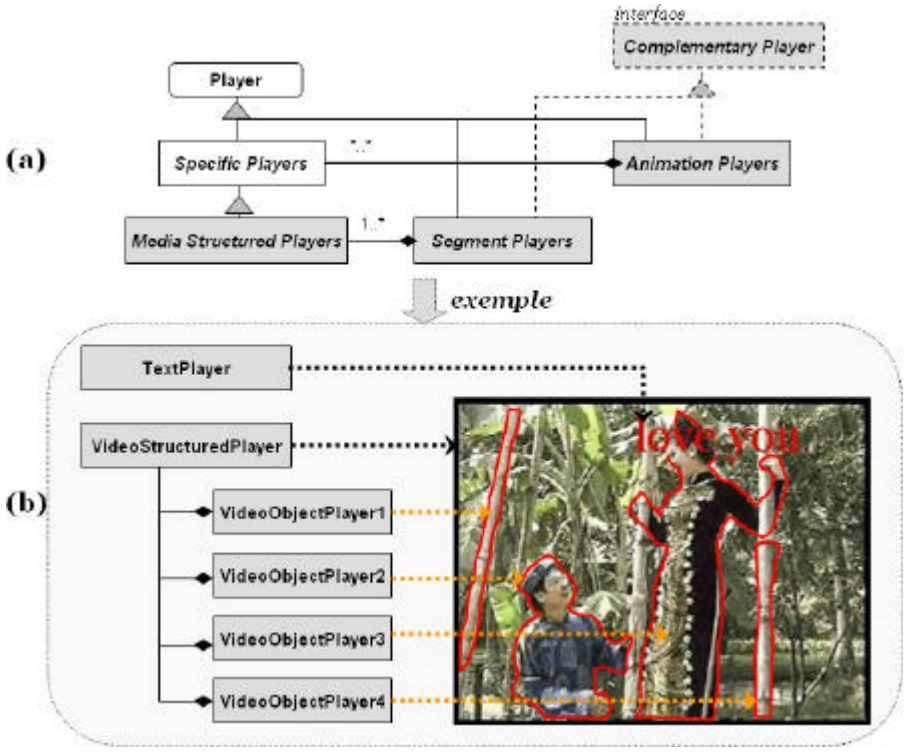

Figure 74. Modèle d'objet de l'exécution de média structuré et de l'exécution de segment (a), et un exemple d'une exécution de vidéo structurée et des exécutions de segments d'objet (b).

L'exécution de média structuré hérite de la classe exécution du type de média spécifique correspondant, par exemple l'exécution de vidéo structurée *StructuredVideoPlayer* hérite la classe *VideoPlayer* ; etc. Les extensions apportées permettent 1) d'accéder aux objets d'éléments médias structurés (voir la section VI.4.1 précédente) qui contient les informations de contenu de média ; et 2) de créer des *processus d'exécution* de média au lieu des simples présentations de média (le *processus d'exécution* de média permet d'interagir plus finement dans le processus de présentation d'un média tandis que la simple présentation de média ne permet pas d'interagir avec la présentation de média).

Ainsi, l'exécution de média structuré permet :

1) de présenter une partie quelconque d'un média par exemple, il peut extraire une scène vidéo, un morceau de musique, une personne d'une image ou même un personnage d'une vidéo. Ce mode d'extraction du contenu et son exécution est effectué sans couper physiquement le média et donc sans avoir besoin de gérer des médias distincts pour chaque segment (voir la section VI.4.4.3) ;

2) de gérer les synchronisations fines avec des segments de médias structurés.

En fait, les exécutions des médias continus structurés comme *VideoStructuredPlayer* et *AudioStructuredPlayer* peuvent posséder une *timeline* ordonnant des présentations des segments de la structure du contenu de média pour synchroniser ces segments de média avec des autres médias ou segments médias (voir la section VI.4.2.3). Cependant, comme la structure du contenu d'un média structuré peut être de grande taille et complexe, la *timeline* ne peut pas représenter tous les segments de la structure du contenu, car sinon elle serait trop lourde et

même redondante. Aussi la *timeline* est construite pour ne représenter que les segments de média qui sont utilisés dans des synchronisations fines de la présentation.

Les exécutions complémentaires n'interprètent pas un contenu mais leur rôle est d'effectuer des modifications sur la présentation de média. C'est pourquoi elles toujours dépendent d'une exécution de média (structuré). L'exécution complémentaire hérite directement de la classe d'exécution la plus abstraite du système (*Player*). De ce fait les objets de l'exécution complémentaire sont plus légers pour effectuer des modifications sur la présentation de média. Deux types d'exécutions complémentaire sont identifiés : l'exécution d'animation et l'exécution de segment.

- L'exécution d'animation implémente des fonctions d'animation qui permettent d'appliquer des spécifications d'animation de l'auteur sur la présentation de média. Les spécifications d'animation sont obtenues à travers l'objet d'animation abstraite (voir la section VI.4.1.2) que possède l'exécution d'animation.
- L'exécution de segment représente des actions sur les segments de média. Plus précisément elle permet d'agir sur des segments d'une présentation de média structuré. Elle possède ou peut accéder à un objet de description d'un segment (un objet sous-acteur et les objets de description du contenu de média, voir la section VI.4.1) et utilise les informations de ces descriptions pour localiser à la fois temporellement et spatialement le segment dans la présentation de média structuré. Son rôle est d'appliquer des actions spécifiées par l'auteur sur cette portion de média. L'exécution de segment possède donc également des fonctions pour modifier la présentation du segment dans la présentation de média, par exemple afficher le contour d'un objet vidéo, sélectionner une phrase de texte, augmenter le volume d'un segment audio, etc. Pour cela, l'exécution de segment doit être une exécution **complémentaire**, i.e., elle doit exister toujours comme une sous exécution d'une exécution de média structuré pour compléter la présentation de ce média structuré par ses modifications sur cette présentation.

A noter que l'exécution d'animation permet d'agir sur la présentation entière du média, tandis que l'exécution de segment permet d'agir sur une portion de la présentation de média. L'exécution de média (structuré) et ses exécutions complémentaires forment un ensemble de composants d'exécution qui permettent de présenter le média de façon plus sophistiquée. L'exemple dans la Figure 74 présente un tel ensemble d'exécution pour une vidéo structurée et quatre objets de cette vidéo.

Discussion des choix d'architecture. Nous avons choisi l'approche de séparation de l'exécution de média (structuré) et des exécutions complémentaires (animation et segment média) au lieu d'intégrer toutes ces exécutions dans une seule exécution complexe qui aurait gérée à la fois la présentation du contenu de média et les modifications sur cette présentation. Une telle exécution complexe peut être quelque fois plus performante à cause des traitements et des synchronisations intra-objet. Cependant, l'implémentation de cette exécution complexe n'est pas conforme à la structure du système sous-jacent. L'exécution

complexe devrait être implémentée par un gestionnaire d'exécution pour l'ensemble des modifications pendant la présentation de média. Ce gestionnaire d'exécution encapsulerait les événements de modifications de média qui ne pourraient pas alors être pris en compte pour la réalisation des synchronisations fines inter médias, par exemple, sélectionner (*highlight*) une phrase d'une chanson lorsque le segment de musique lui correspondant est présenté. C'est en effet le gestionnaire d'exécution du document (GED) qui en a la charge. Tandis que, avec l'approche que nous avons choisie toutes les exécutions (de média, d'animation ou de segment média) sont gérées par le gestionnaire d'exécution du document (GED). De ce fait les synchronisations fines entre des segments de média peuvent être effectuées facilement. Cette architecture présente aussi les avantages apportés par la modularité qui permet de ne pas changer la classe d'exécution de média à chaque fois qu'on ajoute une fonction de modification de média.

VI.4.2.3 L'exécution temporelle d'un segment média

La présentation temporelle d'un segment média est modélisée comme un sous-intervalle dans le modèle (voir la section V.3.3). Comme présenté dans les sections III.4.4 et VI.4.1.4, le modèle de l'intervalle consiste à placer deux délais au début et à la fin du sous-intervalle pour disposer la présentation temporelle d'un segment média par rapport à la présentation temporelle du média qui le contient.

Média non continu. Ce mécanisme de positionnement temporel fin est bien fonctionnel pour des segments non continus comme une phrase de texte et une région d'image dont le temps de présentation dépend uniquement du gestionnaire d'exécution. En effet, les exécutions de segments et de médias non continus sont toujours ordonnées par le gestionnaire d'exécution du document (voir la Figure 75). Ils fonctionnent toujours comme des esclaves du GED. Des synchronisations entre eux sont donc toujours respectées. La Figure 75 présente un exemple d'une présentation temporelle d'un segment du texte où le segment texte "*just tack my hand*" est sélectionné et mis en valeur après trois secondes de la présentation du texte entier, la sélection est dure pendent quatre secondes puis le fragment textuel est désélectionné.

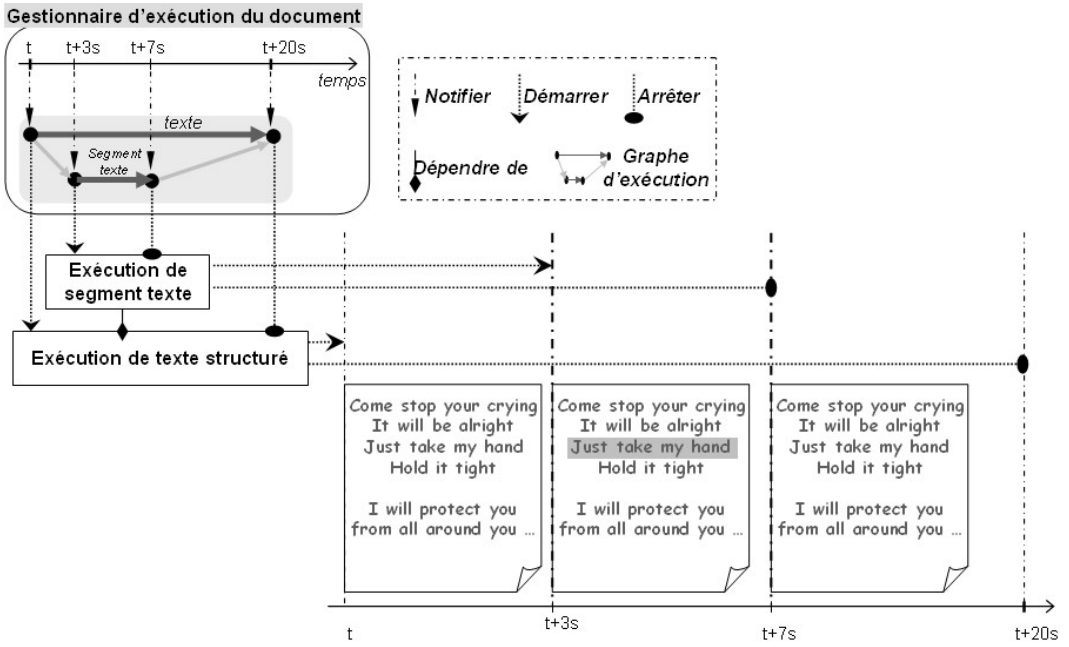

Figure 75. Présentation d'un segment texte dans un texte média.

Média continu. La présentation temporelle d'un segment de média continu dépend fortement du temps de présentation interne du média. Ce temps peut être désynchronisé avec le temps du gestionnaire d'exécution du document. Dans ce cas, l'utilisation du gestionnaire d'exécution pour exécuter le segment continu risque de produire une présentation incorrecte du segment, par exemple avec une désynchronisation perçue par le lecteur. On peut gérer ce problème par la synchronisation des lèvres où le gestionnaire d'exécution va vérifier et ajuster régulièrement le temps interne de média continu. Bien que cette solution puisse assurer une bonne présentation du scénario global du document, la présentation de média continu risque de ne plus être naturelle. Nous avons proposé donc d'utiliser le temps interne de l'exécution du média continu structuré pour ordonner des exécutions des segments. Pour cela, lorsqu'une exécution de média continu structuré démarre, les nœuds de synchronisation intermédiaires de cette exécution (voir le graphe d'exécution de la Figure 76) n'attendent que des notifications en provenance de l'exécution de média structuré. Toutes les notifications de l'horloge du gestionnaire d'exécution du document pendant le temps d'exécution de média structuré sont ignorées (voir la Figure 76). La Figure 76 présente une telle exécution d'un segment vidéo. L'exécution d'un segment vidéo fait apparaître le contour rouge d'un objet vidéo et le synchronise temporellement avec la présentation d'un texte "*love you*". Un *timeline* d'exécution des segments de la vidéo est construit dans l'exécution de la vidéo structurée (voir la section VI.4.2.2 précédente). Ce *timeline* est basé sur le temps interne de média, c'est pourquoi il peut assurer une exécution correcte du segment média. Pendant l'exécution de la vidéo structurée, les deux nœuds synchronisation du segment vidéo dans le graphe d'exécution ne traitent plus des impulsions de l'horloge du GED, ils attendent des notifications de l'exécution de la vidéo structurée pour démarrer et/ou arrêter des exécutions de segment vidéo et du texte.

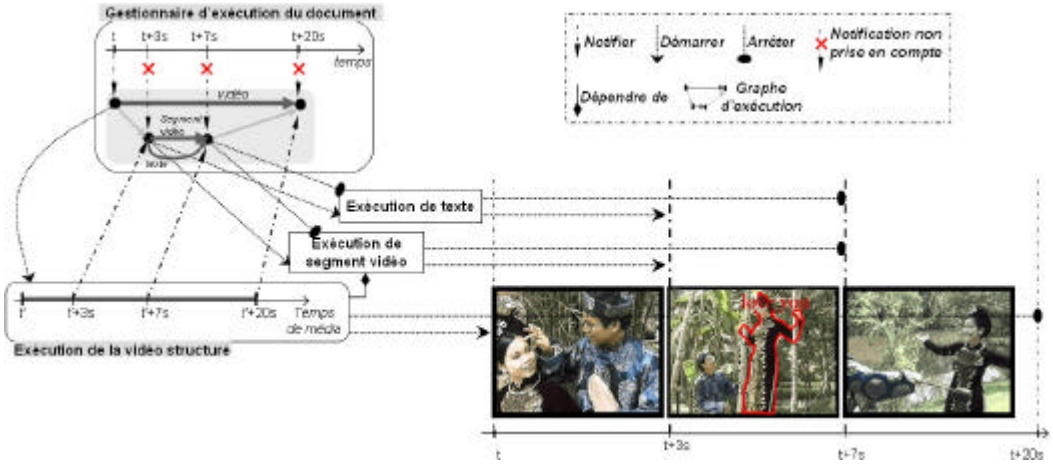

Figure 76. La présentation temporelle d'un segment vidéo.

Ce mécanisme de synchronisation basé sur le temps interne des médias continus peut être comparé avec le modèle de présentation multimédia centré sur un média continu (*video/audio-centred*) [Celentano et al. 99] qui permet de synchroniser finement des médias statiques comme le texte et l'image avec des fragments de média continu. Cependant, le modèle de *Celentano et al.* limite le scénario de document à la structure présentation du média continu et n'est donc appliqué que pour des applications spécifiques. De plus, leur modèle de structuration du contenu des médias n'est pas assez riche. Puisque la structuration du contenu de la vidéo est définie seulement trois éléments : *histoire*, *clip* et *scène* (*story*, *clip*, *scene*). Le travail présenté ici propose d'intégrer la synchronisation basée sur le temps de média continu dans la synchronisation globale du document. De ce fait, la synchronisation dans le document est non seulement enrichie, mais aussi la synchronisation fine avec des segments de média continu est assurée. De plus, nous avons basé sur le modèle de description du contenu des médias riches et utilisé les outils standard MPEG-7 pour décrire le contenu d'un média (voir la section V.2).

Cependant la limite de notre approche vient de ce mécanisme de contrôle du temps pour les médias pour éviter tout conflit dans les horloges, il est nécessaire d'établir une priorités dans les horloges des médias continus.

VI.4.2.4 L'exécution d'un segment spatial

La représentation d'un segment spatial est effectuée par l'exécution de segment (voir la section VI.4.2.2). L'exécution de segment peut accéder à des informations d'un segment spatial, cependant ce n'est pas une exécution autonome de média visuel. Le couple d'exécution de média structuré et d'exécution de segment permet à l'exécution de segment de présenter le segment spatial sur le graphique de l'exécution de média structuré (voir la Figure 76).

Le fait d'utiliser le graphique de média structuré permet à l'exécution de segment d'effectuer des modifications sur la présentation du média. Cependant l'exécution de segment ne peut pas gérer les actions interactives de l'utilisateur sur le segment spatial. C'est pourquoi la gestion d'événements spatiaux de média structuré doit être raffinée pour pouvoir gérer des segments dans le graphique au

lieu de gérer une région entièrement comme dans le cas des exécutions de média atomiques.

En fait, la gestionnaire d'événements spatiaux possède la liste des exécutions de segments. Ces exécutions de segments sont gérées comme des segments spatiaux (sous-région) de la région spatiale de média structuré. Lorsqu'un événement arrive, elle effectue des traitements en se basant sur cette liste de segments spatiaux. Ainsi, quand l'utilisateur fait bouger la souris sur la région spatiale de média, la liste de segments spatiaux sera parcourue pour tester si la souris est entrée dans ou a bougé sur un des segments de la région. Si oui, ce segment sera marqué, ou affecté d'un traitement immédiat. Par exemple, dans le cas d'un hyperlien sur segment, le contour bleu du segment doit apparaître et le curseur de la souris doit être changé en forme de main. Puis si l'utilisateur appuie sur la souris, le segment repéré va être examiné pour savoir s'il contient des actions le concernant. Si oui, l'action sera testée afin de savoir si elle peut être effectuée localement dans l'exécuteur structuré ou doit être envoyée vers le gestionnaire d'exécution pour effectuer globalement l'action sur la présentation. Par exemple l'affichage de contour du segment ou d'un commentaire du segment peut être réalisé localement, par contre le saut temporel dans la présentation ou vers un document externe doit être effectué par le gestionnaire d'exécution.

A noter que pour des médias continus, la liste de segments spatiaux que le gestionnaire d'événements spatiaux obtient, doit être la liste des segments apparaissant sur l'écran. Quand une exécution de segment est activée, elle est alors immédiatement enregistrée dans la liste des segments spatiaux de la gestion d'événement, et dès qu'elle est arrêtée, elle est enlevée de la liste.

VI.4.2.5 Synthèse de la présentation

La présentation du document utilise un système de gestion d'exécution de document et un ensemble de gestionnaires d'exécution de médias. La gestion d'exécution de document se base sur le graphe d'exécution pour contrôler les présentations des médias individuels et peut gérer à gros grain l'exécution d'un média individuel (le démarrage et la fin du média). De plus la gestion d'exécution n'est appropriée que pour le média statique. Elle ne peut pas gérer des exécutions fines des segments d'un média continu. Elle est aussi incapable de gérer les exécutions indéterminées. De même, les gestionnaires d'exécution de média permettent aussi simplement de démarrer et arrêter un média, et mais ne permettent pas d'interagir finement dans la présentation d'un média.

Nous avons donc proposé d'améliorer ce système de présentation par un ensemble de gestionnaires d'exécutions structurées et d'exécutions de segment. L'exécution structurée permet de gérer des synchronisations plus fines avec des éléments du contenu de média. En particulier, elle permet d'utiliser le temps du média continu qui assure aussi une synchronisation correcte avec les éléments du média continu. De plus l'exécution de segment permet d'effectuer des effets particuliers sur un segment temporel et/ou spatial d'un média structuré. Elle permet aussi d'interagir plus finement avec un média comme placer un hyperlien sur un objet vidéo.

Un point important sur lequel on souhaite insister est que les exécutions structurées et les exécutions de segment sont créés et fonctionnent simplement et facilement grâce au modèle basé sur des descriptions du contenu des médias et des sous-éléments. Ce modèle fournit toutes les informations de la structure du contenu des médias ce qui sont nécessaires pour accéder et procéder plus finement à la fois la présentation globale du document et à la présentation de chaque média.

VI.4.3 La vue temporelle

La vue temporelle permet de visualiser et d'éditer graphiquement et directement le scénario temporel du document multimédia. Les descriptions détaillées de la vue peuvent être trouvées dans les travaux précédents du projet [Tardif 00] [Navarros 01]. Dans cette section, nous présentons les principes de la vue pour faciliter à accéder dans la partie étendue de ce travail.

VI.4.3.1 Principes de la vue temporelle de Madeus

Le principe de la vue temporelle est présenté dans la Figure 77. La vue est construite en utilisant la structure temporelle du document interne pour créer un document de la vue temporelle (voir la section VI.3.4). Le document de la vue temporelle représente les données de la vue temporelle. Ces données sont utilisées par la partie de présentation pour représenter graphiquement ces données. Le document de la vue temporelle garde aussi les liens avec la structure temporelle à partir de laquelle il est construit pour assurer la synchronisation entre la vue et le document interne central.

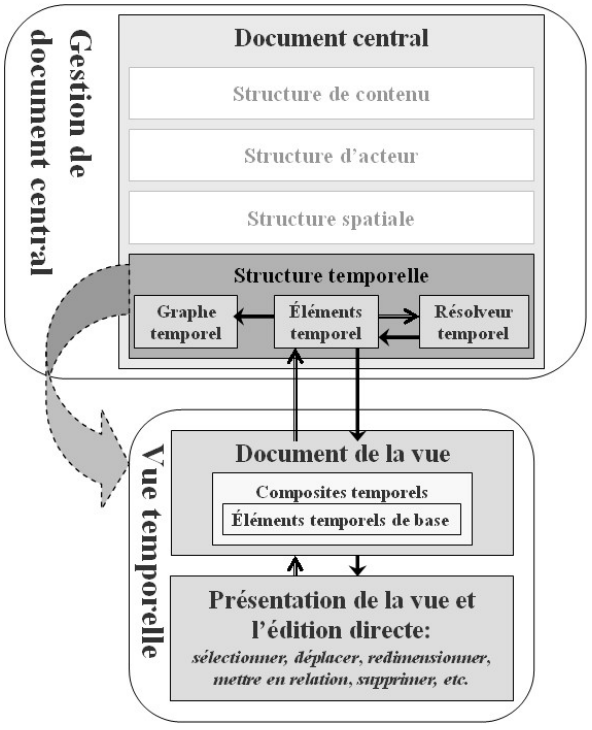

Figure 77. Principe de la vue temporelle.

La vue temporelle représente graphiquement la structure temporelle du document par des rectangles. Les rectangles représentant des éléments temporels composites contiennent des rectangles des éléments temporels qui contient

jusqu'aux éléments temporels de base. Ainsi, la présentation est hiérarchique. Le rectangle d'un élément composite peut être ouvert ou fermé pour visualiser de façon complète ou simplifier le scénario temporel. Ce mécanisme permet de naviguer facilement dans le scénario temporel du document.

L'utilisateur peut directement manipuler les rectangles pour éditer le scénario temporel du document. Ces manipulations sont notifiées au document central pour le mettre à jour. Puis à partir du document central cette mise à jour sera proposée à toutes les autres vues du système. Les vues du système sont ainsi synchronisées.

Cependant la vue temporelle est encore limitée à la représentation temporelle à gros grain des médias. L'utilisateur ne peut pas visualiser ou naviguer plus loin dans le contenu d'un média car la visualisation d'un intervalle de segment est d'une granularité trop petite. La section suivante présente notre solution pour surmonter cette limitation.

VI.4.3.2 Visualisation de la structuration temporelle du contenu des médias

La vue temporelle est étendue pour permettre de visualiser et d'éditer plus finement le contenu des médias. En fait, selon notre modèle de sous intervalle (voir la section V.3.3 du chapitre V), nous avons raffiné la visualisation de l'élément temporel de base pour permettre de visualiser le sous-intervalle. La visualisation des sous-intervalles dans la vue temporelle permet de naviguer et d'effectuer des synchronisations au niveau plus fin. Par exemple, la Figure 78 présente la visualisation des sous-intervalles dans un intervalle de la vidéo et les visualisations des synchronisations fines avec ces sous-intervalles. Bien que les visualisations des sous-intervalles puissent représenter les segments internes d'un média, elles ne sont pas encore suffisantes pour une édition temporelle plus fine et plus sémantique. Parce que l'auteur n'a pas en même temps la visualisation et la possibilité de manipuler la structure du contenu de média pour déterminer ou repérer facilement et sémantiquement des segments structurés à synchroniser.

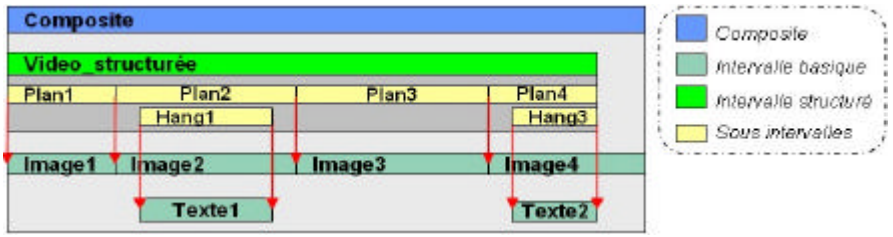

Figure 78. Une représentation hiérarchique dans la vue temporelle avec les sous *timelines*.

Nous avons donc proposé d'intégrer la visualisation de la structure du contenu de média dans la vue temporelle pour offrir des mécanismes d'édition plus confortables à l'auteur. L'exemple dans la Figure 78 peut être représenté avec la visualisation de la structure de la vidéo comme dans la Figure 79.

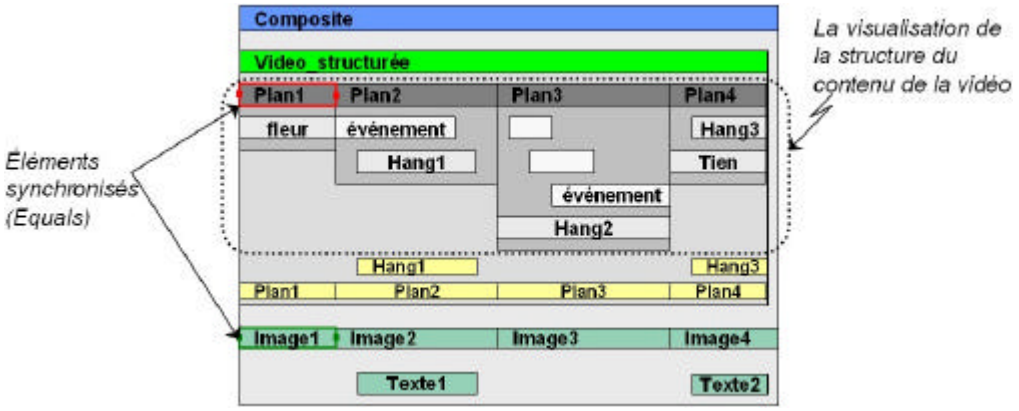

Figure 79. Une représentation de la structure du contenu de la vidéo dans la vue temporelle.

Une telle représentation de la structure du contenu de la vidéo permet à l'auteur de voir tous les éléments de la structure vidéo pour effectuer des synchronisations avec d'autres éléments médias ou repérer les éléments médias synchronisés avec les segments de la structure de la vidéo. Cette visualisation de la structure du contenu de média peut être réalisée facilement grâce à la description du contenu de média et les préparations des structures temporelles et spatiales des éléments de la structure du contenu de média dans le document interne central (voir la section VI.4.1.6).

La visualisation de la structure du contenu de média dans la vue temporelle ne sert qu'à visualiser et repérer finement dans la structure d'un média pour effectuer des synchronisations. Elle permet d'effectuer quelques manipulations limitées, comme naviguer sur la structure, ouvrir ou fermer un niveau de la structure, sélectionner un élément de la structure. Elle ne permet pas d'éditer la structure du contenu de média. En fait, la vue temporelle vise à l'intégration des médias, le fait d'éditer la structure du contenu de média nécessite un autre contexte d'édition qui ne soit pas seulement temporel. Comme on l'a vu, les deux modèles (de média, de document) sont différents ; il est donc nécessaire de laisser à l'auteur la perception de cette différence et de lui permettre de réaliser l'édition de la structure du contenu de média dans une vue spécialisée pour chaque type de média (voir la section VI.4.4 suivante).

Cependant, le fait de ne pas permettre d'éditer le contenu des médias directement dans la vue temporelle du document peut gêner l'auteur puisqu'il doit ouvrir une autre vue pour l'éditer. L'auteur peut donc perdre son contexte d'édition. Toutefois la synchronisation instantanée entre la vue temporelle du document et la vue de structuration du contenu de média limite cet effet perturbant.

VI.4.4 Les vues de médias structurés

La création des vues de média structuré dans le système auteur est un outil intermédiaire de l'analyse et de l'intégration dans la chaîne d'édition de document multimédia (voir les sections III.1, IV.3 et VI.2). Ces vues non seulement permettent de recevoir les descriptions automatiques de l'analyse, de les visualiser et de les intégrer dans le processus d'intégration de média, mais aussi permettent d'éditer manuellement les descriptions pour qu'elles soient plus correctes et plus

appropriées à la composition du document. En fait, dans le modèle de description, certains des éléments sémantiques (par exemple, la scène, le caractère, la relation spatiale/personnelle, etc. du modèle du contenu de la vidéo) ne peuvent pas encore être extraits automatiquement ou même être inférés par des fonctions ou des algorithmes de l'analyse du contenu de média (voir la section III.2). C'est pourquoi, il est nécessaire de fournir un environnement pour aider l'auteur à décrire manuellement les éléments sémantiques. Nous avons découvert à travers notre expérimentation, que même des médias textuels structurés comme HTML ou des documents de familles XML (SVG, SMIL) ont aussi besoin d'être décrits par leur contenu pour mieux les adapter aux besoins de composition de l'auteur (voir la section V.2.4.2.7).

En bref, les vues de médias structurés fournissent une manière automatique pour éditer et visualiser des descriptions de la structure du contenu de média. La suite de cette section présente premièrement les besoins de base de ces vues et ensuite la construction de ces vues dans le système. Nous allons centraliser cette étude sur la vue de vidéo structurée, parce que ce média est le plus complexe et illustre à la fois les médias graphiques et les médias continus. Les résultats de cette vue de vidéo peuvent donc être facilement appliqués aux autres médias.

VI.4.4.1 Les besoins de l'édition des descriptions du contenu des médias

Pour s'adapter au maximum à l'édition de document multimédia, les vues de médias structurés doivent assurer les besoins suivants :

- ♦ intégrer facilement des outils d'analyse existants,
- ♦ réutiliser facilement des descriptions standard,
- ♦ fournir des outils d'extraction/spécification manuelle des éléments temporels, spatiaux et sémantiques du contenu de média,
- ♦ s'assurer que les éditions créent toujours des descriptions valides,
- ♦ visualiser, éditer et naviguer sur des descriptions selon différentes facettes,
- ♦ extraire facilement un élément quelconque de description pour l'intégrer dans le document multimédia,
- ♦ être synchronisée avec les vues d'intégration pour propager automatiquement les modifications apportées aux descriptions.

VI.4.4.2 La vue de vidéo structurée

Dans notre système, la vue de vidéo structurée (voir la Figure 80) permet de visualiser, d'éditer et d'extraire des descriptions du contenu de la vidéo. L'interface présente les descriptions du contenu vidéo selon plusieurs facettes : la vue de structure hiérarchique (1), la vue d'attributs (2), la vue de présentation vidéo (3), la vue de la structure temporelle du contenu ou la vue de structure temporelle de résumé (cette zone est utilisée aussi pour trois autre vues : la vue de la structure des données management, la visualisation de la structure sémantique ou du thésaurus) (4), la vue des informations de la présentation vidéo (5), et enfin la tablette de contrôle d'extraction des objets vidéo (6).

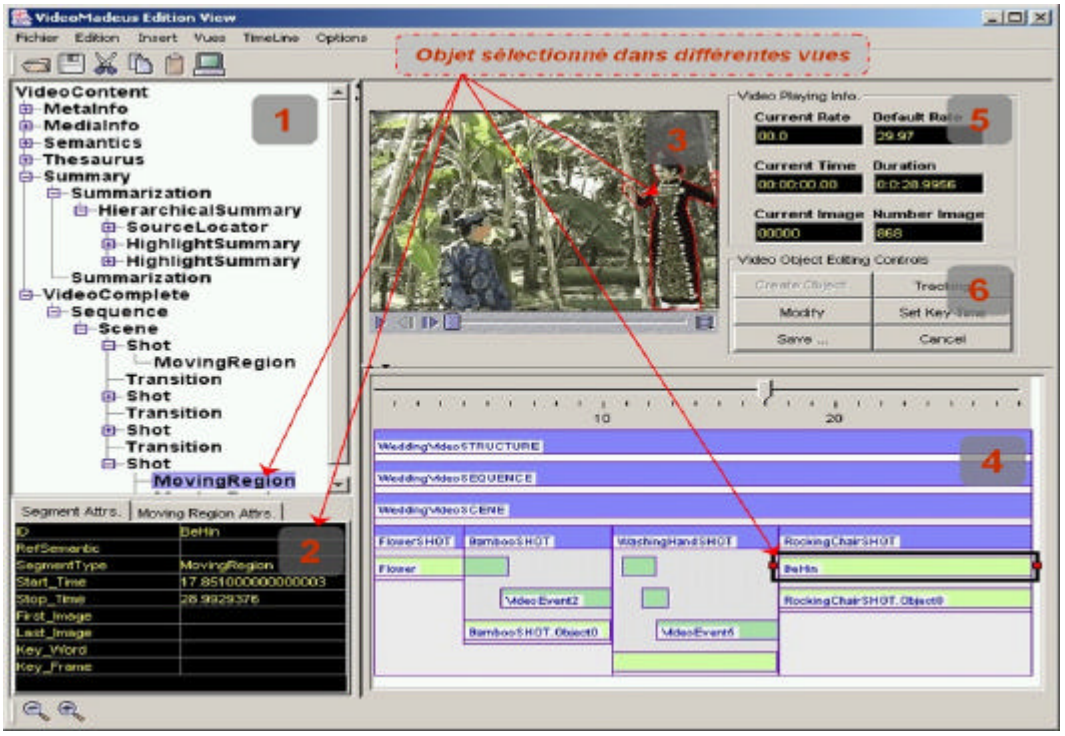

Figure 80. L'interface de la vue vidéo structurée.

A noter que la région (4) de l'interface est une disposition exclusive de plusieurs vues, elle dépend du type de l'élément sélectionné, par exemple, si l'élément sélectionné est un segment vidéo, la vue de la structure temporelle du contenu de la vidéo sera présentée ; si l'élément sélectionné est un segment de résumé, la vue de la structure temporelle de résumé sera affichée ; etc.

Une telle interface multi vues fournit un ensemble d'outils pour la visualisation, la navigation et la modification de la description du contenu vidéo. Ainsi, la vue hiérarchique donne une bonne visualisation et navigation de la structure de description, tandis que la vue temporelle donne une bonne perception et navigation du scénario temporel du contenu de la vidéo. À un niveau plus détaillé, la vue d'attributs donne la visualisation et les capacités d'édition des attributs d'un élément de description sélectionné. La vue présentation permet de jouer chaque segment spatial/temporel sélectionné. La vue des informations donne toutes les informations complémentaires de flux vidéo à chaque instant.

La synchronisation entre des sous vues de la vue vidéo structurée donne à l'auteur la perception de ses actions. Par exemple, si un élément est sélectionné dans la vue hiérarchique, les autres vues sont immédiatement mises à jour pour présenter l'élément sélectionné ; supposons que l'élément sélectionné soit un segment vidéo, l'auteur peut le modifier sur la vue temporelle, par exemple, lorsqu'il redimensionne l'affichage représentant l'élément sélectionné dans la vue temporelle. Dans la vue d'attributs, l'auteur peut voir la valeur chiffrée de cette manipulation, et enfin, dans la vue de présentation, la vidéo est mise en scène conformément à ces manipulations.

Pour aider l'auteur à éditer facilement les descriptions du contenu de la vidéo, la vue fournit des actions d'édition génériques comme la modification des attributs,

l'ajout/suppression des éléments. Des actions plus spécifiques au média sont également accessibles : le groupement des événements en un plan, des plans en une scène, des scènes en une séquence, ou les opérations inverses de dégroupement. De plus l'extraction manuelle des objets mobiles de la vidéo en dessinant directement sur le graphique de la présentation de la vidéo est aussi fournie.

L'intérêt supplémentaire de cette vue vidéo structurée par rapport aux autres applications d'indexation et de consultation du contenu de la vidéo est que notre vue vidéo structurée fournit une sous-vue temporelle hiérarchique qui permet de naviguer et d'éditer facilement le scénario temporel du contenu de la vidéo. En fait, la plupart des outils de vidéo existants ne fournissent qu'une vue temporelle linéaire ou plate. Par exemple, l'application *Vidéoprep*, qui est un outil d'indexation semi-automatique de la vidéo développé au sein du projet MOVI de l'INRIA [Hammoud et al. 00], permet d'extraire automatiquement des informations de structure correspondant à une vidéo et permet d'éditer des regroupements d'images en plan, des objets dans ces plans, des classes d'équivalence d'objets (objets sémantiquement équivalents). Cependant, la navigation temporelle dans le contenu de la vidéo est limitée à une vue linéaire (voir la Figure 81a). L'application *VideoSearch* [Knibb et al. 97] est plus évoluée puisqu'elle permet de visualiser plusieurs objets de la vidéo sur des canaux parallèle, mais cette vue est encore non hiérarchique (voir la Figure 81b).

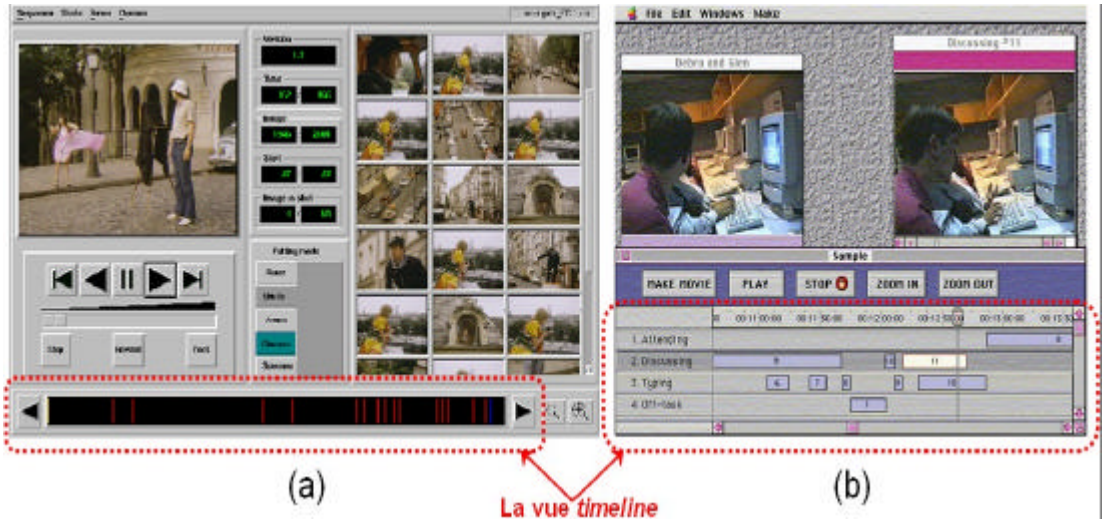

Figure 81. L'interface des applications *Vidéoprep* (a) et *VideoSearch* (b).

En résumé, notre vue de la vidéo structurée est un outil de description du contenu de la vidéo, qui utilise un format compatible avec la norme MPEG-7 (voir les sections V.2, VI.4.1.1). Par cette approche cet environnement est semblable à l'outil *MPEG-7 Visual Annotation Tool* de IBM [Lugeon et al. 00], qui est employé pour décrire des informations audiovisuelles basées sur le schéma standard de description multimédia de la norme MPEG-7. Cet outil permet d'employer tous les schémas de descriptions de MPEG-7 pour décrire manuellement des ressources multimédias (les versions actuelles ne supportent pas des outils d'extraction automatique). Cependant, notre outil est plus approprié pour décrire la structure du contenu de média et il permet d'intégrer des analyseurs du contenu de média et des générateurs qui peuvent générer des descriptions MPEG-7 du contenu de média.

Le fonctionnement de la création et de l'édition d'une nouvelle vidéo structurée

Nous décrivons ici un scénario d'utilisation de cette vue.

Lorsque l'auteur veut ajouter une nouvelle vidéo structurée dans son document, il choisit simplement le fichier de la vidéo (du format *mpeg*, *avi* ou *mov*) à travers le menu *Insérer/Média_Structuré/Vidéo_Structurée*. Un outil d'extraction (détection des changements) intégré dans le système l'analyse et un module d'interprétation regroupe les changements détectés en événements, en plans ou même en scènes pour générer automatiquement des premières descriptions de la structure de vidéo sélectionnée. Dès que les descriptions sont disponibles, une nouvelle vue de vidéo structurée est ouverte pour présenter les premières descriptions de cette vidéo structurée. L'auteur peut alors commencer à éditer les descriptions de la vidéo. L'intérêt du processus est son fonctionnement instantané. Lorsque une vidéo est choisie, l'auteur peut obtenir en temps réel une description initiale qu'il peut facilement continuer à éditer.

Les descriptions générées automatiquement peuvent avoir des erreurs particulièrement au niveau de l'interprétation de la scène (parce que l'algorithme de détection est encore simple, il est basé sur la différence de couleur histogramme entre deux trames vidéo successives). L'auteur peut ajuster ces descriptions et les compléter en ajoutant les niveaux sémantiques (séquence et scène) ou les objets dans les plans qu'il juge à son document. En particulier, l'auteur peut extraire facilement des objets mobiles en spécifiant certains contours clés d'un objet mobile. Le mouvement complet de l'objet est interpolé par des fonctions d'interpolation *linéaire*, *Spline* ou *BSpline* selon différents degrés dépendant du choix de l'auteur.

La vue utilise le schéma de description du contenu de la vidéo (voir la section V.2) pour s'assurer que les éditions dans la vue soit toujours valides. Cette approche de validation est souple, elle permet de mettre à jour facilement des modifications dans le modèle de description.

En particulier, la vue temporelle met en œuvre les contraintes temporelles de la structure du contenu vidéo (voir la section *V.2.2*). Les éditions dans la vue temporelle sont donc toujours validées.

Architecture de la vue vidéo structurée

La vue de la vidéo structurée est une vue extension du système multi vues de Madeus (voir la section VI.3.4). Une représentation de l'architecture générale de la vue média structuré est présentée dans la Figure 82. Elle est construite en se basant sur des descriptions de l'élément structuré (*VideoStructured*, voir les sections V.3.1 et VI.4.1.1) du document interne central et garde toujours la synchronisation avec ces descriptions pour prendre en compte les modifications de la vue et mettre à jour les modifications du système.

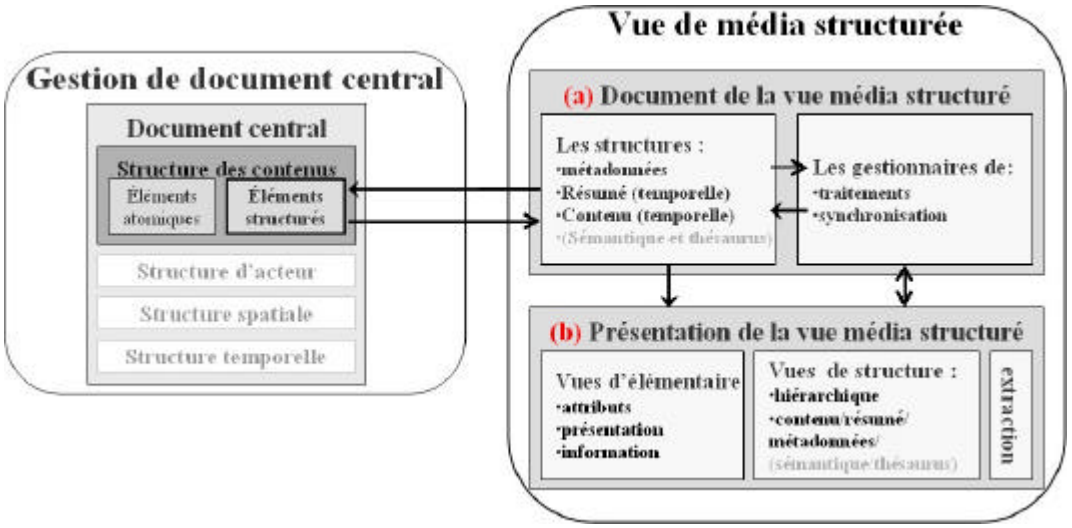

Figure 82. L'architecture de la vue de média structuré.

La vue de la vidéo structurée est un système de multi-vues. Son architecture est organisée en deux parties : (a) la partie de document et (b) la partie de présentation. La partie de document supporte des données et leur gestion pour assurer leur présentation dans la partie présentation.

1. La partie de document possède un document et des gestionnaires :

- Le document de la vue se compose des différentes structures : données management, sémantique, thésaurus, résumé (temporel) et contenu (temporel). Chaque structure est servie à une sous-vue dans la partie de présentation. Pour cette version expérimentale seulement trois structures ont été implémentées. Ce sont : la structure de données management pour la sous vue des données management ; la structure temporelle de résumé pour la sous vue temporelle du résumé ; et la structure temporelle du contenu pour la vue temporelle du contenu. Les présentations et les éditions des descriptions sémantique et thésaurus ne sont pas expérimentées dans cette version. Ces trois sous-vues sont affichées de façon exclusive dans la région (4). La création de ces structures est effectuée en parcourant l'élément vidéo structuré, chaque partie de l'élément structuré peut permettre de créer une structure correspondante, et chaque sous-élément dans la partie permet de créer un élément correspondant de la structure. Chaque structure et ses éléments créés garde toujours un lien avec la partie et l'élément correspondant de l'élément vidéo structuré. Plus précisément :

    a. La structure temporelle de résumé est construite à partir de la description du résumé du contenu *VideoSummaryElement* (voir les sections V.2.1 et VI.4.1.1). Chaque élément de la structure temporelle de résumé possède des informations temporelles extraites de l'élément correspondant de la description du résumé. En fait la structure temporelle du résumé est totalement compatible avec la structure de document de la vue temporelle du système (voir la section VI.4.3) pour pouvoir utiliser des outils existants de la vue temporelle.

b. La structure temporelle du contenu est construite à partir de la description du contenu *VideoStructureElement* (voir les sections V.2.1 et VI.4.1.1). Cette structure est aussi compatible avec la structure de document de la vue temporelle du système.

c. La structure de données management est plus simple car elle référence l'élément *VideoManagementElement* de l'élément structuré (voir les sections V.2.1 VI.4.1.1). Il n'est pas nécessaire de reconstruire d'une nouvelle structure comme dans les deux premiers cas, parce que la vue des données management fait seulement un affichage simple de ces descriptions.

♦ La partie de document est aussi chargé de la gestion de la synchronisation entre les sous vues. En fait, tous les manipulations dans les sous vues de la partie présentation sont notifiées à la partie document. Le gestionnaire de traitement traite les éléments concernant ces manipulations. Ensuite les résultats du traitement sont mis à jour dans les différentes structures du document grâce au gestionnaire de la synchronisation. Dès que les structures sont mises à jour, les sous vues sont informées pour effectuer si besoin le raffermissement graphique.

2. La partie de présentation intègre les différentes sous-vues qui sont : les vues élémentaires, les vues de structure et le contrôle d'extraction.

♦ La vue d'élémentaire consiste en la vue d'attribut (2), la vue présentation (3) et la vue des informations (5). Ces vues permettent de présenter et/ou d'éditer les informations d'un seul élément de description :

a. La vue présentation permet de présenter un segment sélectionné de la vidéo.

b. La vue des informations affiche des informations techniques (la vitesse, numéro d'image, le temps actuel, la durée, etc.) de la présentation d'un segment vidéo sélectionné.

c. La vue d'attribut permet de visualiser tous les attributs d'un élément de description. Elle permet aussi d'éditer la valeur de ces attributs. Lorsque il y a une modification de la valeur d'un attribut, l'élément de cet attribut sera traité, notifié et mis à jour dans tout le système. Si l'attribut est un attribut temporel d'une description de segment vidéo, l'auteur peut voir ses mise à jour sur la vue présentation et la vue temporelle.

♦ Les vues de structure consiste en la vue de structure hiérarchique (1) et les trois vues exclusives dans la région (4).

a. La vue de la structure hiérarchique (1) prend directement l'élément média structuré et affiche la structure de cet élément sous forme d'arbre. Il est très facile d'y naviguer et d'éditer la structure des descriptions de l'élément média structuré.

b. La vue de la structure des données management permet de visualiser et d'éditer des descriptions de données de management dans une visualisation de formulaire (voir la Figure 83b).

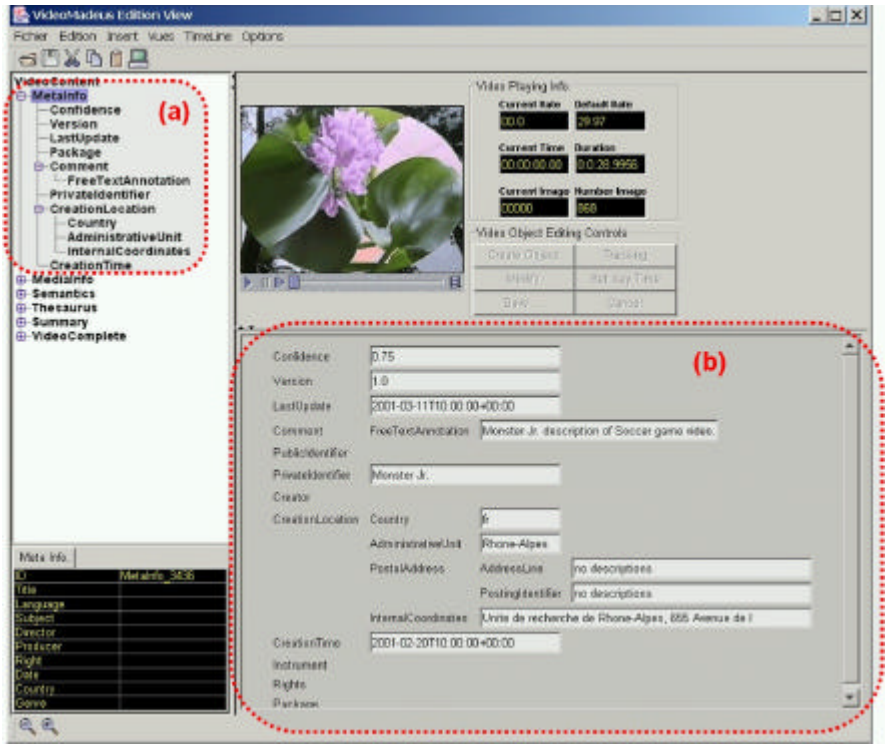

Figure 83. La vue hiérarchique (a) et la vue en formulaire (b) des données management.

c. Les vues de la structure temporelle du résumé (voir la Figure 84b) et du contenu (voir la Figure 85b) permettent de visualiser et d'éditer graphiquement la structure temporelle de la description du résumé et du contenu. En fait ces deux vues temporelles sont les mêmes que la vue temporelle du document (voir la section VI.4.3). Nous pouvons bénéficier cette même vue temporelle au niveau de la vidéo structurée, parce que les structures temporelles de la vidéo structurée sont compatibles avec le document de la vue temporelle du système.

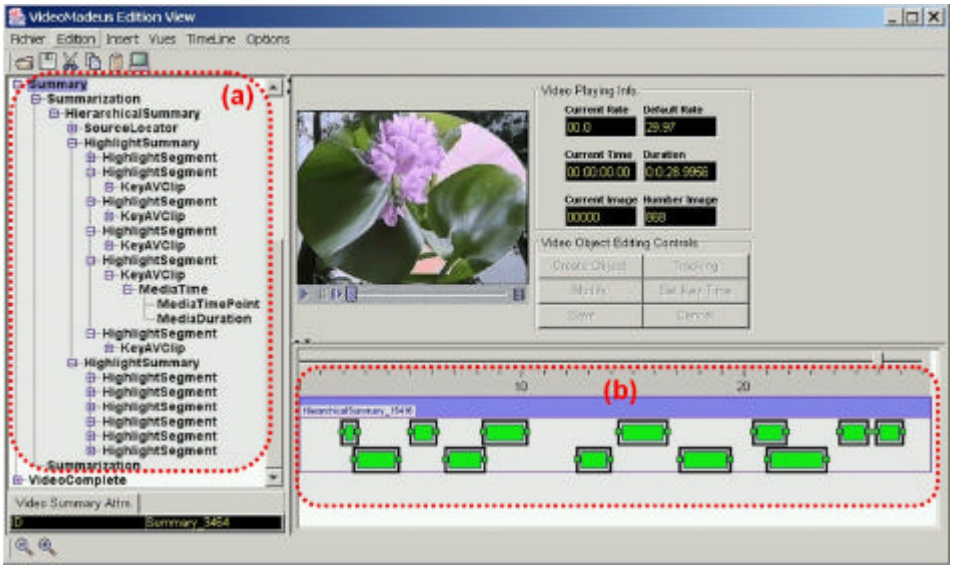

Figure 84. La vue hiérarchique (a) et la vue temporelle (b) du résumé.

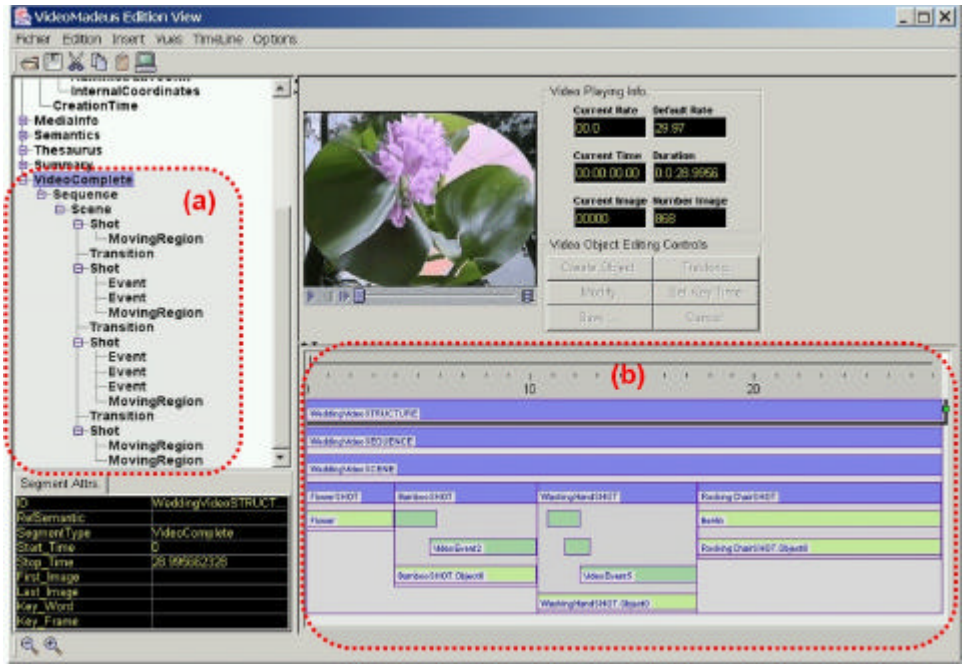

Figure 85. La vue hiérarchique (a) et la vue temporelle (b) du contenu.

♦ La tablette des boutons (6) permet de contrôler l'extraction des objets de la vidéo. Dans notre modèle, les descriptions des objets vidéo peuvent être seulement contenues dans un *plan* vidéo. C'est pourquoi la tablette ne permet que de créer un nouvel objet lorsque un *plan* est sélectionné. Et lorsqu'un objet existant (un élément *MovingRegion*) est sélectionné (voir la Figure 86), la tablette de contrôle permet de modifier les descriptions spatiales et temporelles de cet objet. La création ou/et la modification d'un objet vidéo sont faites directement sur la présentation de la vidéo (voir la Figure 86b). Pour cela, l'auteur affiche l'image appropriée de la vidéo en utilisant des boutons contrôler de la vidéo (voir la Figure 86c). Puis il peut éditer le contour clé de l'objet qu'il souhaite extraire (la fleur dans l'exemple) Il re

commence l'opération sur les différentes images dans les quelles apparaît l'objet. L'auteur peut prévisualiser l'objet extrait pendant l'extraction, et peut modifier un contour clé quelconque pour obtenir un résultat d'extraction plus précis.

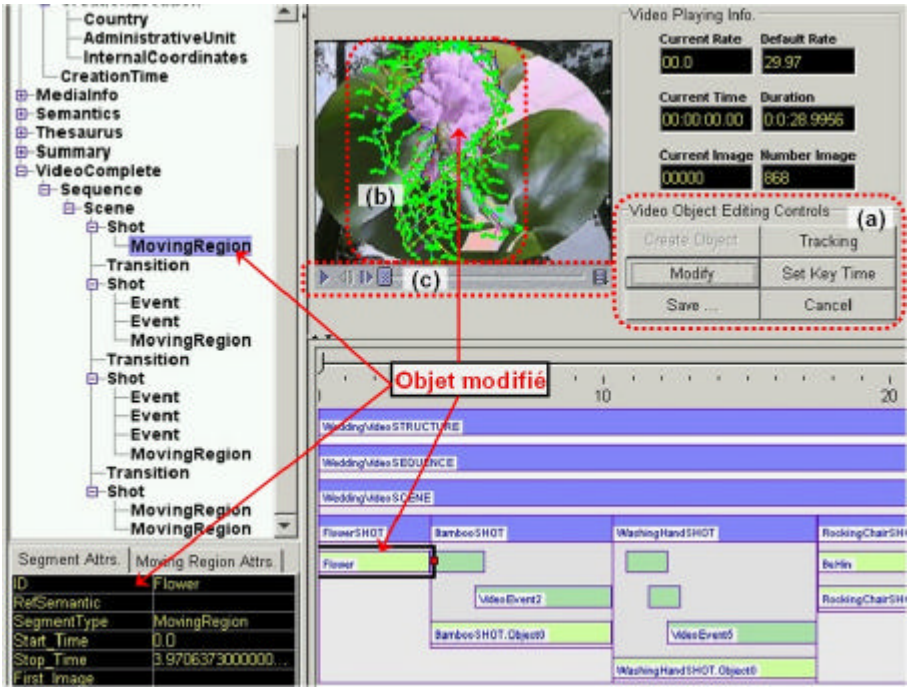

Figure 86. L'extraction et la modification des descriptions d'un objet vidéo.

VI.4.4.3 L'édition du document multimédia avec des médias structurés

L'architecture de la vue de média structuré présentée ci-dessus a des liaisons fortes avec l'environnement auteur de document. Elle permet à l'auteur de facilement utiliser les descriptions en composant des documents multimédia. Par exemple, la Figure 87 présente une scène d'édition de l'environnement auteur Mdéfi qui se compose d'une vue présentation et d'une vue temporelle. Dans cette scène il y a quatre images et textes qui sont présentés. L'objectif ensuite de l'édition est d'insérer un clip vidéo dans la scène et de synchroniser les images et les textes avec les plans du clip vidéo.

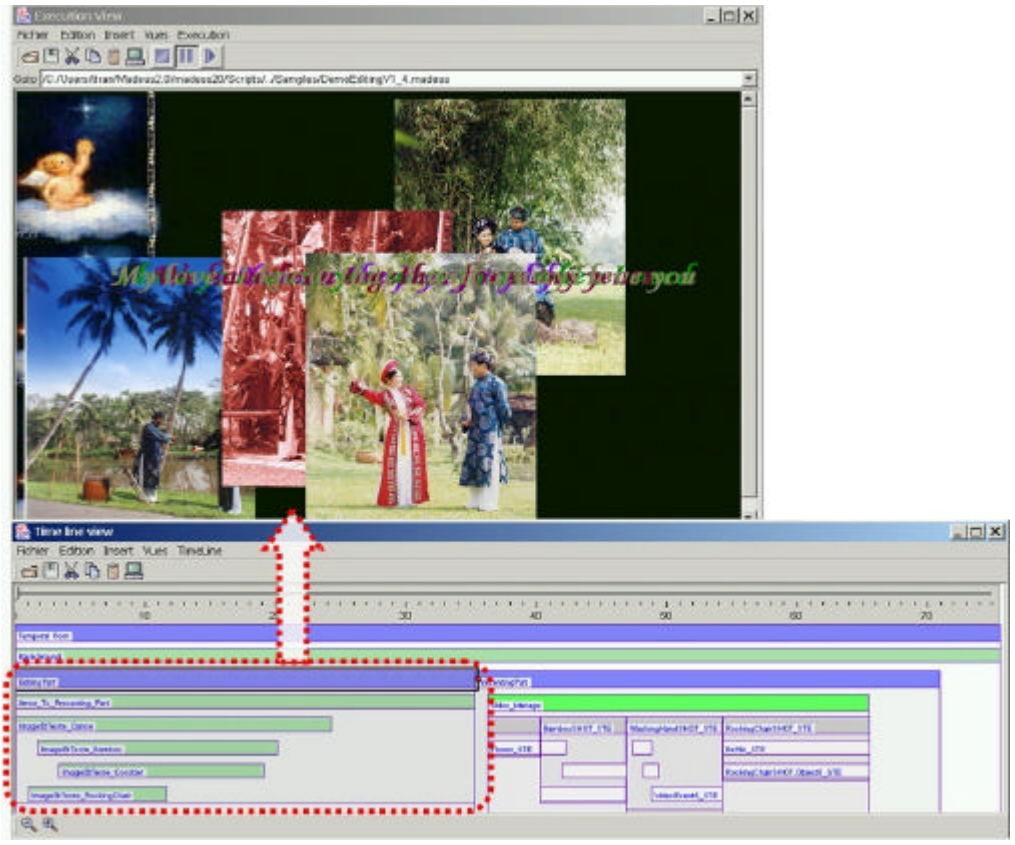

Figure 87. L'interface de l'environnement auteur Mdéfi : la vue présentation et la vue temporelle du document.

Pour faciliter l'identification des plans de la vidéo en éditant le scénario présenté ci-dessus, le clip de vidéo est choisi comme une vidéo structurée (voir la partie *"Le fonctionnement de la création et de l'édition d'une nouvelle vidéo structurée*" de la section VI.4.4.2). Lorsque les descriptions de la vidéo sont disponibles dans la vue média structuré, tous les segments de la vidéo décrits peuvent être facilement insérés dans le scénario du document par des manipulations simples sur le segment sélectionné : le faire glisser et le déposer dans la vue présentation. La Figure 88 présente l'environnement d'édition Mdéfi avec la vue vidéo structurée contenant l'extrait vidéo que l'on souhaite insérer dans le document. La scène de la vidéo structurée est tout d'abord sélectionnée à l'aide d'une des sous-vues de la vue vidéo structurée. Elle est alors glissée et déposée dans la vue présentation du document à l'emplacement spatial souhaité. La présentation temporelle de cette scène vidéo est insérée dans le scénario temporel courant (le scénario des quatre images et textes). Enfin grâce à la représentation de toute la structure de la scène dans la vue temporelle (voir la section VI.4.3), l'auteur peut synchroniser facilement les présentations de chacune des quatre images et textes avec les plans correspondants de la scène vidéo. Les mises en synchronisations (*Equals*) sont effectuées facilement grâce à une tablette de relations (voir la Figure 88d) qui permet de mettre en relations les éléments selon les relations d'Allen [Allen 83].

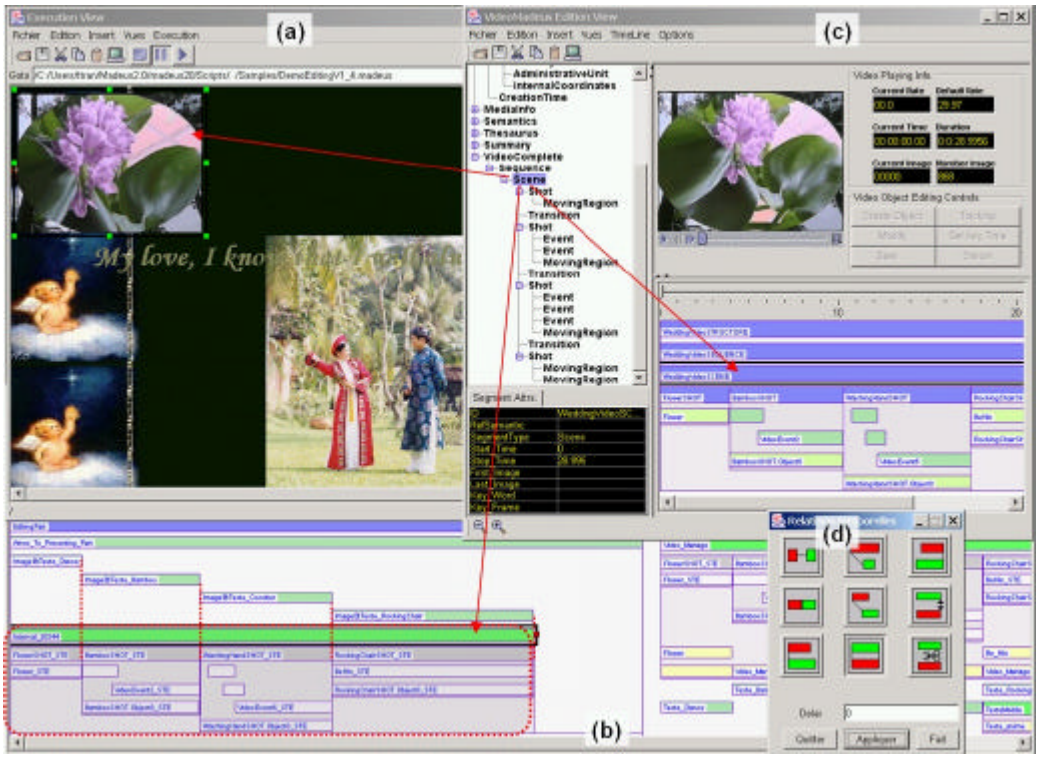

Figure 88. édition du document en utilisant la vue média structuré de l'environnement auteur Mdéfi : (a) la vue présentation du document, (b) la vue temporelle du document, (c) la vue de média structuré, et (d) la tablette de synchronisation.

L'auteur peut aussi choisir une portion spatio-temporelle de la vidéo (un objet de la vidéo) pour l'intégrer dans la présentation du document. La Figure 89 présente une telle composition dans laquelle un seul personnage de la vidéo est intégré dans le document. Une telle présentation de la vidéo est effectuée en se basant sur la description de l'objet de la vidéo. Comme on l'a vu, il n'y a pas besoin d'extraction physique dans cette composition.

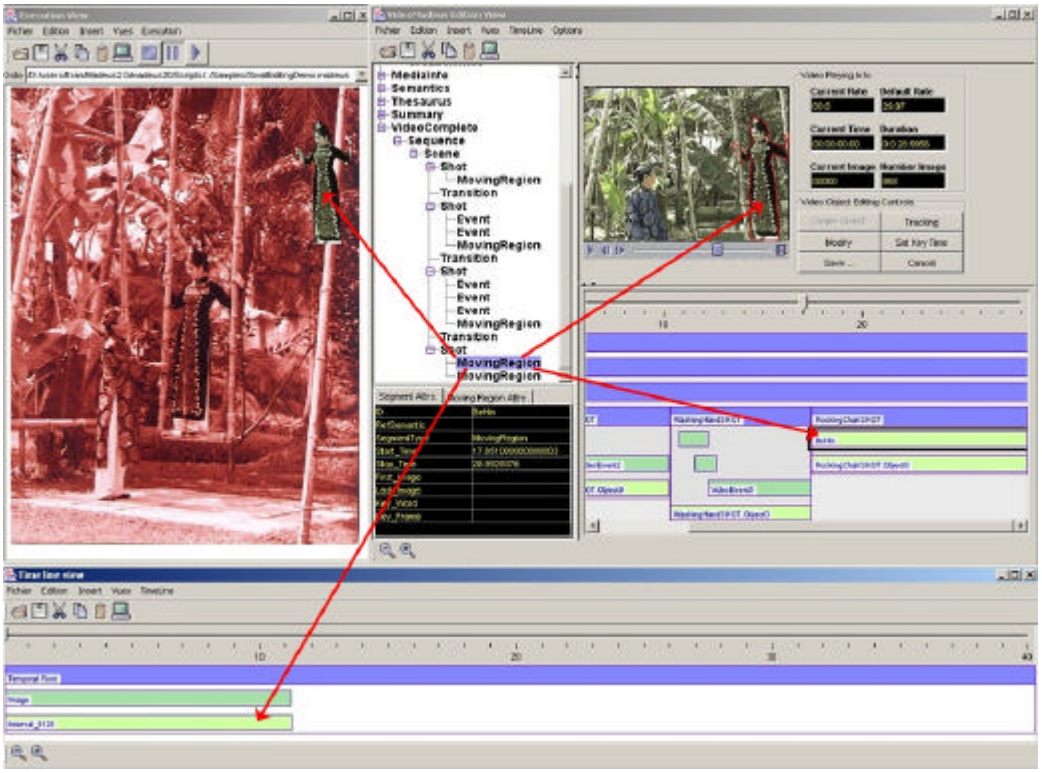

Figure 89. Un exemple de composition du document avec un objet vidéo dans l'environnement auteur Mdéfi.

On peut remarquer que la structure de la scène vidéo est visualisée dans les deux vues : la vue temporelle du document et la vue de vidéo structurée. Cependant, c'est nécessaire car l'objectif de ces visualisations est totalement différent : dans un cas c'est pour éditer la structure temporelle interne du média, en cohérence avec toutes les autres descriptions du média ; et dans l'autre cas, c'est pour réaliser la synchronisation des fragments de cette structure avec d'autre médias (voir plus détaille dans la section VI.4.3).

VI.4.4.4 Les autres vues de médias structurés

L'implémentation expérimentale de la vue de vidéo structurée dans l'environnement auteur a permis d'éditer plus finement un document avec des segments du contenu de la vidéo. Nous avons étendu ces résultats pour implémenter deux autres vues de média : l'audio structurée et l'image structurée. Ces implémentations sont plus simples à réaliser, parce qu'elles sont des cas particuliers de la vue vidéo structurée. En effet, la vue d'audio structurée est implémentée comme la vue de vidéo structurée sans les parties de la présentation (3) et l'extraction d'objet (6) ; à l'inverse la vue d'image structurée peut être appliqué comme la vue de vidéo structurée sans supporter les parties temporelles (les sous vues temporelles dans la régions (4)). Des telles expérimentations plus larges permettent de créer facilement des documents plus complexes, par exemple, le document *Karaoké* avec des synchronisations fines entre des segments de musique et des phrases de chanson ; ou bien un document technique dans lequel des détails techniques d'un appareil (représenté par une image) peuvent être synchronisés avec des commentaires techniques ; etc.

VI.4.4.5 Synthèse

La vue de média structuré est un environnement pour consulter et décrire le contenu de média. Ce type d'application existent déjà plusieurs systèmes [Zhang et al. 95] [Jiang et al. 97] [Knibb et al. 97] [Hammoud et al. 00] [Lugeon et al. 00] [Backfried et al. 01] etc. La contribution du travail présenté ici est que cette vue est construite dans un environnement d'édition et présentation de document multimédia, elle a donc hérité de plusieurs caractéristiques de ce type d'environnement auteur. Par exemple, elle hérite du mécanisme multi vues du système d'édition qui permet d'implémenter une interface intégrant plusieurs facettes. Une telle intégration des vues permet de naviguer, éditer et percevoir parfaitement la description du contenu de média. En particulier, la vue hiérarchique et la vue temporelle peuvent donner des bonnes perceptions de la structure globale de la description et du scénario temporel du contenu. L'architecture de la vue issue aussi du système permet de bien supporter les traitements et les synchronisations internes et externes de la vue, cela permet de coopérer facilement à l'environnement d'édition du document. La vue de média structuré supporte le format de description standard (MPEG-7), mais cette implantation expérimentale n'est concentrée que sur la présentation et l'édition de la description de la **structure du contenu** de média.

VI.5 Conclusion

Nous avons décrit dans ce chapitre notre outil auteur expérimental, appelé Mdéfi, qui fournit un environnement plus confortable et plus logique pour éditer et présenter plus finement des documents multimédias sophistiqués. L'environnement est basé sur une nouvelle architecture d'application d'édition et de présentation pour document multimédia qui permet d'intégrer trois domaines d'application multimédia (*analyse*, *description* et *intégration*) dans un même environnement auteur. L'enchaînement de ces domaines dans l'environnement auteur est souple à cause de l'utilisation de notre modèle d'intégration de description du contenu de média et d'intégration multimédia.

L'environnement est une extension de l'outil auteur Madeus : la structure de document interne du système est enrichie pour s'adapter à l'intégration des descriptions du contenu de média dans le document multimédia ; les modules de présentation et d'édition sont aussi renforcées grâce à la nouvelle structure plus riche de document interne.

Le plus innovant de l'environnement auteur Mdéfi est ses vues expérimentales de média structuré, qui permettent de décrire semi-automatiquement et en temps réel le contenu de média. Les points forts de ces vues par rapport aux autres outils de consultation et d'indexation du contenu multimédia sont : la navigation dans la structure de description du contenu de média ; la capacité de travail en collaboration avec le système d'édition qui permet d'utiliser facilement le contenu décrit pour composer le document multimédia ; et la synchronisation entre ces vues et le système d'édition qui permet de mettre à jour facilement le document lorsqu'il y a des modifications de la description du contenu de média.

Enfin bien que nous ayons expérimenté l'enchaînement complet des trois domaines d'application multimédia, le module d'analyse et d'interprétation reste

encore très modeste. Un module d'analyse et d'interprétation plus puissant doit donc être envisagé afin d'expérimenter plus complètement l'apport des descriptions automatiques de média dans ce type d'application. Cependant l'augmentation de la qualité des descriptions augmente le temps de traitement d'analyse et un compromis doit être trouvé pour notre contexte d'édition.

Chapitre VII. Conclusion

VII.1 Rappel des objectifs

Les travaux présentés dans cette thèse concernent la modélisation de documents multimédias pour leur édition et leur présentation. Notre contribution s'inscrit dans une démarche d'extension des services offerts par les documents multimédias : intégration de médias de plusieurs types, synchronisation entre les médias dans le temps et l'espace, organisation riche de la logique de présentation, interaction avec l'utilisateur, et animation de la présentation des médias. Avec ces caractéristiques, le document multimédia devient de plus en plus important dans les réseaux de communications incluant le Web et la télévision. Cependant, la capacité de composition est encore limitée car les éléments de base sont les médias et non des fragments de média. Sans support d'un mode de composition fin entre des fragments de média, le seul moyen d'accéder aux fragments est d'utiliser un déplacement absolu dans le média, ce qui est fort peu confortable pour l'utilisateur. D'autre part, la définition d'animations est encore concrète ce qui fait augmenter rapidement la complexité du document à cause de l'utilisation répétée d'objets animés. Dans la perspective d'améliorer la composition multimédia, nous nous sommes fixés comme objectifs de contribuer à la fois à :

- La modélisation plus fine du contenu des médias pour répondre aux besoins des applications d'édition et de présentation multimédia ;
- La modélisation plus abstraite de la définition d'animations pour en faciliter la réutilisation dans l'édition et la présentation multimédia ;
- L'intégration de ces modèles dans un modèle d'édition et présentation de document multimédia.

VII.2 Démarche de travail et bilan théorique

Pour atteindre ces objectifs, nous avons étudié les concepts fondamentaux du document multimédia. Nous avons particulièrement concentré notre attention sur l'évolution de la production multimédia dans ses trois générations. Cette étude montre en particulier que la modélisation plus fine du contenu des médias est nécessaire pour produire plus facilement des documents multimédias sophistiqués. C'est particulièrement vrai pour la transformation du document (deuxième génération) et la production automatique de documents en fonction de requêtes de l'utilisateur (troisième génération) qui laissent de plus en plus les tâches de construction multimédia à la machine. Or, nous avons montré l'incapacité des modélisations multimédias existantes pour accéder plus finement au contenu des média, ceci malgré l'évolution importante de la modélisation et du déploiement

multimédia à tous les niveaux, de la description du contenu jusqu'à leur intégration. Cette évolution a abouti aux standards comme MPEG-7, MPEG-4 ou SMIL. Par ailleurs, nous avons analysé les trois types de systèmes de traitement de média existants permettant l'indexation, la production et l'intégration multimédia. Nous avons montré que les besoins pour un environnement auteur idéal ne sont que partiellement couverts par ces systèmes. Un fonctionnement enchaîné de ces trois types de systèmes d'application multimédia est nécessaire à l'environnement auteur idéal.

Le premier résultat de cette thèse est une proposition de modélisation fine du contenu des médias pour l'application de composition multimédia. Nous avons proposé un modèle général avec de multiples axes qui permet de décrire à la fois les informations de management et les informations du contenu de média. Au niveau des descriptions du contenu, nous avons proposé ensuite un modèle avec de multiples abstractions qui contient trois niveaux de description : structure, signification et thésaurus. Bien que ces trois niveaux soient nécessaires à la composition multimédia, nous nous sommes concentrés principalement sur la structure du contenu qui est plus prioritaire pour la composition multimédia. L'innovation majeure de ce résultat est un assemblage judicieux de plusieurs approches de modélisation de la structure du contenu de la vidéo (structure logique en séquences, scènes, plans, structure logique dans le plan, structure d'objet, relation spatio-temporelle et même événement dans le plan) ce qui permet aux utilisateurs et aux programmes d'accéder plus complètement et plus facilement à la structure du contenu de média.

Le deuxième résultat concerne l'adoption de la norme MPEG-7 pour représenter les modèles de description du contenu des médias. Les schémas standard de MPEG-7 sont analysés pour être utilisés de façon la plus adéquate possible à la composition multimédia. Par ce résultat, nous avons montré la puissance l'adaptation de la norme MPEG-7 dans le cas d'un type d'application non prioritaire pour ce standard : l'édition et la présentation multimédia. Une conséquence intéressante pour nos travaux, obtenue grâce à l'utilisation de MPEG-7, est de rendre nos systèmes interopérables avec les autres applications de ce domaine (analyse et indexation par exemple).

Le troisième résultat est la modélisation abstraite de la définition des animations, ce qui permet leur réutilisation dans la composition multimédia. Ce résultat a prouvé que la spécification d'animations abstraites peut être utilisée dans les documents XML pour éviter les répétitions des définitions d'animations que l'on trouve souvent dans les documents Web actuels (avec XHTML+SMIL). Le modèle proposé se veut une extension du modèle d'animations de SMIL.

Le résultat majeur de cette thèse est l'intégration des modèles ci-dessus dans un modèle d'édition et présentation du document multimédia. Pour renforcer la capacité d'édition et de présentation multimédia. Notre contribution se situe principalement dans le raffinement du modèle d'édition et présentation multimédia par la structuration en sous-éléments dans chacune des dimensions du modèle. Ce niveau de structuration supplémentaire permet de déployer parfaitement les descriptions du contenu des médias et de l'animation abstraite dans toutes les dimensions (les actions de présentation, la structure temporelle et la structure

spatiale). Cette intégration des modèles permet de réaliser les synchronisations fines attendues et offre en plus le moyen d'assurer la cohérence de l'ensemble des synchronisations du document : les synchronisations intrinsèques de chaque média avec les synchronisations issues de la composition.

Enfin, le dernier résultat est la conception d'une architecture d'environnement auteur permettant d'enchaîner les trois types d'applications multimédias existants (*analyse*, *description* et *intégration* multimédia). Cette architecture se base sur le modèle d'intégration ci-dessus et permet d'utiliser directement des descriptions du contenu des médias pour la composition multimédia. À partir de cette nouvelle architecture, nous avons conçu et développé un système d'édition et de présentation multimédia appelé Mdéfi. Les composants principaux de ce système sont :

- une nouvelle interface pour accéder aux informations de description d'un objet média,
- des gestionnaires de synchronisation fine qui se basent sur le temps général du système ou sur le temps des médias continus,
- une vue temporelle permettant d'exposer la structure temporelle du contenu des médias,
- et enfin des vues des médias structurés permettant d'éditer semi-automatiquement des descriptions du contenu des médias.

VII.3 Résultats pratiques

Cette étude n'offrait un réel intérêt que si elle était concrétisée par la réalisation d'un prototype, aujourd'hui opérationnel, validant les idées qui en émergeaient. La réalisation de Mdéfi qui se base sur Madeus a comporté une part importante de développement pour mettre en œuvre les modèles proposés.

Mdéfi est un environnement auteur d'intégration pour permettre à la fois d'analyser automatiquement du contenu vidéo, d'éditer manuellement des descriptions du contenu des médias (vidéo, audio et image), et d'intégrer finement des fragments de média (vidéo, texte pur non structuré, HTML, audio et image). La chaîne de composition la plus complète est construite pour la vidéo. Dès qu'une vidéo est choisie en vue de son intégration dans le document, des premières descriptions de son contenu sont générées automatiquement grâce à un module d'analyse/génération du contenu. Ces descriptions sont représentées sous plusieurs formes (hiérarchique, graphique, présentation, etc.) dans la vue de la vidéo structurée. A travers cette vue, l'auteur peut facilement rééditer manuellement les descriptions et les intégrer dans l'environnement de composition multimédia. L'environnement de composition, qui peut exploiter ces descriptions du contenu, permet à l'auteur de réaliser facilement des synchronisations fines.

Le module d'analyse/génération effectue une analyse des différences couleurs entre deux trames successives de la vidéo, puis utilise les résultats de cette analyse pour générer des descriptions selon le modèle du contenu vidéo. Ses capacités d'analyse sont simples mais suffisantes pour obtenir un découpage des vidéos en plans.

Les vues de média structurés (vidéo, audio et image) permettent d'éditer et de naviguer dans la structure des descriptions. L'édition et la navigation sont facilitées par l'approche multi-vues héritée de l'outil Madeus, en particulier, la vue hiérarchique et la vue temporelle. La vue hiérarchique permet de gérer facilement toute la structure de description. Elle permet aussi de modifier la structure en assurant sa validation. La vue graphique de la structure temporelle des descriptions permet de visualiser et de modifier cette structure. Cette vue prend en compte les contraintes temporelles de la structure logique des médias pour assurer leur validation lors des modifications. L'intégration d'un segment média dans l'environnement de composition peut s'effectuer par un simple glisser-coller du segment dans la partie du document souhaitée.

L'environnement d'édition et présentation implémente complètement les modèles de présentation décrits dans le chapitre V. Il permet de créer et de présenter toutes les spécifications fines de document multimédia incluant :

- Les actions sur un objet vidéo : suivi, hyperlien et masquage ;
- La synchronisation fine avec des segments temporels de n'importe quel niveau (scène, plan, événement, segment audio), avec des segments spatiaux (région d'image) et ainsi qu'avec des segments spatio-temporels (objets vidéo en mouvement) ;
- La présentation fragmentée d'un segment issu d'un média. L'extraction de fragment tire parti de la structure logique temporelle (plan, scène, événement, segment audio, …), spatiale (région d'image) et même spatio-temporelle (objet vidéo en mouvement) ;
- L'animation de la présentation des médias, comme le changement de couleur, le mouvement, …

L'édition directe dans l'environnement est encore limitée à quelques expérimentations comme l'intégration d'un objet vidéo (*occurrence*) dans le document et la composition des synchronisations temporelles avec des segments médias. Pour mettre en relation temporelle les segments des structures de média, nous avons construit une vue temporelle du document qui permet de visualiser la structure temporelle du contenu média. L'utilisateur peut ainsi sélectionner facilement des segments médias pour composer les synchronisations fines. De plus, cette vue permet aussi de repérer facilement les alignements fins entre les différents segments des médias.

Enfin grâce au système Mdéfi, nous avons édité et présenté plusieurs types de documents de démonstration complexes : un document *karaoké*, un document cinématographique synchronisant une vidéo/audio avec son script, un document avec de nombreuses animations, etc. Ces documents ont suscité un vif intérêt de la part des personnes qui les ont vus.

VII.4 Perspectives

Cette thèse se place sur le contexte très large du multimédia et les résultats présentés ci-dessus en sont seulement une première étape. En plus des améliorations et compléments des modèles et des réalisations, plusieurs perspectives sont envisageables. Nous avons organisé ces perspectives autour des

domaines multimédias abordés dans ce mémoire : l'analyse, la description, l'intégration et l'application d'édition et présentation multimédia.

VII.4.1 Analyse du contenu de média

Cette thèse a permis de réaliser et expérimenter un environnement de visualisation et de navigation du contenu des médias ainsi la composition fine entre les segments de ce contenu. Cependant, les segments et objets de média utilisés dans l'environnement sont en grande partie obtenus manuellement (à l'aide de l'environnement auteur). Le prototype contient simplement un algorithme de découpage en plan des vidéos. Il est clair que le domaine de l'analyse de médias est très actif et qu'il serait nécessaire d'utiliser ses résultats pour renforcer cette phase du système. Les projets MOVI [MOVI], VISTAS [VISTA], ICTT [ICTT], PAROLE [PAROLE], etc. sont les spécialistes de ce domaine. Par exemple, le travail de R. Hammoud dans [Hammoud 01] permet non seulement de segmenter automatiquement des éléments temporel et spatio-temporel de la vidéo (plans et régions mobiles), mais aussi de les grouper automatiquement dans les éléments plus sémantiques (scènes et objets).

Comme notre système supporte MPEG-7, le standard de représentation des résultats d'analyse, il sera facile d'utiliser les résultats d'analyse de ces outils sans pour autant nécessiter leur intégration au sein de notre environnement. De plus, les contraintes de temps dues à notre environnement interactif nous empêchent d'envisager un appel à un système d'analyse dont la qualité du résultat impose des temps d'exécution très longs.

A l'inverse, l'utilisation de notre environnement dans des systèmes d'analyse permettrait de visualiser les résultats d'analyse, puis d'offrir aux utilisateurs le moyen de modifier ces résultats lorsque ceux-ci sont jugés incorrects. Dans cette direction, nous avons collaboré avec le projet MOVI pour visualiser des résultats de l'indexation du contenu d'un film par son script de scénario. Cette visualisation est présentée comme une présentation multimédia dans laquelle les phrases du script donnent accès aux segments de vidéo correspondants et/ou sont mises en valeur les unes après les autres au fur et à mesure de la présentation des segments vidéo correspondants. Cette voie peut être continuée pour générer plus automatiquement et plus précisément des synchronisations. Elle peut être aussi élargie par la collaboration avec d'autres projets comme PAROLE (reconnaissance de la parole) et ORPAILLEUR (analyse linguistique) pour réaliser la visualisation de l'alignement automatique des paroles et des scripts.

VII.4.2 Description du contenu multimédia

Dans ce travail, nous avons principalement cherché à décrire les structures temporelles et spatiales du contenu des médias. En fait, notre objectif premier était de montrer comment réaliser une intégration fine des médias dans les documents multimédias. Cependant, les descriptions de ce niveau ne permettent de spécifier que des segments concrets pour intégrer des documents multimédias. La description plus abstraite des contenus des médias (au niveau sémantique et thésaurus) et plus complète (notamment par la prise en compte d'autres caractéristiques audiovisuelles), permettra de spécifier des segments de médias de façon plus générique pour mettre en scénario des documents multimédias. La

spécification des segments de média de façon générique peut non seulement simplifier les spécifications du scénario des documents multimédias mais aussi permettre d'envisager la spécification de « scénarios génériques » qui pourront s'appliquer sur des flux de média temps réel.

Le principe de composition que nous proposons à travers nos modèles (médias, document) et notre prototype Mdéfi est basé sur un processus en deux étapes : la description de chaque média (selon le degré de décomposition souhaité), puis l'intégration de ces médias à l'aide de l'environnement auteur : placement de relations spatiales et temporelles, ajout de nœuds composites spatiaux ou temporels. Quelle que soient les qualités de l'environnement auteur, cette approche demande un travail important pour l'auteur puisqu'il doit placer manuellement toutes les relations entre les médias et fragments de média. Pour aller plus loin dans l'automatisation du processus de composition multimédia, il est nécessaire d'étudier la modélisation des relations entre les structures du contenu des médias. Elle peut permettre de décrire des relations entre les éléments et/ou types d'éléments des contenus, ce qui permettra de générer automatiquement des intégrations multimédias. De façon plus large, cette modélisation peut être implantée sous la forme d'une base de données multimédias sémantiques qui peuvent être utilisées pour générer automatiquement des présentations multimédias selon des requêtes de l'utilisateur. Une collaboration avec le projet MOVI a pour but de construire une telle base de données de la vidéo indexée par ses scripts. Cela peut permettre de générer une présentation multimédia en réponse à une requête d'utilisateur au lieu de retourner simplement un segment de vidéo.

VII.4.3 Intégration multimédia

Nous avons proposé dans cette étude une approche d'intégration multimédia qui prend en compte des caractéristiques de trois niveaux applications multimédia (analyse, production et intégration). Le modèle que nous avons proposé reste encore un premier résultat qui permet seulement d'accéder finement et logiquement au contenu des médias. Toutefois elle permet de prévoir des problèmes qu'il nous faut envisager pour aller plus loin.

Intégration multi-types de média. Les types de médias sont de plus en plus riches et variés, ils peuvent être sous format textuel (HTML/XHTML, SVG, SMIL, VRML), sous format binaire (MPEG-1/2/4, AVI, MOV, JPEG, GIV, etc.) ou même sous format de programme exécutable (javascript, applet). La composition multimédia doit supporter le plus largement possible ces types des médias ainsi que leurs portions du contenu. Elle doit être aussi suffisamment souple pour intégrer facilement de nouveaux types de média.

Base de descriptions du contenu des médias. Actuellement les descriptions du contenu des médias sont intégrées dans le document, ce qui rend le document rapidement complexe et très gros. Une autre conséquence est que les descriptions de médias sont difficiles à gérer et à partager entre plusieurs documents. Une base qui contiendrait toutes les descriptions du contenu des médias peut limiter ces problèmes. De plus, une telle base de descriptions peut évoluer vers une base sémantique qui peut renforcer beaucoup la capacité d'intégration de même que l'intégration automatique.

Génération automatique de compositions multimédias. L'évolution actuelle des médias sémantiques (comme Web sémantique et MPEG-7/21) nous permettent d'envisager l'existence de bases de données sémantiques dans lesquelles les relations entre des ressources et les ressources elles-mêmes peuvent être décrites sémantiquement. Cela peut fournir la capacité de générer automatiquement à partir une requête/demande un ensemble de ressources qui peuvent posséder une structure ou des relations sémantiques entre elles. Le problème qui reste à résoudre est de savoir comment présenter cet ensemble de ressources en se basant sur leurs relations sémantiques. Le défi actuel vient de ce que les standards d'intégration actuels comme SMIL ne sont conçus que pour être générés comme le dernier résultat des processus automatiques. Ces formats sont donc très différents des structures et des relations sémantiques. C'est pourquoi il est très difficile de générer automatiquement une présentation multimédia en se basant uniquement sur la conception logique des médias. Actuellement, les techniques de transformation sont utilisées pour générer automatiquement des présentations multimédia adaptées aux profils des utilisateurs à partir d'un document logique (*document métier*, voir la section II.3.3.1.2). Cependant, à cause de la distance entre le modèle de présentation et le modèle métier, le processus de transformation doit passer par plusieurs étapes compliquées. De plus, la solution est souvent spécifique pour un seul type de document métier. En bref, la solution actuelle a pour but d'adapter une classe spécifique de documents à plusieurs environnements de présentation, elle n'a pas la capacité de transformer automatiquement des résultats multimédias des requêtes/demandes qui sont générées dynamiquement et qui donc peuvent avoir une structure très variée. L'étude de l'application de la technologie des transformations pour les documents sources génériques peut être très utile. De plus, des travaux sur des modèles génériques, neutres et interopérables actuelles comme RDF peuvent servir d'infrastructure pour cette étude.

Description de document. Le travail propose d'utiliser des descriptions du contenu d'objet du document. Cependant, la description de la structure du contenu du document lui-même n'est pas encore mentionnée. La création de documents multimédias qui possèdent dans leur contenu des descriptions sémantiques peut de leur faciliter les tâches de traitement (indexation, recherche, etc.). Les normes comme MPEG-7 et RDF peuvent être de bons outils pour cette étude.

Feuilles d'animation. Notre modèle d'animation abstraite qui permet de définir séparément des animations abstraites et de les réutiliser plusieurs fois dans le document est très proche de l'idée des feuilles d'animation (*cascading animation sheets*[18]). Pour prendre en compte totalement cette idée, nous pouvons envisager sauvegarder des animations abstraites dans des feuilles communes au lieu de les stocker dans chaque document concret. Un tel stockage séparé des documents rejoint le même besoin identifié pour les descriptions du contenu des médias. Il permettrait ainsi de réutiliser des animations abstraites non seulement dans un document mais aussi dans plusieurs documents différents.

[18] Proposals for additions to the SVG-specification, pro http://www.pinkjuice.com/SVG/spec-prop.xhtml

VII.4.4 Application d'édition et présentation multimédia

Enfin, l'obtention d'un prototype opérationnel comme Mdéfi, nous permet d'envisager son évaluation pour l'appliquer dans d'autres domaines.

Etudes post-doctorales. Plus précisément, le travail qui sera la suite de cette thèse se déroulera dans le cadre d'un post-doc industriel chez Httv, un spécialiste dans le domaine de l'édition et diffusion de télévision numérique interactive. Le post-doc industriel proposé concerne l'édition de services interactifs de télévision numérique synchronisés avec un contenu audiovisuel. Plus précisément Httv a réalisé un environnement (PrimeTV) intégré permettant l'édition, la production et la consultation de services interactifs autonomes (service plein écran non lié à un contenu audiovisuel). Httv a pour objectif de faire évoluer cet environnent en intégrant la possibilité de synchroniser les éléments interactifs entre eux et avec un support audiovisuel. Les travaux réalisés durant la période du post-doctorat porteront sur le module d'édition de PrimeTV et le module de consultation (« player »). Les études et réalisations porteront sur les aspects suivants :

- ♦ Spécification du format de représentation des données multimédia structurées (synchronisation) adapté aux applications de télévision interactive.
- ♦ Intégration dans le module d'édition de PrimeTV d'une composante temporelle (« Timeline »), pour faciliter l'utilisation de contenus audiovisuels indexés ou bruts.

Editeur MPEG-4. La possibilité d'intégrer des objets du contenu des médias permet de pouvoir considérer le développement d'un éditeur MPEG-4 à partir de ce prototype. Ce type d'application peut être facile à réaliser parce que l'extension du format textuel de MPEG-4 (XMT) prend en compte les modèles de SMIL qui est très similaire à notre modèle. Cependant, un gros travail sera nécessaire pour prendre en compte l'édition de scènes 3D puisque notre travail est limité aux scènes 2D.

Référence

[Allen 83] J.F. Allen. *Maintaining Knowledge about Temporal Intervals*. Comm. ACM, 26(11):832-843, 11 1983.

[Allsopp et al. 01] D. N. Allsopp, P. Beautement, J. Carson and M. Kirton, *Toward Semantic Interoperability in Agent-Based Coalition Command Systems*, Semantic Web Workshop (SWWS), July 30 - August 1, 2001.

[André et al. 89] J. André, R. Furuta et V. Quint, *Structured documents*, Cambridge University Press, Cambridge, 1989.

[André-Obrecht et al. 02] R. André-Obrecht and J. Pinquier, *Reconnaissance et Indexation de documents sonores*, Journée AIM, Bordeaux, France, Juillet 2002.

[Ardebilian 00] M. Ardebilian, *Une contribution pour l'accès par le contenu de la vidéo*, Thèse de doctorat de l'Université de Technologie de Compiègne (UTC), Décembre, 2000.

[Ardizzone et al. 96] E. Ardizzone, M. La Cascia, V. Di Gesu' and C. Valenti, *Content Based Indexing of Image and Video Databases by Global and Shape Features*, Int. Conf on Pattern Recognition, ICPR, Wien, Austria, Aug. 1996.

[Ardizzone et al. 97] E. Ardizzone and M. L. Cascia, *Automatic video database indexing and retrieval*, Multimedia Tools and Applications, 4(1), 1997.

[Auffret et al. 98] G. Auffret, J. Carrive, O. Chevet, T. Dechilly, R. Ronfard, B. Bachimont, *Audiovisual Event Description Interface AEDI v1.0*, User guide, INA, France, 1998.

[Backfried et al. 01] G. Backfried and J. Riedler , *Multimedia Archiving with Real-time Speech and Language Technologies*, IEEE Conference on Information, Communications & Signal Processing (ICICS 2001), Singapore.

[Badros et al. 98] G. J. Badros and A. Borning, *The Cassowary Linear Arithmetic Constraint Solving Algorithm: Interface and Implementation*, Technical Report UW-CSE-98-06-04, June 1998.

[Bailey et al. 01a] B. Bailey, J. Konstan, and J. Carlis, *DEMAIS: Designing Multimedia Applications with Interactive Storyboards*. Proceedings ACM Multimedia 2001.

[Bailey et al. 01b] B. P. Bailey, J. A. Konstan, and J. V. Carlis, *Supporting Multimedia Designers: Towards More Effective Design Tools*, 8th International Conference Multimedia Modeling (MMM01), Amsterdam, 5-7 November 2001.

[Beek et al. 01] P. v. Beek, A. B. Benitez, J. Heuer, J. Martinez, P. Salembier, Y. Shibata, J. R. Smith, T. Walker, *Text of 15938-5 FCD Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes*, ISO/IEC JTC 1/SC 29/WG 11/N3966, March 2001, Singapore.

[Benitez et al. 99] A. B. Benitez, S. Paek, S.-F. Chang, C. Judice, and A. Puri, *Proposal for MPEG-7 Home Media Description Scheme*, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P479, Lancaster, U.K., Feb 1999.

[Berners-Lee et al. 01] T. Berners-Lee, J. Hendler and O. Lassila, *The Semantic Web*, Scientific American Feature Articles, May 2001.

[Bes et al. 01] F. Bes, M. Jourdan, F. Khantache, *A Generic Architecture for Automated Construction of Multimedia Presentations*, 8th International Conference Multimedia Modeling (MMM01), Amsterdam, 5-7 November 2001.

[Blakowski et al. 96] G. Blakowski and R. Steinmetz, *A Media Synchronization Survey: Reference Model, Specication, and Case Studies*, IEEE Journal on Selected Areas in Communications, Vol. 14, No. 1, pp. 5-35, 1996.

[Boll et al. 99a] S. Boll, W. Klas, J. Wandel, *A Cross_Media Adaptation Strategy for Multimedia Presentation*, Proceeding on ACM Multimedia'99, Orlando, Florida, USA, 1999.

[Boll et al. 99b] Susanne Boll, Wolfgang Klas, Utz Westermann, *Multimedia Document Formats - Sealed Fate or Setting Out for New Shores?*, In Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS 99), June 7-11, 1999, Florence, Italy.

[Boll et al. 99c] Susanne Boll, Wolfgang Klas, Utz Westermann, *A Comparison of Multimedia Document Models Concerning Advanced Requirements*, Technical Report TR-99-01, Department of Computer Science, University of Ulm, February 1999, Ulm, Germany.

[Boll et al. 99d] Susanne Boll, Wolfgang Klas, Utz Westermann, *Exploiting OR-DBMS Technology to Implement the ZYX Data Model for Multimedia Documents and Presentations*, In Proceedings 8. GI-Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Freiburg, March 1-3, 1999.

[Boll et al. 99e] S. Boll and W. Klas. *ZYX — A Semantic Model for Multimedia Documents and Presentations*. In Proc. of the 8th IFIP Conference on Data Semantics (DS-8): "Semantic Issues in Multimedia Systems". Kluwer Academic Publishers, Rotorua, New Zealand, January 1999.

[Boll et al. 00] S. Boll, W. Klas, *-ZYX – A Multimedia Document Model for Reuse and Adaptation of Multimedia Content*, Transaction on Knowledge and Data Engineering, DS-8 Special Issue, IEEE, 2000.

[Buchanan et al. 93] M.C. Buchanan and P.T. Zellweger (1993) *Automatically Generating Consistent Schedules for Multimedia Documents*, ACM/Springer-Verlag Journal of Multimedia Systems, vol. 1, no. 2, 1993.

[Carcone 97] L. Carcone, *Formatage spatial dans un environnement d'édition/presentation de documents multimédias*, Mémoire, Cnam, décembre 1997.

[Carrive 00] J. Carrive, *Classification de séquences audiovisuelles*, Thèse de doctorat de l'niversité Paris 6, Septembre 2000.

[Carrive et al. 00] J. Carrive, F. Pachet, R. Ronfard. *Logiques de descriptions pour l'analyse structurelle de film.* Ingénierie des connaissances, in Ingéniérie des connaissances, évolutions récents et nouveau défis, J. Charlet, M. Zacklad, G. Kassel, D. Bourigault (Ed.), Eyrolles, pp. 423-438, 2000.

[Celentano et al. 99] A. Celentano, O. Gaggi, *A Synchronization Model for Hypermedia Documents Naviga-tion*, ACM Symposium on Applied Computing 2000.

[Chua et al. 95] T. S. Chua and L. Q. Ruan, *A video retrieval and sequencing system*, ACM Transactions on Information System, 13(4), 1995.

[Cieplinski et al. 01] L. Cieplinski, M. Kim, J. Ohm, M. Pickering, A. Yamada, Text of ISO/IEC 15938-3/FCD Information Technology – *Multimedia Content Description Interface* – Part 3 Visual,ISO/IEC JTC1/SC29/WG11/N4062, Singapore, March 2001.

[Chang et al. 87] S.-K. Chang, Q. Y. Shi, and C. Y. Yan, *Iconic indexing by 2-D strings*, IEEE Trans. Pattern Anal. Machine Intell, 9(3): 413-428, May, 1987.

[Dattolo et al. 01] Dattolo, V. Loia, M. Quaggetto, *Synchronizing Interactive Web Documents with FD-Java Constraints*, Handbook of Software Enginnering and Knowledge Engineering, World Scientific Press, December, 2001.

[Day 01] N. Day, *MPEG-7 Projects and Demos*, AHG on MPEG-7 Applications and Promotions to Industry, March. 2001 – Singapore.

[Decker et al. 99] C. Decker, and M. S. Hacid, *A database approche for modeling and querying video data*, In IEEE Data Engineering, Australia, 1999.

[Decker et al. 00] S. Decker, D. Fensel, F. v. Harmelen, I. Horrocks, S. Melnik, M. Klein and J. Broekstra, *Knowledge Representation on the Web*, Proceedings of the International Workshop on Description Logics (DL2000).

[Dillon et al. 98] C. Dillon, T. Caelli, *Learning Image Annotation: The CITE System*, Videre: Journal of Computer Vision Research, Quarterly Journal, Volume 1, Number 2, Winter 1998.

[Dubuisson et al. 01] M.-P. Dubuisson-Jolly and A. Gupta, *Tracking Deformable Templates Using a Shortest Path Algorithm*, Computer Vision an Image Understanding, 2001.

[Duda et al. 95] Duda A, Keramane C, *Structured Temporal Composition of Multimedia Data*, In: Proceedings of the 1st IEEE InternationalWorkshop for MM-DBMSs, IEEE Computer Society, Los Alamitos, Calif, 1995.

[Dumas et al. 00] M. Dumas, R. Lozano, M.-C. Fauvet, H. Martin and P.-C. Scholl, *Orthogonally modeling video structuration and annotation: exploiting the concept of granularity*, In Proceedings of the AAAI-2000 Workshop on Spatial and Temporal Granularity, Austin, Texas, July 2000.

[Flickner et al. 95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, *Query by Image and Video Content: the QBIC System*, IEEE Computer , September 1995.

[Friedlander et al. 96] N. Friedlander, R. Baecker, A. J Rosenthal, E. Smith, *MAD: A Movie Authoring and Design System*, Electronic Proceeding, CHI 96.

[Gates 01] B. Gates, *Why We're Building .NET Technology*, http://www.microsoft.com/presspass/misc/06-18BillGNet.asp, June 2001.

[Gauvain et al. 99] J.-L. Gauvain, L. Lamel and G. Adda, *Audio Partitioning and Transcription for Broadcast Data Indexation*, European Workshop on Content-Based Multimedia Indexing (CBMI'99), Toulouse, France, October 25-27, 1999.

[Hakkoymaz et al. 99] V. Hakkoymaz, J. Kraft and G. Ozsoyoglu, *Constraint-Based Automation of Multimedia Presentation Assembly*, Proceedings of ACM Multimedia'99, ACM Press, Orlando, USA, November 1999.

[Hammoud et al. 98] R. Hammoud, L. Chen and D. Fontaine, *An Extensible Spatial-Temporal Model for Semantic Video Segmentation*, First International Forum on Multimedia and Image Processing, Anchorage, Alaska, 10-14 Mai, 1998.

[Hammoud et al. 00] R. Hammoud and R. Mohr, *Interactive Tools for Constructing and Browsing Structures for Movie Films*, ACM Multimedia, Los Angeles, California, USA, pp. 497-498 (Demo session), October 30 - November 3, 2000.

[Hammoud 01] R. Hammoud, *Constructing and Browsing of Interactive videos*, Thèse de doctorat, INPG – Grenoble, Février 2001.

[Hardman et al. 93] L. Hardman, G. van Rossum, D. C. A. Bulterman, *The Amsterdam Hypermedia Model: extending hypertext to support real multimedia*, Hypermedia, May 1993, 5(1).

[Hsu et al. 99] L. H. Hsu, P. Liu and T. Dawidowsky, *A Multimedia Authoring-in-the-Large Environment to Support Complex Product Documentation*, Multimedia Tools and Application 8, 11-64 (1999).

[Hunter 01] J. Hunter, *Adding Multimedia to the Semantic Web-Building an MPEG-7 Ontology*, Semantic Web Workshop (SWWS), July 30 - August 1, 2001.

[Hunter et al. 01] J. Hunter, C. Lagoze, *Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles*, WWW10, HongKong, May 2001.

[Hunter et al. 00] J. Hunter, J. M. Martínez, E. Oltmans, *MPEG-7 harmonisation with Dublin Core: current status and concernsII*,

[Hunter 99] J. Hunter, *A Proposal for an MPEG-7 Description Definition Language*, MPEG-7 AHG Test and Evaluation Meeting, 15-19 Feb 1999, Lancaster.

[Hunter et al. 99] J. Hunter, L. Armstrong, *A Comparison of Schemas for Video Metadata Representation*, WWW8, Toronto, May 10-14, 1999.

[Hunter 98] J. Hunter, R. Iannella, *The Application of Metadata Standards to Video Indexing*, Second European Conference on Research and Advanced Technology for Digital Libraries ECDL'98, Crete, Greece, 19 - 23 September, 1998.

[Jacopo et al. 96] M. C. Jacopo, D.B. Alberto, D. Lucarella and He Wenxue, *Multiperspective Navigation of Movies*. Journal of Visual Languages and Computing (1996) 7, 445-466.

[Jiang et al. 97] H. Jiang, D. Montesi and K. Elmagarmid, *Videotext database system*, in IEEE int'l Conf. on Multimedia Computing and System, Ontario, Canada, June 1997.

[Jourdan et al. 98a] Jourdan M., Roisin C., Tardif L., *Édition et Visualisation Interactive de Documents Multimedia*, Proceedings of Electronic Publishing'98, St Malo, LNCS n.1375, pp. 370-380, Springer, Avril 1998.

[Jourdan et al. 98b] Jourdan M., Roisin C., Tardif L., *Multiviews Interfaces for Multimedia Authoring Environments*, Proceedings of the 5th Conference on Multimedia Modeling, pp. 72-79, IEEE Computer Society, Lausanne, Octobre 1998.

[Jourdan et al. 98c] Jourdan M., Layaïda N., Roisin C., Sabry-Ismail L., Tardif L., *Madeus, an Authoring Environment for Interactive Multimedia Documents*, ACM Multimedia'98, pp. 267-272, ACM, Bristol (UK), Septembre 1998.

[Jourdan et al. 00] M. Jourdan, C. Roisin, L. Tardif, *A Scalable Toolkit for Designing Multimedia Authoring Environments*, *Multimedia Tools and Applications Kluwer Academic Publishers*, vol. 12, num. 2/3, pp. 257-279, November 2000.

[Jourdan et al. 01] M. Jourdan and F. Bes, *A new step towards multimedia documents generation*, International Conference on Media Futures, pp. 25-28, Florence, Italy, 8-9 May 2001.

[Kahan et al. 01] J. Kahan, M.-R. Koivunen, E. Prud'Hommeaux, and R. R. Swick, *Annotea: An Open RDF Infrastructure for Shared Web Annotations*, in Proc. of the WWW10 International Conference, Hong Kong, May 2001.

[Keylly et al. 95] P. M. Kelly, T. M. Cannon and D. R. Hush, *Query by image example : the CANDID approach*, Proc. of the SPIE, Storage and Retrieval for Image and Video Databases III, Vol. 2420, pages 238-248, 1995.

[Kim et al. 00] M. Kim, S. Wood, L.T. Cheok, *Extensible MPEG-4 textual format (XMT)*, ACM Press, pages: 71 – 74, Series-Proceeding-Article, New York, NY, USA, 2000.

[Knibb et al. 97] K. Knibb, Jan Herrington, Judy Denham, *Digital Video Analysis of a Multimedia Product*, ASCILITE97, 7-10 Decembre, 1997.

[Koemen 01] R. Koenen, *From MPEG-1 to MPEG-21: Creating an Interoperable Multimedia Infrastructure*, ISO/IEC JTC1/SC29/WG11 N4518, Pattaya, December 2001.

[Kunze 99] J. Kunze, *Encoding Dublin Core Metadata in HTML*, Dublin Core Metadata Initiative December 1999.

[Kunze et al. 01] M. Kunze and D. Roesner, *An XML-based approach for the presentation and exploitation of extracted information*, Proceedings of the First International Workshop on Web Document Analysis (WDA2001), Seattle, Washington, USA, September 8, 2001.

[Lagoze et al. 01] C. Lagoze, J. Hunter, *The ABC Ontology and Model*, International Conference on Metadata, October 2001.

[LaCascia et al. 96] M. La Cascia and E. Ardizzone, *"JACOB: Just a content-based query system for video databases"*, ICASSP-96, May 7-10, Atlanta, Georgia (USA), 1996.

[Layaïda 97] Nabil Layaïda, *Madeus : Système d'édition et de présentation de documents structurés multimédia*, Thèse, Université Joseph Fourier, juin 1997.

[Lewis 96] Paul H. Lewis, Hugh C. Davis, Steve R. Griffiths, Wendy Hall, Rob J. Wilkins, *Media-based Navigation with Generic Links*, Hypertext'96, Washington DC USA.

[Li et al. 98] C.-S. Li, R. Mohan and J. R. Smith, *Multimedia Content Description in The Infopyramid*, IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing (ICASSP) , Seattle, WA, Special session on Signal Processing in Modern Multimedia Standards, May, 1998.

[Lin et al. 97] Y.-T. Lin and Y.-L. Chang, *Tracking Deformable Objects with the Active Contour Model*, Proceedings of the 1997 International Conference on Multimedia Computing and System (ICMCS'97), 1997.

[Little et al. 90] T.D.C. Little and A. Ghafoor, *Synchronization and Storage Models for Multimedia Objects*, IEEE J. on Selected Areas of Communications, vol. 8, no. 3, pp. 413-427, April 1990.

[Little et al. 93] Little T, Ghafoor A, *Interval-Based Conceptual Models for Time Dependent Multimedia Dat*, IEEE Trans Knowl Data Eng. 5(4):551–563, 1993.

[Lugeon et al. 00] B. Lugeon and J. R. Smith, *MPEG-7 Visual Authoring Tool*, IBM T. J. Watson Research Center, 2000.

[Mariano 01] G. Mariano, *Tech giants push MPEG-4 standard*, CNET News.com, October 4, 2001.

[Matsuyama et al. 90] Matsuyama, T. and Hwang, V. *SIGMA: A Knowledge-Based Aerial Image Understanding System*. Plenum Press, 1990.

[Merialdo et al. 99] B. Merialdo et al., *Automatic Construction of Personalized TV News Programs*, Proceedings of ACM Multimedia'99, ACM Press, Orlando, USA, November 1999.

[Meyer et al. 95] T. Meyer-Boudnik and W. Effelsberg, *MHEG Explained*, IEEE Multimedia 1995.

[Mikolajczyk et al. 01] Krystian Mikolajczyk, Ragini Choudhury, Cordelia Schmid, *Face detection in a video sequence - a temporal approach*, Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA , Decembre, 2001.

[Nanard 01] M. Nanard and J. Nanard, "*Towards Multimedia Computing, Lessons Learned From A Half Century of Computer Science*", International Conference on Media Futures, Florence, Italy -8-9 May 2001.

[Navarros 01] P. Navarros, *Edition de documents multimédias SMIL*, Mémoire d'ingénieur C.N.A.M. en Informatique, février 2001.

[Newman et al. 00] M.W. Newman, and J.A. Landay. *Sitemaps, Storyboards, and Specifications: A Sketch of Web Site Design Practice*. Designing Interactive Systems, 2000.

[Oomoto et al. 93] E. Oomoto and K. Tanaka, *Ovid: Design and implementation of a video-object batabase system*, IEEE Trans. on Knowledge and Data Engineering, 5, August 1993.

[Ort et al. 02] E. Ort and R. Mandava, *JavaTM Web Services Developer Pack Part 1: Registration and the JAXR API*, Article in a series on the Java Web Services Developer Pack, February 2002, http://developer.java.sun.com/developer/technicalArticles/WebServices/WSPack/.

[Ossenbruggen et al. 99] J. Ossenbruggen, L. Hardman and D. Bulterman, *Multimedia document abstractions for multi-platform delivery publishing*, 1999.

[Ossenbruggen et al. 01] J. Ossenbruggen, J. Geurts, F. Cornelissen, L. Hardman and L. Rutledge, *Towards Second and Third Generation Web-based Multimedia*, WWW10, May 1-5, 2001, Hong Kong.

[Ossenbruggen et al. 02] J. v. Ossenbruggen and Lynda Hardman. *Smart Style on the Semantic Web*. In: Semantic Web Workshop, WWW2002, May 2002.

[Pachet et al. 00] F. Pachet, P. Roy and D. Cazaly, *A Combinatorial approach to content-based music selection*, IEEE Multimedia, March 2000.

[Paek et al. 99a] S. Paek, A. B. Benitez, and S.-F. Chang, *Self-Describing Schemes for Interoperable MPEG-7 Multimedia Content Descriptions*, Proceedings of the SPIE 1999 Conference on Visual Communications and Image Processing (IST/SPIE-1999), Vol. 3653, San Jose, CA, Jan 1999.

[Paek et al. 99b] S. Paek, A. B. Benitez, S.-F. Chang, A. Eleftheriadis, A. Puri, Q. Huang, C.-S. Li, J. RSmith, and L. D. Bergman, *Proposal for MPEG-7 Video Description Scheme*, Proposal to ISO/IEC JTC1/SC29/WG11 MPEG99/P481, Lancaster, U.K., Feb 1999.

[Pentland et al. 93] A. Pentland, R. W. Picard and S. Sclaroff, *Photobook: Content-Based Manipulation of Image Databases*, MIT Media Laboratory Perceptual Computing Technical report, No 255, Nov 1993

[Pfeiffer et al. 00] S. Pfeiffer, U. Srinivasan, *TV Anytime as an application scenario for MPEG-7*, Workshop on Standards, Interoperability and Practice, Proc. ACM Multimedia 2000, Los Angeles, October 2000.

[Ponceleon et al. 99] D. Ponceleon, A. Amir, S. Srinivasan, T. Syeda-Mahmood, and D. Petkovic, *CueVideo: Automated Multimedia Indexing and Retrieval*, Proceedings of the seventh ACM international conference on Multimedia, Orlando, Florida, United States, 1999.

[Price 93] R. Price, *MHEG : An Introduction to the futureInternational Standard for Hypermedia Oject Interchange*, ACM Multimedia 93, USA, 6-93.

[Roisin et al. 99] C. Roisin, T. Tran Thuong, L. Villard, *Integration of structured video in a multimedia authoring system*, Proc. of the Eurographics Multimedia'99 Workshop, Springer Computer Science, ed., pp. 133-142, Milan, septembre 1999.

[Roisin et al. 00] C. Roisin, Tien Tran_Thuong, Lionel Villard, *A Proposal for a Video Modeling for Composing Multimedia Document*, International Conference on Multimedia Modeling 2000 (MMM2000), 13 to 15 Nov., 2000 at Nagano, Japan.

[Rust et al. 00] G. Rust, M. Bide, *The <indecs> metadata framework Principles, model and data dictionary*, WP1a-006-2.0, June 2000.

[Rutledge et al. 01a] L. Rutledge and L. Hardman, *The rise and fall of multimedia authoring*, International Conference on Media Futures, Florence, Italy, 8-9 May 2001.

[Rutledge et al. 01b] L. Rutledge and P. Schmitz, *Improving Media Fragment Integration In Emerging Web Formats*, 8th International Conference Multimedia Modeling (MMM01), Amsterdam, 5-7 November 2001.

[Rutledge et al. 00] L. Rutledge, B. Bailey, J. Ossenbruggen, L. Hardman, and J. Geurts. *Generating Presentation Constraints from Rhetorical Structure. In:* Proceedings of the 11th ACM conference on Hypertext and Hypermedia (pages 19-28), May 30 -- June 3, 2000, San Antonio, Texas, USA.

[Rutledge et al. 99a] L. Rutledge et al., *Adaptable Hypermedia with Web Standards and Tools*, The Active Web -- A British HCI Group Day Conference, London, January 1999.

[Rutledge et al. 99b] L. Rutledge, L. Hardman, J. v. Ossenbruggen, Bul terman D. C.A., *Mix'n'Match: Exchangeable Modules of Hypermedia Style*, Proceeding of ACM Hypertext'99, 1999.

[Saarela 98] J. Saarela, *Video Content Models based on RDF*, W3C workshop on "Television and the Web", Sophia-Antipolis, France, June 1998.

[Sabry 99] Loay Sabry-Ismaïl, *Schéma d'exécution pour les documents multimédia distribués*, Doctorat d'informatique, Université Joseph Fourier, janvier 1999.

[Salembier et al. 01] P. Salembier and J. Smith, *MPEG-7 Multimedia Description Schemes*, IEEE Transac-tions on Circuits and Systems for Video Technology, Vol. 11, NO. 6, June 2001.

[Schmitz et al. 01] P. Schmitz, A. Cohen, K. Day, *The SMIL 2.0 Animation Modules*, W3C Recommendation: Synchronized Multimedia Integration Language (SMIL 2.0), 07 August 2001. (see also: http://www.w3.org/TR/smil20/animation.html).

[Seyrat 01] C. Seyrat, *MPEG 7 et binarisation XML*, Ecole d'été, Production et Diffusion sur l'Internet de Documents Multimédias Synchronisés (PDMS 2001), à Autrans (Isère), France, du 27 au 30 août 2001.

[Smith et al. 96] J. R. Smith and S-F Chang, *VisualSEEk: a fully automated content-based image query system*, ACM Multimedia'96, Boston MA, USA, pages 87-98, 1996.

[Smith et al. 00] J. R. Smith and A. B. Benitez, *Conceptual Modeling of Audio-Visual Content*, Proceedings of the 2000 International Conference On Multimedia & Expo (ICME-2000), New York, NY, July 30-Aug 2, 2000.

[Stefan et al. 01] S. Stefan, G. Engels, *UML-based Behavior Specification of Interactive Multimedia Applications*, In IEEE Symposia on Human-Centric Computing Languages and Environments, Stresa, Italy, September 2001.

[Tardif 00] L. Tardif, *Kaomi: réalisation d'un boite à outils pour la construction d'environnements d'édition de documents multimédia*, Thèse de doctorat, INPG – Grenoble, Décembre 2000.

[Tirakis et al. 99] A. Tirakis, P. Katalagarianos, M. Papathomas, C. Hamilakis, *Distributed audio-Visual Archives Network (DiVAN)*, International Conference on Multimedia Computing and System, Florence Italy, June 7-11, 1999.

[Tonomura et al. 94] Y. Tonomura, A. Akutsu, Y. Tangiguchi, G. Suzuki, *Structured Video Computing*, IEEE MultiMedia 1(3): 34-43 (1994).

[Tran et al. 00] D. A. Tran, K. A. Hua, and K. Vu, *VideoGraph: A Graphical Object-based Model for Representing and Querying Video Data*, 2000.

[Tran_Thuong et al. 02a] T. Tran-Thuong and C. Roisin , *A Multimedia Model Based on Structured Media and Sub-elements for Complex Multimedia Authoring and Presentation*, Special Issue on "Image and Video Coding and Indexing", Int'l Journal of Software Engineering and Knowledge Engineering, , 2002.

[Tran_Thuong et al. 02b] T. Tran_Thuong, C. Roisin, *An Abstract Animation Model for Integrating SMIL Basic Animation Elements with Multimedia Documents*, IEEE International Conference on Multimedia & Expo 2002 (ICME 2002), Lausanne, Switzerland, August 26-29, 2002.

[Tran_Thuong et al. 02c] T. Tran_Thuong, C. Roisin, *Edition of Media Content Description for Multimedia Document Composition*, the article submited to ACM Multimedia 2002.

[Tran_Thuong et al. 01] T. Tran_Thuong, C. Roisin, *Structured Media for Multimedia Document Authoring*, International Workshop on Web Document Analysis (WDA'2001), Seattle, Washington, USA, 8 Sept., 2001.

[van Rossum et al. 93] G. van Rossum, J. Jansen, K. Mullender, D. Bulterman, *CMIFed: a presentation Envi-ronment for Portable Hypermedia Documents*, Proc. of the ACM Multimedia Conference, California, 1993.

[Vasconcelos et al. 98] N. Vasconcelos and A. Lippman, *Bayesian Modeling of Video Editing and Structure: Semantic Features for Video Summarization and Browsing*, ICIP'98, Chicago, 1998.

[Vazirgiannis et al. 98] M. Vazirgiannis, Y.Theodoridis, T.K. Sellis. *Spatio-Temporal Composition and Indexing for Large Multimedia Applications.* In Multimedia Systems, 6(4): 284-298, 1998.

[Vetro et al. 01] A. Vetro, H. Sun and Y. Wang, *Object-based transcoding for adaptable video content delivery*, IEEE Transactions on Circuit and Systems for Video Technology, vol. 11, no. 3, pp. 387-401, Mars, 2001.

[Vilain et al. 86] M. Vilain and H.A. Kautz. *Constraint propagation algorithms for temporal reasoning*. In AAAI-86 Philadelphia, PA, pages 132 – 144, 1986.

[Villard et al. 00] L. Villard, C. Roisin and N. Layaïda, *A XMLbased multimedia document processing model for content adaptation*, Digital Documents and Electronic Publishing (DDEP00), September 2000.

[Villard 02] L. Villard, *Modèles de documents pour l'édition et l'adaptation de présentations multimédias*, Thèse de doctorat, INPG – Grenoble, Décembre 2000.

[Vodislav 95] D. Vodislav, *Visual Programming for Animation in User Interfaces*, (poster) Proceedings of the 11th IEEE Symposium on Visual Languages (VL'95), Darmstadt, Germany, September 1995.

[Wahl et al. 94] T. Wahl, K. Rothermel, *Representing Time in Multimedia-Systems*, IEEE Int. Conf. on Multimedia Computing and Systems, May 1994.

[Wang et al. 01] J. Wang, T.S. Chua, L. Chen, *Cinematic-Based Model for Scene Boundary Dectection*, 8th International Conference Multimedia Modeling (MMM01), Amsterdam, 5-7 November 2001.

[Weiss et al. 94] R. Weiss, A. Duda, and D.A.Gifford, *Content-based access to algebraic*, In IEEE int'l Conf. on Multimedia Computing and System, Boston, USA, June 1994.

[Weiss et al. 95] R.Weiss, A.Duda, D.A.Gifford, *Composition and Search with a Vidéo Algebra*, IEEE, 1995.

[Weitzman et al. 94] L. Weitzman and K. Wittenburg, *Automatic Presentation of Multimedia Documents Using Relational Grammars*, Proceedings of 2nd ACM Conference on Multimedia, ACM Press, pp. 443-451, San Francisco, California, October 1994.

[William et al. 89] William C. Mann, Christian M. I. M. Matthiesen, and Sandara A. Thompson. *Rhetorical Structure Theory and Text Analysis.* (technical report ISI/RR-89-242), November 1989.

[Zhang et al. 95] H.J. Zhang et al., *Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution*, ACM Multimedia 95 - Electronic Proceedings, San Francisco, California, November 5-9, 1995

[Zelenika 01] Z. Zelenika, *CARNet Media on Demand - Metadata model*, web edition – 21 mai 2001.

Standard

[ATA 00] *ATA spécification 2100*, Air Transport Association of America (ATA), 2000.

[Docbook 01] Oasis, *Docbook*, 2001. http://www.docbook.org/.

[DSSSL:ISO] International Organization for Standardization/International Electrotechnical Commission. Information technology --- Processing languages --- *Document Style Semantics and Specification Language (DSSSL)*. 1996 (Note: International Standard ISO/IEC 10179:1996).

[HyTime:ISO 97] ISO/IEC JTC1/SC18/WG8 N1920, *Information Technology: Hypermedia/Time-based Structuring language (HyTime)*, Second edition, ISO/IEC, August 1997.

[IMS] *IMS Metadata*, IMS Global Learning Consortium, Inc., http://www.imsglobal.org/metadata/index.cfm

[MPEG-4] R. Rob Koenen, *MPEG-4 Overview* - (V.21 – Jeju Version), ISO/IEC JTC1/SC29/WG11 N4668, March 2002.

[MPEG-7] J. M. Martínez, *MPEG-7 Overview* (version 8), ISO/IEC JTC1/SC29/WG11, Klangenfurt, July 2002.

[MPEG-21] J. Bormans, K. Hill, *MPEG-21 Overview* (v.5), /IEC JTC1/SC29/WG11/N5231, Shanghai, October 2002.

[RDF 99] O. Lassila, R. R. Swick, *Resource Description Framework (RDF) Model and Syntax Specification*, W3C Recommendation, 22 Février 1999.

[SMIL2.0 01] *Synchronized Multimedia Integration Language (SMIL 2.0)*, W3C Recommendation 07 August 2001.

[SVG1.1 03] W3C, *Scalable Vector Graphics* (SVG) 1.1 Specification, W3C Recommendation 14 January 2003.

[TEI 01], *"The TEI Guidelines"*, Text Encoding Initiative, http://www.tei-c.org.

[VRA] VRA - Visual Resources Association, http://www.vraweb.org/.

[XHTML 00] S. Pemberton et al., *XHTML 1.0: The Extensible HyperText Markup Language*, 2000, http://www.w3.org/TR/xhtml1.

[XSLT:W3C] James Clark. *XSL Transformations (XSLT)* Version 1.0. 16 November 1999 W3C Recommendations are available at http://www.w3.org/TR/. http://www.w3.org/TR/xslt.

[W3C SW01] *W3C Semantic Web Activity*: www.w3c.org/2001/sw/.

Applications

[ASP 01] Microsoft ASP, http://www.microsoft.com, 2001.

[GR*i*NS] GR*i*NS Pro Editor for SMIL 2.0 by Oratrix, *http*://www.oratrix.com/Products/G2E.

[Image-Indexer] Image-Indexer™, LTU Technologies commercialise des logiciels innovants d'analyse atomatique des images et des vidéos, http://www.ltutech.fr/Image-Indexer.htm.

[JSP 01] JSP, http://java.sun.com/products/jsp, 2001.

[LimSee] LimSee : Un éditeur temporel pour les documents au format SMIL de projet Opéra/Wam, http://opera.inrialpes.fr/LimSee.html.

[Macromedia] Macromedia Director, http://www.macromedia.com/.

[Opera_Kaomi] Opéra, Kaomi, http://opera.inrialpes.fr/Kaomi.html.

[PHP 01] *PHP*, http://www.php.net, 2001.

[VRML] VRML - Virtual Reality Modeling Language, http://www.web3d.org/.

[Xalan 01] *Xalan*, http://xml.apache.org, 2001.

Projets

[AGIR] AGIR- Architecture Globale pour l'Indexation et la Recherche par le contenu de données multimédia, http://www.telecom.gouv.fr/rnrt/pagir.htm.

[DICEMAN] DICEMAN - Distributed Internet Content Exchange with MPEG-7 & Agent Negotiations, http://www.teltec.dcu.ie/diceman/intro.html.

[ICTT] ICTT - Interaction Collaborative, Téléformation, Téléactivités, http://ictt.insa-lyon.fr/default.htm.

[MOVI] MOVI - MOdels for Vision, http://www.inrialpes.fr/movi/.

[PAROLE] Analysis, Perception and Automatic Recognition of Speech, http://www.loria.fr/equipes/parole/.

[VISTA] VISTA - VIsion Spatio-Temporelle et Active, http://www.irisa.fr/vista/.

Annexe : Les schemes standards de MPEG-7

La syntaxe XML du type *SegmentType* est comme suivant :

```
<complexType name="SegmentType" abstract="true">
        <complexContent>
            <extension base="mpeg7:DSType">
                <sequence>
                    <choice minOccurs="0">
                        <element name="MediaInformation" type="mpeg7:MediaInformationType"/>
                        <element name="MediaLocator" type="mpeg7:MediaLocatorType"/>
                    </choice>
                    <element name="StructuralUnit" type="mpeg7:ControlledTermUseType" minOccurs="0"/>
                    <element name="CreationInformation"
                                type="mpeg7:CreationInformationType" minOccurs="0"/>
                    <element name="UsageInformation" type="mpeg7:UsageInformationType" minOccurs="0"/>
                    <element name="TextAnnotation" minOccurs="0" maxOccurs="unbounded">
                        <complexType>
                            <complexContent>
                                <extension base="mpeg7:TextAnnotationType">
                                    <attribute name="type" use="optional">
                                        <simpleType>
                                            <union member Types="mpeg7:termReferenceType string"/>
                                        </simpleType>
                                    </attribute>
                                </extension>
                            </complexContent>
                        </complexType>
                    </element>
                    <element name="MatchingHint" type="mpeg7:MatchingHintType"
                                minOccurs="0" maxOccurs="unbounded"/>
                    <element name="PointOfView" type="mpeg7:PointOfViewType"
                                minOccurs="0" maxOccurs="unbounded"/>
                    <element name="Relation" type="mpeg7:RelationBaseType"
                                minOccurs="0" maxOccurs="unbounded"/>
                </sequ ence>
            </extension>
        </complexContent>
</complexType>
```