



République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Bejaia

Faculté des Sciences Exactes
Département d'Informatique

ECOLE DOCTORALE RESEAUX ET SYSTEMES DISTRIBUES

Mémoire de Magistère

En Informatique

Option : Réseaux et Systèmes Distribués

Thème

Modèles et Outils Documentaires pour les Hypervidéos sur le Web

Présenté par

Madjid SADALLAH

Devant le jury composé de :

Président	Abdelkamel TARI,	M.C.A	Université de Bejaïa
Rapporteur	Yannick PRIE	M.C. (HDR)	U. Lyon1, France
Examinateur	Amar BALLA	Professeur	ESI
Examinateur	Abdessamed-Reda GHOMARI	M.C.A	ESI
Invité	Azze-Eddine MAREDJ	Docteur	CERIST

Promotion : 2008/2009

Remerciements

Je tiens en premier lieu à exprimer ma profonde gratitude à Monsieur Yannick Prié, maître de conférences à l'université Claude Bernard Lyon 1 pour avoir accepté de m'encadrer dans ce travail, pour sa disponibilité et son accompagnement tout au long du parcours donnant naissance à ce mémoire, de ses innombrables idées et conseils sans lesquels je n'aurais certainement pas pu avancer.

Je remercie aussi Monsieur Azzeddine Maredj, chargé de recherche au CERIST et co-directeur de mémoire, de m'avoir orienté, corrigé mon travail, soutenu et encouragé, surtout pour sa disponibilité durant les moments les plus difficiles.

J'aimerais exprimer ma reconnaissance envers les membres du jury, Monsieur Kamel TARI, maître de conférences à l'université de Béjaia, Monsieur Amar BALLA, professeur à l'école supérieure d'informatique et Monsieur Abdessamed-Reda GHOMARI, maître de conférences à l'école supérieure d'informatique pour m'avoir fait l'honneur d'accepter de juger ce travail.

Je tiens particulièrement à remercier Monsieur Badache Nadjib, professeur à l'US-THB et Directeur du CERIST, de m'avoir donné la chance d'entrer dans le monde de la recherche.

Toute mon amitié va aussi à Monsieur Olivier Aubert que je souhaite remercier très chaleureusement pour sa disponibilité, ses orientations, son aide et son engagement à mes côtés durant toute la durée de ce mémoire et pour le temps qu'il m'a accordé pour discuter et débattre des concepts et des technologies.

Merci à tous les membres de l'équipe de recherche SMDS (Systèmes Multimédia et Documents Structurés), Mr Maredj, Noureddine et Lamia, tous mes collègues de la promotion ainsi que tous mes amis du CERIST dont je cite Mourad, Rafik et Rabah. Enfin, je remercie vivement tout le personnel du service formation du CERIST pour leur disponibilité infaillible.

Sans l'appui de mes proches et amis, ce travail n'aurait pas pu s'accomplir. Merci à mes parents, ma femme et mes deux enfants ainsi que mes frères et mes sœurs pour leur soutien et leur encouragement.

Tout ceci n'aurait pu avoir de sens sans ma merveilleuse épouse qui m'a soutenu sans réserve pendant tout ce temps, supportant les égarements de mes humeurs.

*À mes parents,
pour leur perpétuel soutien et leur amour incessant.*

*À ma tendre épouse Saliha,
pour sa patience et ses encouragements.*

*À nos deux anges Ilyas et Malek,
pour leurs petits cœurs aimant.*

À toute ma famille !

Table des matières

Introduction Générale	1
1 Introduction	1
2 Motivations et objectifs	2
3 Cadre du travail	3
4 Organisation du mémoire	4
Chapitre 1	
Hypervidéo : Concepts et Généralités	
1.1 De l'hypertexte à l'hypervidéo	5
1.1.1 Historique	5
1.1.2 Documents audiovisuels numériques	7
1.1.3 Vidéo	8
1.1.4 De la vidéo structurée à l'hypervidéo	9
1.2 Hypervidéo	10
1.2.1 Définition	10
1.2.2 Terminologie usuelle	10
1.3 Propriétés des hypervidéos	12
1.3.1 Différences importantes entre hypermédia et hypervidéo	12
1.3.2 Classification des hypervidéos	14
1.3.3 La désorientation	15
1.3.4 La surcharge cognitive	16
1.3.5 La rhétorique des hypervidéos	16
1.4 Hypervidéos et annotations	17
1.4.1 Structure et description des documents audiovisuels	17
1.4.2 Annotation	18
1.4.3 Annotation de la vidéo	18

1.4.4	Processus d'annotation	19
1.4.5	Techniques d'annotation	19
1.4.6	Définition de la structure vidéo	19
1.4.7	Hypervidéo comme vidéo interactive à base d'annotations	21
1.4.8	Annotations dans les hypervidéos	22
1.5	Conclusion	22

Chapitre 2	
Etat de l'Art	23

2.1	Modèles, Langages et Standards Multimédia	23
2.1.1	Modèles de référence de l'hypermédia	23
2.1.2	Modèles spatiaux et temporels du multimédia	25
2.1.3	Langages et Standards Multimédia	26
2.1.4	Discussion	29
2.2	Hypervidéos et Annotations	30
2.2.1	Standards d'annotation multimédia	30
2.2.2	Outils d'annotation vidéo	32
2.2.3	Discussion	36
2.3	Systèmes et Environnements Hypervidéos	36
2.3.1	Premiers systèmes hypermédias à forte composante vidéo	36
2.3.2	Systèmes d'édition et de présentation des hypervidéos	37
2.3.3	Comparaison entre les systèmes	44
2.3.4	Discussion	45
2.4	Hypervidéo sur le Web	46
2.4.1	Le multimédia sur le Web	46
2.4.2	Outils de diffusion multimédia sur le Web	47
2.4.3	Vidéo enrichie sur le Web	49
2.4.4	Hypervidéo et Web	50
2.4.5	Discussion	51
2.5	Conclusion	52

Chapitre 3	
CHM : Modèle de Documents Hypervidéos	53

3.1	Principes du modèle	53
3.1.1	Concept d'hypervidéo dans CHM	53

3.1.2	Les Annotation comme infrastructure de conception	54
3.1.3	Modélisation logique à base de composants	54
3.1.4	Adaptation des paradigmes de l'hypermédia	55
3.1.5	Dimensions multimédia de l'hypervidéo	55
3.1.6	Gestion dynamique à base d'événements	55
3.2	Modèle d'annotations	56
3.3	Modèle logique à base de composants	57
3.3.1	Composants de base (CHM Core Components)	58
3.3.2	Composants d'interface simples (CHM Plain Components)	61
3.3.3	Composants de haut niveau (CHM High Level Components)	64
3.3.4	Composants d'accès aux données (CHM Data Access Components) .	67
3.4	Dimension spatiale des hypervidéos CHM	68
3.5	Dimension temporelle des hypervidéos CHM	70
3.5.1	Paradigme TimeLine Reference (TLR)	70
3.5.2	Attributs temporels et de synchronisation	72
3.5.3	Synchronisation multimédia	72
3.5.4	Gestion de la dé-synchronisation	73
3.6	Dimension hypermédia et interactionnelle	73
3.6.1	Interactions dans les hypervidéos	73
3.6.2	Notion de lien	74
3.6.3	Ancres hypervidéos	75
3.7	Modèle à base d'événements	77
3.7.1	Événements	77
3.7.2	Traitements et actions sur le document hypervidéo	79
3.8	Discussion du modèle CHM	81
3.8.1	Caractéristiques du modèle	81
3.8.2	Relation de CHM aux modèles DHM/AHM	82
3.9	Conclusion	84

Chapitre 4

Hypervidéos CHM sur le Web : Architecture et Mise en Œuvre

85

4.1	Motivations pour l'hypervidéo sur le Web	85
4.2	Choix technologiques	86
4.3	Scénario : Intervention de Tim Bernes Lee lors de la conférence TED'09 .	88
4.4	Architecture du prototype WebCHM	90

4.5	Annotation comme pré-traitement	92
4.6	Édition et spécification des composants	95
4.6.1	Déclaration de composants CHM	95
4.6.2	Composants CHM du Scénario	97
4.7	Interprétation, transformation et génération de code	101
4.7.1	Moteur d'interprétation	101
4.7.2	Moteur de transformation	102
4.7.3	Accès aux données	103
4.7.4	Gestionnaire de composants	104
4.8	Formatage spatial, temporel et synchronisation	105
4.8.1	Formatage spatial et mise en style	105
4.8.2	Gestion du temps et de la synchronisation	106
4.9	Comportement dynamique des hypervidéos	110
4.9.1	Gestionnaire d'interactions	110
4.9.2	Gestionnaire événementiel	110
4.10	Interface de présentation	111
4.10.1	Templates de rendu	111
4.10.2	Lecteur vidéo sur Web <i>CHMVideoPlayer</i>	112
4.10.3	Exemples d'éléments de rendu hypervidéos	113
4.11	Discussion	116
4.12	Conclusion	117

Conclusion Générale	118
----------------------------	------------

1	Synthèse et rappel des objectifs	118
2	Bilan scientifique	119
2.1	Aspect <i>recherche</i>	119
2.2	Aspect <i>développement</i>	120
3	Perspectives	120
3.1	Aspect <i>Recherche</i>	121
3.2	Aspect <i>Développement</i>	122

Bibliographie	123
----------------------	------------

Table des figures

1.1	Structures hypervidéo	14
2.1	Architecture en couche du modèle de Dexter	24
2.2	Prototype HyperCafe	37
2.3	Prise d'écran du prototype Hyper-Film	38
2.4	Interface utilisateur du système VideoClix	39
2.5	Interface utilisateur du système Hyper-Hitchcock	40
2.6	Interface utilisateur du système HyperSoap	41
2.7	Interface utilisateur du système VisuaSHOCK Movie	41
2.8	Prise d'écran de la démonstration Hot Norman	42
2.9	Interface du prototype Advene	43
2.10	Modules de l'Environnement SIVA Suite	43
2.11	Interface riche de Youtube	50
2.12	Prototype Popcorn	50
3.1	Vue générale du modèle logique de CHM	57
3.2	Les composants de base de CHM	58
3.3	Composants d'interface simples de CHM	61
3.4	Composants CHM de haut niveau	64
3.5	Composants CHM d'accès aux données	69
3.6	Model spatial des hypervidéos CHM	69
3.7	Les événements et les actions dans une hypervidéo CHM	78
3.8	Hypervidéo <i>Story Of PI</i>	81
4.1	Représentation formelle du scénario	90
4.2	Processus général de création hypervidéo	91
4.3	Architecture générale du prototype <i>WebCHM</i>	92
4.4	Annotation de la vidéo par Advene	94
4.5	Lecteur CHM dans le contexte d'un export Web de Advene	113
4.6	WebCHM en action	116
A.1	Diagramme UML du modèle CineLab	126

Liste des tableaux

2.1	Comparaison des outils d'Annotation de la Vidéo	34
3.1	Éléments CHM de base : Hypervidéo et lecteurs d'annotation	59
3.2	Éléments CHM de base : les composants atomiques	59
3.3	Éléments CHM de base : les composants temporalisés et de temporalisation	61
3.4	Composants de Haut Niveau : Historique et Trace	65
3.5	Composants de Haut Niveau : Index de navigation	66
3.6	Composants de haut niveau : ligne de temps	67
3.7	Attributs spatiaux	69
3.8	Attributs temporels	72
3.9	Ancres et hotspots CHM	76
3.10	Événements et actions de l'hypervidéo	77
3.11	CHM exprimé par le modèle de Dexter	84
4.1	Exemples d'événements <i>WebCHM</i>	96
4.2	Valeurs possibles de l'attribut <i>timeAction</i>	108

Introduction Générale

1 Introduction

La mutation de l'univers informationnel induite par la vulgarisation des nouvelles technologies de l'information et de la communication a provoqué une reconfiguration des pratiques intellectuelles humaines faisant émerger de nouvelles conceptions et formes de perception de la connaissance. La numérisation de la vidéo et l'évolution récente et constante des systèmes permettant de la manipuler ont coïncidé avec l'apparition du Web comme canal de transmission et son arsenal d'outils de diffusion multimédia. Les réseaux sociaux et les sites de partage tels que Youtube¹, ont fait qu'aujourd'hui l'échange de données audiovisuelles devienne un phénomène social envahissant avec des ramifications qui atteignent un large et croissant éventail de consommateurs numériques.

La vidéo interactive est une forme récente et très attractive de la vidéo numérique qui présente des caractéristiques *informationnelles* et *interactionnelles* nouvelles. Les possibilités offertes à l'utilisateur d'influencer l'expérience audiovisuelle à priori *filmique* résultent en le développement de nouvelles formes d'expérience utilisateur² qui s'étendent au delà de celles répandues actuellement par exemple dans les forums vidéos, chat et mail vidéos et les videoblogs, vers des utilisations plus avancées comme l'apprentissage et la publicité interactive.

Les systèmes hypermédias conventionnels qui ont su s'ouvrir sur le multimédia et intégrer la vidéo - bien que ça soit d'une manière souvent rudimentaire - sont de nouveau confrontés à d'autres besoins issus de l'émergence de la vidéo interactive. Une variété d'utilisations inédites apparaît dans le but de changer les manières et les moyens selon lesquels l'information multimédia, principalement audiovisuelle, est créée, reçue et perçue.

1. <http://www.youtube.com>

2. L'expérience utilisateur (UX : *User Experience*) est une notion qui qualifie la perception engendrée chez un utilisateur en dialogue avec l'interface d'un environnement. Axée sur des mesures subjectives et émotionnelles et d'après l'encyclopédie *Wikipédia*, elle n'est réussie que si elle est utile (la présentation de l'interface est intéressante et utile pour l'utilisateur), accessible, capable de mettre l'utilisateur en confiance avec son système et apte à assurer de la valeur ajoutée grâce aux interfaces et aux interactions).

Dans le cadre de ce mémoire, nous nous intéressons à l'hypervidéo, un type particulier de documents hypermédias, qui combine la richesse de l'audiovisuel avec les capacités du multimédia et l'interactivité de l'hypermédia. Ayant la vidéo comme centre de gravité sémantique, spatial et temporel, elle définit un cadre intégrateur permettant de tirer profit des nouvelles formes d'interactivité et d'acquisition de l'information qu'offre la vidéo interactive en la combinant dans des espaces informationnels, enrichie par les capacités avérées des documents multimédia/hypermédias. L'intégration que propose l'hypervidéo a pour objectif de permettre un meilleur contrôle tout en garantissant plus de liberté dans l'activité d'acquisition de l'information par la composition de structures de transmission de connaissance plus élaborées, riches et plus flexibles.

2 Motivations et objectifs

L'hypervidéo permet d'envisager de nouvelles formes de création et de présentation de l'information audiovisuelle traitée dans sa granularité la plus fine ou la plus complexe et contrainte en espace et en temps. L'expérience *filmique* combinée à celle hypertextuelle suggère des modèles de représentation qui dépassent ceux - très rigides et restrictifs - définis par les modèles hypermédias existant. En effet, comme le souligne Chambel et al. [19], les formes illustratives suivant lesquelles la vidéo est aujourd'hui traitée par les systèmes hypermédias ne peuvent appréhender la distribution spatiale et temporelle de l'information résultant du caractère continu de la vidéo. À cela s'ajoutent les besoins de représentation formelle dédiée et de modalités de présentation nécessaires. Par conséquent, la prise en charge réelle des hypervidéos ne peut être actuellement que très partielle.

Bien que certains concepts propres aux hypervidéos soient discutés depuis la fin des années 1990s [82], leur application effective n'est pas encore largement répandue [27]. Peu de systèmes ont proposé de conforter les nouvelles pratiques audiovisuelles par une théorisation qui est au demeurant primordiale pour mieux les appréhender. Au delà des défis technologiques que la vidéo - sous toutes ses formes - soulève, le manque de formalisation a grandement entravé les hypervidéos d'être examinées et utilisées à une plus grande échelle. Les environnements permettant d'interagir avec la vidéo en mettent œuvre des solutions techniques étroitement liées à leurs systèmes hôtes et leurs implémentations et environnements cibles, pour exhiber des utilisations *exotiques* de la vidéo mais pas des concepts. La conceptualisation des hypervidéos n'a pas été largement examinée et à notre connaissance, au delà des modèles généraux de l'hypermédia qui ont montré leurs limites, aucun modèle assez générique et peu restrictif n'est venu mettre les bases théoriques permettant l'émergence de l'hypervidéo comme domaine et comme base d'applications.

Partant de ces constats, nous proposons dans le cadre de ce mémoire d'aborder la définition formelle de l'hypervidéo dans un cadre applicatif qui exige de confronter les apports théoriques avec la réalité applicative. Cela nous amène à chercher un équilibre entre théorie et application en se fixant deux types de résultats :

Proposition d'un modèle de document hypervidéo : Pour répondre aux exigences de théorisation des hypervidéos, nous proposons un modèle générique à base de composants qui repose sur le concept d'annotation pour ajouter des capacités à la vidéo et pour l'intégrer dans des espaces hypermédias hétérogènes. Cette notion d'annotation permet de mettre en œuvre une passerelle qui assurera l'intégration des documents audiovisuels et des hypermédias dans un nouveau type de document multimédia présentant pleinement la robustesse du hypermédia et intégrant correctement l'audiovisuel interactif.

Validation du modèle proposé dans un système hypermédia réel : La validation de ce modèle est envisagée dans le cadre des technologies multimédia pour le Web. Cette mise en œuvre est l'occasion d'aborder les limites du support qu'offre le Web au multimédia en général et à la vidéo en particulier. Les concepts des hypervidéos sont donc mis en œuvre pour améliorer ce support afin d'assurer une meilleure intégration de la vidéo et de rehausser les formes de son utilisation tout en tirant avantage des percées technologiques du Web actuel. La vidéo étant le média le plus riche tout en étant le plus récalcitrant de part ses dimensions logique, interactive, spatiale et temporelle et les relations complexes entre elles, nous pensons que la dialectique autour de ce média permettra d'envisager de nouvelles pratiques pour tout le multimédia sur le Web.

3 Cadre du travail

Le travail présenté est réalisé au CERIST (*Centre de Recherche sur l'Information Scientifique et Technique*) dans le cadre du projet MediaStudio2 de l'équipe *Systèmes Multimédia et Documents Structurés* (SMDS) en coopération avec l'équipe SILEX du LIRIS dans le cadre du projet Advene (*Annotate Digital Video, Exchange on the Net*) et dont l'objectif est de fournir des outils de lecture active et d'interprétation de documents audiovisuels.

4 Organisation du mémoire

Ce mémoire est organisé en deux parties. Les deux premiers chapitres introduisent le domaine des hypervidéos et présentent les travaux antérieurs associés, alors que le troisième et le quatrième chapitres développent notre proposition de modèle et sa mise en œuvre. Le mémoire s'achève sur une conclusion et un exposé des perspectives futures. Nous décrivons dans la suite le contenu de chaque chapitre.

Le premier chapitre retrace les évolutions ayant permis l'émergence de la vidéo interactives et évoque les principes des technologies hypertextuelles, hypermédias et multimédia. L'hypervidéo est introduite comme étant la convergence de tous ces domaines vers un espace d'information audiovisuel ; les différentes caractéristiques des hypervidéos sont alors présentées. L'hypervidéo supposant la description et la structuration de la vidéo, le besoin d'annoter les vidéos associées est mis en évidence tout en discutant de l'apport des annotations pour la conception des hypervidéos.

Le deuxième chapitre présente l'état de l'art relatif aux hypervidéos développé suivant plusieurs angles importants : le multimédia et ses dimensions, les hypermédias, les hypervidéos, les annotations et les technologies audiovisuelles sur le Web. Le chapitre conclut au besoin manifeste de modèles hypervidéos dédiés et à l'intérêt d'investir encore mieux dans les approches à base d'annotations pour avoir des réponses concrètes aux besoins d'étendre les représentations et les techniques des hypervidéos.

Le troisième chapitre détaille le modèle proposé pour la conceptualisation des documents hypervidéos. Justifiant la démarche générale et la stratégie de conception retenue, il introduit un formalisme logique à base de composants. La description de la structure et du contenu est réalisée par les annotations pouvant servir à diverses utilisations. Les différentes dimensions multimédia des hypervidéos sont par la suite modélisées par des représentations dédiées. Une évaluation du modèle conclut le chapitre.

Le quatrième chapitre propose une implémentation du modèle dans le but de valider ses concepts et son formalisme. Pour cela, les concepts avancés sont confrontés aux exigences de l'hypermédia le plus répandu, le Web. La mise en œuvre du modèle permet de proposer des mécanismes pouvant améliorer la prise en charge de la vidéo et du multimédia en général. Le développement de l'architecture de l'implémentation est accompagné d'un exemple concret illustrant l'utilisation du prototype développé.

La conclusion générale résume les points essentiels du travail présenté et ouvre la voie à d'autres perspectives de recherche et de développement pouvant être envisagées.

Chapitre 1

Hypervidéo : Concepts et Généralités

L’OBJECTIF de ce chapitre est de retracer l’évolution des concepts hypertextuels et audiovisuels qui ont permis l’émergence de l’hypervidéo en tant que forme et modèle de documents hypermédias à forte composant audiovisuelle. Ceci nous permet de situer et de définir les hypervidéos et de présenter leurs caractéristiques et implications. Par la suite, les concepts d’annotation et de modèle à base d’annotations sont introduits et discutés dans le contexte des documents hypervidéos.

1.1 De l’hypertexte à l’hypervidéo

L’hypervidéo est le résultat de l’intersection entre le développement parallèle des technologies de l’hypertexte/hypermédia avec celles de la vidéo numérique. Étant centrée sur la vidéo, l’hypervidéo hérite la *dynamicité* du contenu et dépend à priori de la linéarité stricte du flux présenté. D’autre part, étant hypermédia, elle se libère de cette linéarité pour être plus réactive voire interactive. Ainsi, l’hypervidéo résulte d’une combinaison forte et à granularité très fine de la vidéo linéaire avec l’hypermédia.

1.1.1 Historique

Traditionnellement, le texte est structuré d’une manière séquentielle définissant implicitement un ordre de lecture figé appelé linéaire. Cette linéarité a fini par être détruite grâce aux documents numériques. L’avènement et la vulgarisation de ces derniers ont largement contribué à l’émergence de nouvelles formes de création et d’échange de l’information. De nouveaux modèles de représentation ont ainsi émergé, essayant de mieux appréhender les relations complexes entre idées, relations qui sont souvent mieux représentées suivant des structures en réseau plutôt qu’en linéaire.

1.1.1.1 Hypertexte

L'hypertexte a été pensé comme moyen d'interconnecter différents nœuds - documents ou parties textuelles de documents - [20] ; le nœud de départ est appelé ancre source, celui d'arrivée est appelé ancre destination, reliés par un mécanisme de branchement ou *navigation*. La séquence effective du texte parcouru est déterminée par les choix de l'utilisateur au moment de la lecture. Alors qu'en général l'auteur a un contrôle total sur l'ordre de lecture, dans l'hypertexte il ne peut qu'émettre des suggestions, via des liens, pour des séquences possibles [55]. L'hypertexte libère donc l'auteur de l'obligation de créer du texte séquentiel [66] et du fait, la réorganisation et la restructuration du document suivant des considérations personnelles et intellectuelles propres au lecteur deviennent une activité courante [62] permettant à ce dernier de mieux percevoir le contenu présenté et allant parfois jusqu'à se l'approprier.

1.1.1.2 Multimédia

Un document multimédia est défini dans [44] comme étant un ensemble d'objets de différents médias (texte, image, vidéo, audio, etc.) qui sont organisés spatialement et temporellement et sur lesquelles une structure de navigation peut être définie. C'est donc l'agencement interactif et hétérogène, dans le temps et dans l'espace, de données issues de plusieurs types de média.

Le besoin de composition des documents multimédia a fait l'objet de nombreux travaux qui ont abouti à l'identification de quatre dimensions usuelles pour la structuration multimédia : logique, temporelle, spatiale et hypermédia. La dimension logique correspond à l'organisation sémantique du document [44] par le biais de relations de haut niveau conceptuel, compréhensibles uniquement par l'être humain. La structure spatiale, quant à elle, décrit les contraintes d'organisation des différentes parties d'un document sur le support de présentation. L'enchaînement des éléments du document dans le temps est décrit par la dimension temporelle qui permet d'organiser et de synchroniser les diverses données *temporalisées* afin d'exprimer leurs liens dans le temps et la façon dont chaque donnée réagit à l'écoulement de ce temps. Enfin, la dimension hypermédia décrit les possibilités de navigation à travers le document et les formes d'interactivité permises. Elle regroupe des liens hypermédias qui font partie de la présentation multimédia, pouvant être intra ou extra document.

1.1.1.3 Hypermédia

L'introduction des systèmes hypertextuels a permis de s'affranchir de la linéarité figée de la lecture textuelle traditionnelle. Les évolutions technologiques ont fait que le mode textuel pour les documents numériques et les applications informatiques a fini par ne plus régner en maître absolu. La prise en charge puis l'adjonction d'images, de séquences vidéo et du son ont fait inventer le concept du média, puis de document multimédia pour enfin, recréer l'hypertexte en ayant en base des documents et des parties de documents multimédia, ce qu'on connaît par *hypermédia*.

L'hypermédia est une extension logique des concepts de l'hypertexte applicables uniquement sur du texte pour inclure tout type de données multimédia. De nouveaux paradigmes et des aspects étroitement liés à la nature de ses composants ont été créés. S'ajoutant aux dimensions classiques - logique (organisationnelle), spatiale et hypertextuelle (les liens entre parties) - présentes dans tout document hypertexte, la dimension temporelle redéfinit complètement la façon de considérer le document et son contenu.

1.1.2 Documents audiovisuels numériques

L'émergence de l'hypervidéo n'est rendue possible que grâce à l'évolution des techniques de transmission de l'information audiovisuelle. Cette dernière s'est faite progressivement engendrant différents modes d'exploitation de la vidéo.

Un document audiovisuel est défini dans [10] comme étant « *un document composé d'images animées et/ou de sons se déroulant de manière linéaire selon un rythme temporel particulier le rendant intelligible pour un lecteur* ». Cette définition suggère que, contrairement aux documents textuels, une vision globale et une perception complète de la connaissance transportée dans les documents audiovisuels ne peuvent être fournies à un instant donné. Autrement dit, les informations contenues dans les documents audiovisuels sont situées spatialement et temporellement, ce qui rend difficile leur analyse. Cela est d'autant plus compliqué que la cadence de lecture de document est imposée et aucun moyen ne permet d'accéder directement à un contenu, sauf à le repérer et à noter explicitement sa position dans le temps pour être capable de s'y reporter directement. En effet, il faut une heure pour visualiser une vidéo d'une heure, et si l'information recherchée est à la 20^{eme} minute, alors il faut passer les 19 premières pour la retrouver [94].

Le document audiovisuel diffère du document multimédia qui « mêle différents médias » tout en étant « construit à partir de différents sous-documents unis par un réseau de liens » [10]. En effet, la dimension hypermédia (les liens) et par conséquent la notion d'interactivité sont inexistantes dans l'audiovisuel. Par conséquent un document audiovi-

suel est un *document multimédia appauvri* qui ne devient pleinement multimédia que si on lui ajoute des possibilités d'interaction [72].

1.1.3 Vidéo

La vidéo est un type de document audiovisuel formé d'une séquence d'images appelée *frames* et d'un bande sonore synchronisée et arrangée dans le temps et dans l'espace d'une manière à avoir une cohérence dans le sens et dans la structure. De point de vue forme (physique), la vidéo est une combinaison de sous-médias - *pistes* - organisés suivant un axe de temps. Chaque piste est présentée sous la forme d'un flux d'éléments. La superposition des flux et leur mise en relation impliquent une structure physique qui décrit le regroupement d'images et la synchronisation du son - plus ou moins complexe - , et une structure logique - non explicitée - qui recouvre la notion du contenu de document, un contenu désignant toutes les informations formant une vidéo (image, audio, texte).

Vidéo Linéaire : en général, les programmes disponibles sur des bandes vidéo sont linéaires et leur contenu est considéré comme une seule unité informationnelle sur laquelle la lecture progresse d'une manière séquentielle. Les techniques de la vidéo linéaire sont destinées à montrer le flux d'information dans un contexte où les programmes ont une durée connue et où ils sont supposés être lus du début à la fin. La vidéo numérique linéaire ne présente pratiquement aucun moyen d'interaction au-delà de ceux conventionnels offerts par les lecteurs (lecture, pause, volume, etc.), son mode de perception impliquant un rôle du lecteur beaucoup plus passif. L'exploration du contenu résultante est très restrictive et inefficace pour une lecture plus intéressée, plus active. Ces conditions, normalement absente dans la vidéo interactive, sont encore moins probables dans l'hypervidéo [52].

Vidéo Interactive : la vidéo interactive est une forme numérique enrichie de la vidéo linéaire, reflétant l'idée d'une vidéo qui présente des formes inhabituelles d'interaction avec le lecteur [34], en lui assurant une lecture active à travers les possibilités de navigation. Elle permet par exemple de sélectionner des détails de flux pour aller vers d'autres détails, de telle manière que la ligne de temps du document n'impose plus son chemin narratif et l'utilisateur devienne libre d'explorer des segments suivant ses propres choix.

Le concept de *vidéo interactive* est associé à la définition de Michael K. Stenzler et al., stipulant qu'une application vidéo est interactive si l'utilisateur peut affecter le flux de la vidéo et que cette influence, au retour, affecte les choix futurs de l'utilisateur [91].

L'ajout de l'interactivité à la vidéo nécessite des phases de transformation et de structuration. Les deux formes de la vidéo - la forme linéaire et celle interactive - ont en

substance le même contenu et diffèrent sur une seule donnée : la structure. Alors qu'elle est implicite avec la vidéo linéaire, la vidéo interactive est munie d'une structure explicite où des objets sont identifiés et désignés et des liens sont définis. L'intégration de la structure peut être réalisée au niveau du flux physique et dans ce cas, l'application chargée du rendu doit comprendre le nouveau codage. L'intégration peut aussi être réalisée par l'association de la structure à la vidéo, ce qui requiert de l'application hôte sa prise en compte lors du rendu.

Grâce à la vidéo interactive, le document audiovisuel possède une infrastructure pour la génération de représentations faisant de lui un document multimédia à part entière. La forme d'interactivité qu'il présente peut être facilement combinée avec les technologies de l'hypermédia pour générer un tout nouveau type de document avec triple facette : audiovisuel, multimédia et hypermédia. Ce type de document - sur lequel nous focalisons notre étude - est communément appelé *hypervidéo*.

1.1.4 De la vidéo structurée à l'hypervidéo

L'adjonction d'une structure explicite à la vidéo permet d'analyser cette dernière suivant différents points de vue : (1) *logique*, correspondant aux relations entre les différents éléments et segments définis par la structure ; un segment correspond à l'intervalle qu'une entrée de la structure définit ; (2) *spatial*, se référant à la description des emplacements spatiaux des différents objets repérés et décrits ; (3) *temporel*, correspondant à la description des attributs temporels des éléments identifiés et (4) *hypermédia*, décrivant la définition de relations permettant de lier des éléments identifiés et décrits par la structure et permettant de définir des liens entre ces éléments. Ces mêmes dimensions peuvent être retrouvées dans le multimédia. Un document audiovisuel auquel on associe de la structure permet donc une analyse et une représentation multimédia de son contenu. L'ajout d'un comportement hypermédia à l'audiovisuel est au cœur de la technologie hypervidéo qui suppose non seulement le traitement multimédia/hypermédia du document audiovisuel mais l'étend pour relier les objets audiovisuels à tout son contexte.

La vidéo *hyperliée*, vidéo *cliquable* ou *hypervidéo* reflète donc l'idée d'une véritable intégration de la vidéo dans l'espace hypermédia [61], où elle n'est pas considérée comme simple illustration, mais peut aussi être structurée par des liens définis suivant des dimensions spatiales et temporelles [15, 16], permettant à l'utilisateur d'interagir avec son contenu et de naviguer au sein de sa structure. Des liens relient des noeuds audiovisuels à divers éléments multimédia (texte, images, graphiques, audios, animations, vidéos) et ressources disponibles dans l'espace hypermédia[102].

1.2 Hypervidéo

1.2.1 Définition

Le terme hypervidéo a été introduit pour la première fois lors de la présentation du système *HyperCafe* [82] et sa nouvelle façon de structurer et de présenter dynamiquement une combinaison entre des flux vidéos et d'autres données. Selon ses auteurs, l'hypervidéo dénote une combinaison entre la technologie vidéo et celle d'hypertexte, offrant ainsi à l'auteur aussi bien qu'au lecteur du document la richesse de récits multiples (*multiple narratives*) ainsi que différents moyens pour la structuration narrative et l'interactivité.

L'idée fondamentale des hypervidéos est que, au lieu d'une visualisation de la vidéo suivant une structure linéaire et rigide, le lecteur doit pouvoir interagir avec cette dernière. Il doit pouvoir par exemple changer le récit en cours pour avoir un supplément d'information, revenir en arrière ou aller vers un autre nœud de l'espace hypermédia ou vers un autre instant d'une autre vidéo et naviguer à travers les segments audiovisuels suivant ses intérêts et ce, durant toute la présentation [101]. Par conséquent, l'hypervidéo peut être représentée par un graphe dont certains de ses noeuds sont des fragments audiovisuels [3].

La structure hypervidéo est constituée de deux composantes complémentaires : le flux audiovisuel continu et les données associées. Ces dernières enrichissent le flux et permettent d'interagir avec lui et de le contrôler [3], déterminant ainsi plusieurs alternatives pour voir le même contenu. L'hypervidéo offre une appréhension différente des médias du document, en fournissant un moyen de s'affranchir de leur temporalité linéaire [2]. Chaque hypervidéo générée d'un document audiovisuel est considérée comme une vue possible du document mettant en relation la structure de données et le flux associé [3].

1.2.2 Terminologie usuelle

La recherche dans le domaine d'hypervidéo, contrairement à celle dans les cas généraux de l'hypertexte et de l'hypermédia, est encore en voie de maturation. Aucune standardisation des concepts et aucun consensus sur la terminologie n'existent encore. La terminologie qui est souvent associée à ce type documents peut être résumée en un ensemble de concepts :

Structure hypervidéo : cette structure contient toutes les informations de l'espace hypermédia : nœuds, liens et ancrés hypervidéos.

Document hypervidéo : composé du contenu et de la structure de l'hypervidéo, un document hypervidéo est généralement appelé simplement hypervidéo.

Système hypervidéo : un tel système permet de construire, modifier et visualiser des hypervidéos à des degrés variables. Sa fonction principale est relative à l'édition et peut ne pas offrir un moyen de modifier ou de visualiser l'hypervidéo.

Espace d'information : l'espace d'information d'un document hypervidéo inclut tout le contenu vidéo et le contenu d'information utilisé pour augmenter la vidéo et définir un hypermédia.

Nœud hypervidéo : un nœud représente une unité d'information autonome. On peut distinguer deux types de nœud :

- *Nœud Vidéo* : un tel nœud fait référence à une vidéo ou à des unités d'information de la vidéo (objet, image - *frame* -, scène, séquence, etc.).
- *Nœud d'Information* : un tel nœud représente principalement les unités d'information discrètes, y compris la vidéo dans son intégralité.

Lien hypervidéo : un tel lien doit mettre en relation au moins un nœud vidéo comme extrémité d'un lien.

Ancre hypervidéo : une ancre hypervidéo est le point de rencontre d'un nœud hypervidéo avec un lien vidéo. Une telle ancre a comme paramètre son emplacement sur la vidéo, un temps de début et un temps de fin correspondant à sa durée de vie.

Zone sensitive, région sensitive ou hotspot : une région sensitive désigne la représentation graphique d'une ancre spatio-temporelle au sein du nœud vidéo, utilisée pour la définition de liens contraints temporellement et spatialement. Ces termes sont parfois synonymes d'ancres hypervidéos.

Enrichissement : un enrichissement représente tout contenu visuel ou audible défini dans l'espace d'information et/ou représenté sur l'espace d'affichage dans le but d'accompagner le flux audiovisuel et ajouter de l'information au message véhiculé.

Annotation : Une annotation est une information associée à la vidéo ou à une partie de la vidéo sous forme textuelle ou autre.

Séquence narrative : C'est un chemin possible à travers un ensemble de nœuds (vidéo ou d'information) liés et assemblés de manière dynamique selon l'interaction utilisateur.

Liens temporels : un tel lien représente une référence basée sur le temps entre deux ancrages, où un instant précis de l'ancre source peut servir à déclencher la présentation de l'ancre de destination.

Liens spatio-temporels : un tel lien représente une référence entre différentes scènes vidéo, où un emplacement spécifié sur la vidéo source déclenche la lecture d'une vidéo destination à un instant précis.

1.3 Propriétés des hypervidéos

L'hypervidéo suit l'idée de base de l'hypertexte remplaçant le texte par un médium riche et dynamique qui est la vidéo. L'importance cruciale de l'audiovisuel infère à l'hypervidéo des caractéristiques qui ne sont pas propres aux documents hypermédias. D'autre part, les hypervidéos, en tant que cas spécifique des hypermédias, développent les particularités de ceux-ci et leurs problèmes majeurs tels que la désorientation et la surcharge cognitive, discutés dans [74]. En plus, la composante audiovisuelle induit d'autres propriétés et défis aux documents hypervidéos.

1.3.1 Différences importantes entre hypermédia et hypervidéo

L'hypervidéo se distingue de l'hypermédia par l'importance cruciale que revêt la vidéo elle-même et qui forme le *backbone* du document [102]. Les vidéos, en parallèles avec d'autres éléments additionnels sont entrelacés de manière à ce que le flux vidéo peut être exploré interactivement et navigué suivant un ordre délinéarisé [90].

Définition des concepts : la façon originale de considérer la vidéo remet en cause la définition préétablie de concepts tels que les nœuds et les liens qui ont été grandement influencés par la façon suivant laquelle l'hypermédia considère les médias. Ces concepts, dans le cadre des hypervidéos, ont du fait d'autres définitions et introduisent d'autres implications sur le concepteur aussi bien que le lecteur du document. Par exemple, la notion de nœud vidéo dans les hypermédias conventionnels est prise dans un sens plutôt rudimentaire considérant le média comme entité atomique sans moyens de définir un adressage fort à partir et vers des *détails vidéos* (scène, objet spécifique de la vidéo, etc.).

Modes de lecture : alors qu'à la présentation d'un document hypermédia, l'utilisateur se retrouve dans une activité de lecture hypertextuelle rigide (lecture classique presque papier [102]), avec l'hypervidéo et son orientation manifestement audiovisuelle, l'expérience utilisateur s'apprête beaucoup plus à la lecture *filmique* d'un document augmenté avec des liens spéciaux, du texte, des graphiques, etc. [32] et par des artefacts de présentation et des données multimédia subordonnant et complétant le flux vidéo principal.

Structure complexe : la structure de l'hypervidéo suit une logique centrée sur le contenu sémantique de la vidéo, qui est combinée avec d'autres structures d'information permettant à l'utilisateur de sélectionner des chemins narratifs selon le contenu de la vidéo et ses propres intérêts. Cette structure implique la segmentation et la structuration de

la vidéo et l'ajout voire le codage de données nécessaires pour la génération du contenu multimédia et interactionnel de l'hypervidéo.

Cadence de présentation pilotée par la vidéo : le texte et les graphiques sont des objets statiques dont le message est présent en tout temps à l'écran et le rythme d'acquisition de l'information est réglé manuellement par l'utilisateur. Par contre, le message de la vidéo et le rythme d'acquisition de l'information dépendent du mouvement des images et du son accompagnant au fil du temps. La cadence de présentation des hypervidéos est donc dictée par celle du flux audiovisuel en l'absence de toute interaction.

Temporalité forte de l'hypervidéo : l'hypervidéo implique la prise en compte effective de la temporalité de la vidéo qui devient la dimension la plus importante du document. Bien que l'hypermédia intègre du contenu dynamique, la notion du temps des éléments audiovisuels est plutôt locale aux médias inclus et la temporalisation globale du document n'en dépend pas. Dans le cas des hypervidéos, tout le contenu de la présentation est contrôlé dans le temps en rapport avec la ligne de temps de la vidéo.

Concept de nœud : en raison de la nature statique du texte et des graphiques, il est plus facile d'encapsuler des idées ou des concepts dans des nœuds. En vidéo par contre, il est plus difficile de cerner le concept de nœud à des segments de vidéo qui peuvent n'être qu'un une seule scène voire une seule image.

Dimensions spatiale et temporelle des liens : une particularité cruciale des hypervidéos réside dans la définition des liens. Les systèmes hypermédias classiques sont conçus avec des nœuds définis par des média discrets. Même les liens les plus sophistiqués pointant à partir des média continus ou vers des instants précis d'un média continu sont eux-mêmes indépendants du temps. Les ancrages hypervidéos et les régions sensitives, se référant à des détails précis au sein d'une séquence vidéo, ont leur propre dynamicité en espace et en temps, pouvant même se changer d'endroit par instants.

La structure hypervidéo offre des possibilités de sélection de certaines unités d'information par l'utilisateur grâce aux mécanismes d'interaction mis en œuvre. Les régions cliquables, appelées *hotspots*, peuvent être reliées au flux vidéo par l'espace et aussi par les instants de leur activation.

1.3.2 Classification des hypervidéos

Un document hypervidéo est organisé suivant différents niveaux et dépend du procédé suivant lequel la vidéo est intégré dans l'hypermédia existant [102]. Suivant les types des médias qui sont impliqués dans l'hypervidéo, on peut distinguer entre :

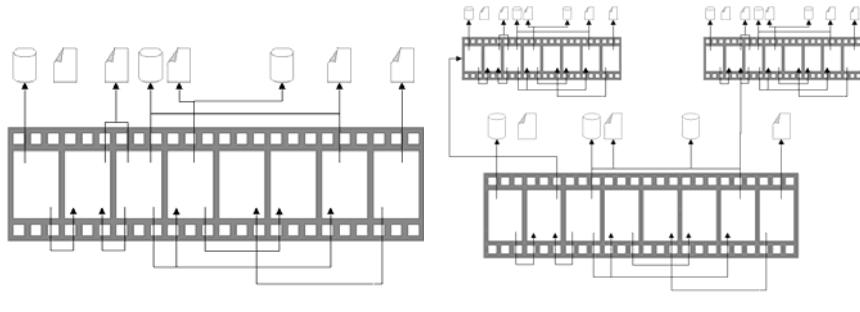


Figure 1.1 – Structures hypervidéo

- *Hypervidéo homogène (Structure fermée)* : la vidéo est le seul média impliqué, comprenant de l'information audiovisuelle présentée comme un flux continu d'images en mouvement qui peut être parcouru par l'utilisateur. Les liens définis sont du fait interne comme montré sur la figure 1.1(a) ;
- *Hypervidéo hétérogène (Structure ouverte)* : l'hypervidéo intègre d'autres médias, donnant d'autres informations relatives à la vidéo, l'illustrant et/ou la complétant. Pour cette plus large perspective, le nom de «video based hypermedia» ou «hyperlinked video»[16] est parfois adopté. Les liens peuvent être internes ou externe (voir la figure 1.1(b)).

1.3.2.1 Types de liens dans les hypervidéos

Les types de liens pouvant exister quand un objet de la vidéo est mis en relation comme origine et/ou comme destination sont explorés dans [82] et [16]. Quand la vidéo est à l'origine du lien, des conditions spatiales et temporelles peuvent être définies pour restreindre l'étendue du lien :

- *lien inconditionnel* : le lien est toujours présent quelle que soit la position temporelle dans la vidéo ;
- *lien spatial* : le lien est localisé spatialement sur la vidéo, mais indépendant de la position temporelle ;
- *lien temporel* : le lien est disponible pendant un certain intervalle de temps.

- *lien spatio-temporel* : le lien est valide pendant un certain temps, et disponible sur une région donnée de l'image.

Lorsque la vidéo est la destination, un intervalle de temps délimitant un segment peut être spécifié. Les deux bornes de l'intervalle sont facultatives, ayant le début et la fin de la vidéo pour valeurs par défaut.

1.3.2.2 Niveau de détail des ancrées hypervidéos

A partir des possibilités de segmentation d'une vidéo (voir §1.4.6.1), il est possible de distinguer trois niveaux de détail pour la définition de l'ancré d'un lien :

- *séquence vidéo* : une séquence vidéo peut être utilisée en tant qu'unité comme cible ou source d'un lien hypervidéo. Aucune connaissance de la structure interne de la séquence n'est nécessaire ;
- *scène vidéo* : une scène peut être référencée dans un lien hypervidéo mais cela suppose que l'utilisateur ait déjà connaissance et accès à la structure interne de les séquences de la vidéo ;
- *objet vidéo* : le niveau de détail pour une référence d'un lien hypervidéo peut aller jusqu'à sélectionner des objets d'une *frame* de la vidéo. Ainsi, il est possible de créer ou de pointer plusieurs liens sur la même scène. Des objets peuvent être en relation avec d'autres unités d'information de l'hypervidéo et ainsi, de fortes relations sémantiques peuvent être induites. Cependant, ce niveau de détail est trop fin et introduit une complexité importante du fait que le système hypervidéo doit considérer le mouvement de chaque objet à travers les différentes *frames* où il apparaît.

1.3.3 La désorientation

Correspondant au sentiment de perte dans l'espace documentaire (*lost in a hyperspace*) [20], la désorientation est un effet cognitif répandu dans les hypermédias. Elle est produite chez l'utilisateur qui, à un instant donné, devient incapable de placer le nœud courant sur la carte imaginaire qu'il construit de l'espace hypermédia pour avoir une vue d'ensemble, assez globale, du document [50]. Une forme de désorientation, dite conceptuelle survient lorsque l'utilisateur, en visitant un espace hypermédia, n'arrive pas à intégrer l'information sémantique visitée à l'ensemble des informations déjà acquises et ce, pour construire une représentation cohérente de l'information globale. L'information présentée au niveau d'un certain nœud est jugée inconsistante et incompréhensible par rapport à des connaissances antérieures.

Le caractère continu de l'audiovisuel dans les hypervidéos amplifie cette sensation de

ne plus pouvoir se situer dans l'hyperespace. En général, cela survient suite au manque d'information sur la structure de l'hyperespace, du moment que l'information est répartie aussi bien en espace que dans le temps.

Maintenir une vision d'ensemble du document et rendre disponible à tout point sa structure sont des éléments importants pour guider l'utilisateur et prévenir une éventuelle désorientation. Pour prévenir la désorientation conceptuelle, des descriptions sémantique des informations présentées sur chaque noeud peuvent être insérées afin d'éclaircir les relations entre ces noeuds et son rôle dans la construction de l'information globale.

1.3.4 La surcharge cognitive

La surcharge cognitive est un effet produit chez l'utilisateur qui est devant une grande quantité d'informations et qui s'efforce de trouver ce à quoi telle information doit être associée pour en tirer un sens et la mémoriser. Le passage vers une nouvelle zone d'information efface les connaissances acquises de la zone précédente. C'est pourquoi les traces qui permettent de rebrousser chemin et les cartes explicitant les réseaux de liens sont largement utilisées.

La surcharge dite *informative* reflète l'idée que l'effort de concentration nécessaire pour gérer plusieurs pistes de recherche diminue la capacité d'attention disponible pour la tâche principale. Plusieurs types d'informations pas toutes nécessaires s'imposent à l'utilisateur durant la navigation.

1.3.5 La rhétorique des hypervidéos

L'un des axes les plus importants dans l'exploration et la compréhension des mécanismes de l'hypervidéo est relatif à l'aspect rhétorique de la présentation [17, 82, 51] qui est très différent des documents classiques du fait de la temporalité et de la continuité de la vidéo. *Hypercafe* propose un modèle rhétorique propre aux hypervidéos et ce, en considérant que les nouvelles exigences rhétoriques et esthétiques de l'hypervidéo impliquent des changements dans des concepts traditionnels tels que les noeuds et les liens pour inclure la notion du temps dans l'ensemble de la présentation [82].

Matérialisation de liens : plusieurs approches pour la matérialisation des liens sont utilisées. Certains systèmes exploitent l'idée des notes de bas de page (*footnotes*) en définissant sur des fenêtres externes à l'interface vidéo des liens donnant accès à des informations supplémentaires et éventuellement externes [51]. D'autres systèmes définissent le concept d'*opportunité de lien* [82] (*spatiale*, *temporelle* ou *spatio-temporelle*) qui est munie d'un

intervalle pendant lequel une fenêtre est affichée sur une zone spécifique dont l'activation déclenche la lecture d'une cible donnée. Enfin, des systèmes plus sophistiqués [38] prennent en charge la notion de *hotspot* qui est une matérialisation de l'ancre source. Elle consiste à définir des régions sensitives dans chaque *frame* de la vidéo puis de relier ces régions à une destination commune. La définition de ces régions peut faire l'objet d'algorithmes de suivi des mouvements, l'intervention de l'utilisateur étant nécessaire uniquement à l'échec de ces derniers.

Notion de *Link Awareness* : l'hypervidéo implique l'ajout de l'interactivité à des objets spatio-temporels de la vidéo. La représentation des ancrages et des liens ainsi définis soulève des difficultés à lesquelles les techniques usuelles répondent très partiellement.

Permettre à l'utilisateur d'avoir connaissance des possibilités de navigation se réfère au problème de *link awareness*[15]. Il correspond aux mécanismes nécessaires pour représenter visuellement les liens présents à tout instant et faire en sorte que leur existence soit perçue correctement par l'utilisateur. La difficulté provient du fait que, d'une part les liens peuvent avoir une durée d'existence limitée correspondant à l'intervalle de temps de présentation de l'objet référencé, et d'autre part qu'il peut y avoir à un instant donné plusieurs liens actifs qui se chevauchent temporellement voire même spatialement. Toute solution à ce problème doit tenir compte de deux facteurs supplémentaires : d'abord, la matérialisation des liens ne devrait pas perturber le lecteur ni le désorienter ; puis, ayant la vidéo en progression avec le temps, une attention permanente de l'utilisateur est requise, des liens pouvant passer inaperçus.

1.4 Hypervidéos et annotations

1.4.1 Structure et description des documents audiovisuels

Une particularité bien propre à l'audiovisuel est que « la forme audiovisuelle ne mobilise pas d'unités répertoriées composant le contenu (comme les lettres de l'alphabet) » [5]. Bien que l'on puisse lui définir une certaine structure (découpage en scènes, plan, etc), aucun découpage ne peut être considéré comme modèle universel de définition d'unités élémentaires des documents audiovisuels.

Les modèles de représentation des documents audiovisuels sont d'une grande importance pour leur analyse. Au cœur de ces modèles se trouvent le concept d'*annotation* représentant diverses descriptions attachées aux différentes parties d'un document audiovisuel. Le besoin de recourir à la structuration et l'annotation vidéo comme moyen de la

décrire est à l'origine de la connivence forte que les hypervidéos entretiennent avec les annotations. L'idée de l'annotation vidéo peut être considérée comme la suite du développement des domaines de l'hypermédia et de l'hypertexte [24]. Les concepts de l'annotation hypermédia ont été adoptés dans le domaine de la vidéo résultant en ce qui est considéré par certains auteurs comme hypervidéo [24].

1.4.2 Annotation

Plusieurs définitions sont associées aux annotations. Nous retenons celle fournie dans [9] et qui définit l'annotation comme étant «*une note particulière attachée à une cible. La cible peut être une collection de documents, un document, un segment de document (paragraphe, groupe de mot, mot, image ou partie d'image, etc.), une autre annotation. A une annotation correspond un contenu, matérialisé par une inscription, qui est une trace de la représentation mentale que l'annotateur se fait de la cible.*»

A partir de cette définition, l'annotation, qui reflète une idée significative pour l'annotateur, peut être vue comme une inscription sur un document ou liée à celui-ci, décrivant, mettant en évidence ou ajoutant de l'information à une partie ou à la totalité du document [75]. Elle possède un lien implicite ou explicite avec ce document étant ancrée à une cible [25] mais prise séparément de cette dernière, elle peut avoir une existence propre et indépendante du document. Cependant, l'annotation ne conserve tout son sens que si elle est prise avec son ancre, lui redonnant ainsi son contexte documentaire [76].

Une annotation n'est pas forcément une métadonnée. En s'appuyant sur le processus de leur mise en place, la distinction mise en évidence dans [73] considère qu'une métadonnée est plutôt attachée à une ressource identifiée en tant que telle sur le Web, ayant plutôt une pertinence à priori et saisie suivant un schéma, alors qu'une annotation est plutôt située au sein de cette ressource et écrite au cours d'un processus d'annotation/lecture.

1.4.3 Annotation de la vidéo

L'annotation vidéo est le processus permettant d'associer des données textuelles ou multimédia à une vidéo ou à une partie de la vidéo, enrichissant le contenu sans altérer le flux original. Grâce à cette possibilité d'ajouter de l'information au contenu visuel et sonore, les annotations résultant peuvent être exploitées de diverses manières lors de la création et de la présentation des documents multimédia, induisant toutes sortes de possibilités de développement de ressources médiatiques. Par exemple, l'annotation rend possible la sélection du contenu, via la recherche ou le filtrage, permettant de capturer les éléments du contenu pour des visualisations, des traitements, de la navigation au sein de

ces contenus et d'accès aux éléments décrits. Les annotations peuvent servir à l'élaboration des programmes par profils et préférences, voire gérer les droits d'accès et de propriété intellectuelle [41].

1.4.4 Processus d'annotation

Le processus d'annotation peut être *automatique* ou *manuel*. L'annotation automatique procède par traitement informatique du signal vidéo ou sonore pour en extraire des unités d'information. Elle est effectuée par machine et peut être plus ou moins complexe selon les techniques et la sophistication des outils d'intelligence artificielle utilisés. L'annotation la plus simple à ce niveau consiste à extraire des informations sur les caractéristiques du média comme la durée intrinsèque.

L'annotation manuelle, quant à elle, exige une intervention humaine. Ce type d'intervention est nécessaire lorsque le traitement automatique ne peut suffire pour catégoriser des segments vidéo. L'annotation manuelle n'a pas de contraintes particulières sauf qu'elle peut demander du temps et de l'expertise de la part de la personne qui annotate.

1.4.5 Techniques d'annotation

Deux techniques d'annotations non exclusives et souvent entrelacées peuvent être distinguées [103] : l'association des métadonnées et l'enrichissement du contenu.

L'association de métadonnées utilise des modèles spécifiques pour la construction de structures sémantiques permettant des opérations diverses telles que la recherche dans le contenu. Cette approche requiert la compréhension du modèle d'annotation sémantique pour pouvoir associer aux média des métadonnées conformément au modèle.

La deuxième méthode, enrichissement de contenu, utilise d'autres éléments multimédia (images, graphiques, texte, audio, vidéo,...) pour enrichir les objets dans un flux audiovisuel générant un nouveau flux composé. L'utilisateur insère ses annotations interactivement en associant du texte, des images, des clips, etc., de sorte que le flux original soit enrichi par un nouveau contenu.

1.4.6 Définition de la structure vidéo

La connaissance contenue au sein de la vidéo ne pouvant être directement accessible, les informations décrivant ses données doivent être organisées d'une manière structurée offrant un moyen d'accès homogène aux différentes séquences [81].

1.4.6.1 Segmentation de la vidéo

Durant l'édition d'un document hypervidéo, il est nécessaire de faire référence à des vidéos et des parties de vidéos. Par exemple, un prétraitement nécessaire pour la définition des fragments et l'annotation consiste en la structuration du flux continu en une organisation permettant d'adresser des parties particulières de la vidéo. La sélection d'un objet audiovisuel est étroitement liée au processus de segmentation, qui permet de définir un découpage en unités de base [31].

Segmentation : les techniques de segmentation permettent l'organisation de la vidéo suivant une structuration claire en tenant compte de ses composantes sonore et visuelle. Le processus de segmentation - virtuelle ou réelle - de la vidéo en sous-intervalles et sous-régions peut être automatisé par des algorithmes de traitement de signal et de détection de discontinuités spatiales et temporelles.

Suivant le regroupement de ses images (*frames*), différentes entités peuvent être distinguées, résultant en une représentation hiérarchique du flux vidéo. La structuration la plus classique de la vidéo suit une décomposition linéaire. Se basant sur des descripteurs visuels extraits à partir de l'ensemble des images composant la vidéo, les techniques de segmentation permettent d'en extraire des unités élémentaires appelées *shots* ou *plans*. L'ensemble des plans successifs sera regroupé en *scènes* suivant l'unité de lieu. Les scènes ou plans, formant la même unité de sujet, seront regroupées en *séquences*.

Fossé sémantique : introduit dans le cadre de l'indexation sémantique des images, le concept de *fossé sémantique* est défini dans [89] comme étant le manque de concordance entre les informations que les machines peuvent extraire depuis les documents numériques, et les interprétations que les humains en font. Dans le domaine audiovisuel, l'absence de la sémantique des descripteurs visuels est le problème majeur de la segmentation, particulièrement quand elle est automatisée. Les flux audiovisuels qui n'ont pas de structure bien définie souffrent de manque d'applicabilité alors que la segmentation automatique comme seul moyen de décrire la vidéo n'est pas satisfaisante.

Avec une segmentation réalisée manuellement, l'aspect sémantique de cette tâche pourrait être respecté. Alors que les unités sémantiques détectées par l'analyse automatique sont beaucoup plus des *descripteurs*, l'analyse manuelle et la classification reconnaissent d'autres unités généralement appelées *concepts* [94]. Par conséquent, l'une de ces méthodes ne peut remplacer l'autre ; l'indexation automatique pourrait plutôt servir d'assistance à tout processus de description manuelle de documents [97].

1.4.7 Hypervidéo comme vidéo interactive à base d'annotations

1.4.7.1 Séparation contenu/rendu

Un document électronique n'ayant pas une forme physique inhérente, n'est perceptible que selon l'interprétation qu'un système fait des données qu'il contient. Un document électronique peut être donc défini comme un ensemble de données organisées en une structure stable associée à des règles de mise en forme pour lui permettre d'être lu aussi bien par son concepteur que par ses différents lecteurs [69]. Vue l'importance du rendu d'un document, sa forme perceptible peut être adaptée suivant un certain nombre d'exigences en ajustant le processus de création du rendu. Cette notion de séparation des données de leur rendu est connue dans la littérature par le paradigme de *document engineering* (ingénierie documentaire) correspondant à la séparation du contenu de son style de rendu.

La structure d'annotation associée à une vidéo peut être utilisée pour assurer la séparation du contenu de son rendu, lors de la création des hypervidéos.

1.4.7.2 Modélisation à base d'annotations

L'association d'une structure d'annotations à la vidéo permet de définir une infrastructure qui peut être utilisée pour augmenter le média avec des données pouvant être instrumentées pour construire multiples vues du même document. La modélisation à base d'annotations greffe une nouvelle couche de données sur celle du flux vidéo [47, 23]. Chaque annotation est associée à un segment vidéo logique qui est en général un extrait du flux défini par ses instants de début et de fin correspondant aux emplacements des *frames* de début et de fin.

Les modèles de documents audiovisuels qui séparent le contenu de son rendu moyennant une structure d'annotations ont plusieurs avantages [47] :

- différentes granularités d'accès à la vidéo peuvent être considérées car les annotations peuvent adresser des segments logiques de taille arbitraire ;
- plusieurs annotations peuvent être associées au même segment logique pour fournir différentes vues de la même donnée ;
- des segments logiques peuvent se chevaucher ou s'imbriquer arbitrairement ;
- les annotations peuvent être ajoutées ou supprimées indépendamment des flux vidéo associés, permettant la création et la modification dynamique et incrémentale des annotations vidéo ;
- la recherche et l'extraction à base de contenu sémantique grâce aux annotations manuelles peuvent être réalisées à un niveau que les techniques actuelles d'édition graphique et techniques de vision par ordinateur ne peuvent pas réaliser.

L'ajout des annotations améliore donc de manière considérable le rôle actif de l'utilisateur et augmente l'interactivité de la vidéo [28] car sans moyens de naviguer dans un document audiovisuel, celui-ci demeure essentiellement un médium narratif qui supporte difficilement une lecture active [86].

1.4.8 Annotations dans les hypervidéos

La numérisation des documents permet d'instrumenter leur description [94]. Les annotations forment une modélisation intermédiaire décrivant l'organisation et le contenu multimédia constitutif de la vidéo. Les annotations aussi diverses de part leur mécanisme d'extraction/définition que le moyen de réalisation (manuelle/supervisée/automatique), permettent de décrire la nature et la connaissance propre à la vidéo et les relations entre ses différentes parties dans un langage accessible à l'homme ainsi qu'à la machine.

La présence d'annotations permet de développer des techniques qui assurent des activités documentaires diverses telles que la définition de multiples parcours de navigation et la recomposition dynamique de documents [94]. L'adoption des concepts de l'hypermédia au domaine de l'annotation vidéo permet de briser la linéarité de la vidéo et de créer un espace d'information non linéaire, définissant le concept d'hypervidéo [60].

1.5 Conclusion

La richesse, le réalisme et l'expressivité de l'audiovisuel font que la vidéo soit le média le plus intéressant des documents multimédia tout en étant le plus difficile et le plus déifiant, de part sa complexité et sa structure non linéaire, continue et fortement dépendante du temps. Les hypervidéos sont une forme de production nouvelle issue de l'application des concepts de l'hypertexte aux hypermédiyas à base audiovisuelle. C'est une forme intéressante de présentation et de visualisation liée au document et éventuellement aux annotations associées. Une synergie entre les différents domaines constituant les facettes des hypervidéos est au préalable nécessaire pour mieux les appréhender, s'en servir et les améliorer. Partant de ce constat, un tour d'horizons de l'état de l'art propre à tous ces domaines sera étalé dans le prochain chapitre ainsi que les travaux relatifs à la conception et la présentation des hypervidéos avec un accent particulier sur le Web comme environnement de diffusion.

Chapitre 2

Etat de l'Art

L'UTILISATION de la vidéo comme véhicule de transmission de la connaissance date de plus d'un quart de siècle. Néanmoins, l'émergence de systèmes permettant de tirer profit des opportunités et potentialités de l'hypermédia à forte composante audiovisuelle a mis du temps pour se vulgariser. Dans ce chapitre, l'état de l'art associé aux hypervidéos - modèles et systèmes - est présenté puis discuté et les insuffisances relevées permettront d'envisager d'autres alternatives pour de meilleures réponses aux nouvelles exigences.

2.1 Modèles, Langages et Standards Multimédia

2.1.1 Modèles de référence de l'hypermédia

Un modèle de document hypermédia fournit une description de l'organisation des données contenues dans un document [99] et lui donne un cadre applicatif général. Pour cela, il définit la nature de données manipulées, la méthode de stockage et de mise en relation des éléments et comment le document peut être visualisé et navigué [76].

2.1.1.1 Modèle de référence de Dexter

Le modèle de référence de l'hypertexte de Dexter (DHM : *Dexter Hypertext Model*) [33] a été créé à la fin des années 1980s comme résultat de plusieurs rencontres d'un groupe de chercheurs et développeurs de systèmes hypermédias. Il définit une architecture articulée sur trois couches distinctes, présentée à la figure 2.1 :

- Couche de stockage (*Storage Layer*) : elle représente le point focal du modèle et est responsable du stockage, localisation et accès aux composants du document. L'entité fondamental dans cette couche est le composant (*component*), une généralisation des

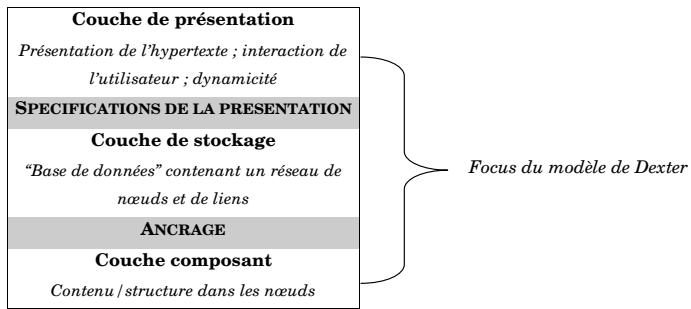


Figure 2.1 – Architecture en couche du modèle de Dexter

nœuds, liens et composites. Un composant peut être atomique, un lien ou un objet composite construit à partir de plusieurs autres composants.

- Couche de présentation (*Runtime Layer*) : elle gère les fonctionnalités temps réel et la représentation des composants permettant à l'utilisateur de voir et manipuler les différentes structures, prenant en charge les interactions entre l'utilisateur et le document.
- Couche de composant (*Within-component Layer*) : responsable des contenus et des structures internes des nœuds, cette couche n'est pas élaborée dans le modèle, elle est du ressort des applications implémentant le modèle. Point fort du modèle, cela lui permet d'être appliqué à des types arbitraires de données et ne restreignant pas les types de structure et de contenu qui peuvent être inclus dans les nœuds [35].

Le principal intérêt de ce modèle est l'indépendance entre la couche de stockage et la couche de contenu grâce à un mécanisme d'ancre des éléments de la couche de stockage dans ceux de la couche de contenu (i.e. un mécanisme d'adressage des seconds par les premiers). Bien qu'aucun système n'ait pu implémenter intégralement ses spécifications, le modèle a servi de base pour la développement de plusieurs systèmes dont le Web. L'avènement du multimédia fait naître le besoin à un modèle plus large, prenant en charge l'aspect multimédia des documents, particulièrement de ceux dépendants du temps.

2.1.1.2 Modèle de référence de Amsterdam :

Le but du modèle d'Amsterdam (AHM : *Amsterdam Hypermedia Model*) [36] est de considérer les aspects hypertexte et multimédia dans une description unifiée du document. Il étend le modèle de Dexter en lui permettant la prise en compte du contexte dans le suivi des liens, ainsi que la gestion de la temporalité, nécessaire pour les documents multimédia. Le modèle AHM définit la notion de *Synchronization Arcs* qui sont des arcs permettant aux composants d'être décrits les uns relativement aux autres. Pour mieux gérer la demande et le partage de ressources, le modèle définit la notion de *Channels* ou canaux qui

sont des attributs de présentation de haut niveau. Décrits au niveau de la couche *Storage* au même titre que les mécanismes permettant de gérer la politique d'allocation de ces derniers aux différents composants, les *channels* permettent la communication entre l'utilisateur et le composant. La stratégie d'allocation et la gestion de la concurrence sont assurées par le couche *Runtime*. Les *channels* peuvent aussi être utilisés pour définir des caractéristiques de présentation telles que le style et la taille de la police, le volume sonore, la langue par défaut, etc.

2.1.2 Modèles spatiaux et temporels du multimédia

2.1.2.1 Modèles de placement et de formatage spatial

Deux familles de modèles spatiaux peuvent être distinguées : les modèles absous et les modèles relatifs [7]. Dans un modèle absolu, la description de l'espace et de l'emplacement des médias est faite en termes de coordonnées à deux ou trois axes par rapport à un référentiel spatial. Les modèles spatiaux relatifs, quant à eux, utilisent des formes continues et le placement se fait en spécifiant des relations spatiales entre les médias. Ces relations peuvent être classés en deux familles : *les relation directionnelles* et les *relations topologiques*. Les relations directionnelles [67] décrivent l'ordre des médias dans l'espace. Une région est présentée par deux intervalles correspondant respectivement aux axes *x* et *y*. L'ensemble des relations directionnelles est obtenu en combinant les intervalles de deux médias sur ces deux axes. Les relations topologiques décrivent la position relative d'un objet par rapport à d'autres objets et indiquent si deux médias se superposent, se touchent ou sont disjoints [22].

2.1.2.2 Modèles temporels et de synchronisation

Les approches de spécification de la synchronisation peuvent être regroupées en trois grandes familles [54] : les approches opérationnelles à base de programmation événementielle, les approches opérationnelles à base de structures d'arbres ou de graphes et les approches à base de contraintes d'intervalles ou d'instants.

La définition du temps des documents multimédia est classifiée dans [7] en modèle à instants, modèles événementiels, modèles à intervalles et modèles à base de scripts.

Les modèles à instants : ces modèles définissent les instants de début et de fin des objets temporalisés via la notion d'instant ou point temporel dont la taille en termes de moments temporels est nulle. Le modèle le plus représentatif de cette classe est celui de l'*axe temporel absolu* ou *Timeline*. Ce modèle s'appuie sur la métaphore de ligne de

temps qui place chaque élément sur un axe temporel absolu en l'estampillant avec son instant d'occurrence relativement au référentiel. LimSee [21] est l'un de système à modèle absolu les plus récents. Une telle modélisation est néanmoins statique, rend difficiles les modifications et ne permet pas la prise en compte d'interaction totalement imprévisible.

Les modèles à intervalles : cette classe de modèles considère chaque élément multimédia comme une entité temporelle décrite par sa durée exprimée en un intervalle qui est défini comme une portion ou étendue temporelle, souvent délimité par ses instants de début et de fin [1].

Langages de scripts : une approche à base de scripts met en œuvre une représentation procédurale pour spécifier le comportement de la présentation. Les outils à base de timelines, du fait de leur pouvoir d'expression limité, ont été le plus souvent complétés par des langages de programmation.

Structures de graphes : dans les approches fondées sur les graphes, le document est représenté sous la forme d'un diagramme de flot de contrôle qui décrit l'interaction entre les médias du document à l'exécution. L'édition d'un document peut ainsi s'appuyer sur une description plus riche et plus facile à utiliser de la synchronisation temporelle.

2.1.3 Langages et Standards Multimédia

Les standards de conception et de présentation multimédia ont pour but de stabiliser la définition des formats, les cadres logiciels et les interfaces applicatives tout en incorporant la conceptualisation des médias et leurs techniques de composition.

2.1.3.1 HyTime

HyTime (Hypermedia/Time-based Structuring Language) [26] est un standard ISO pour la représentation et la structuration des documents multimédia et hypermédias, la représentation des liens hypertextuels et la spécification de l'ordonnancement des événements spatiaux et temporels ainsi que de la synchronisation des médias. N'étant pas un langage de présentation, il peut être vu comme un simple méta-langage de description de documents multimédia. Par conséquent, un outil de transformation doit être utilisé pour produire une version exécutable du document. HyTime étant d'un niveau d'abstraction très élevé, l'élaboration de formes de présentation est très compliquée, ce qui limite son utilisabilité et sa popularité.

2.1.3.2 MHEG

Le but du groupe de travail MHEG (Multimedia and Hypermedia information coding Expert Group) [58] était de proposer des normes de spécification afin de standardiser la représentation et donc l'échange d'objets élémentaires multimedia et hypermédias. Le standard MHEG se compose de plusieurs parties notamment les parties MHEG-5 et MHEG-6. La partie MHEG-5 est destinée à supporter la distribution des applications multimédia dans une architecture client/serveur à travers différents types de plates-formes. MHEG-5 est spécifiée afin de permettre le développement d'un interpréteur du format d'échange MHEG, qui requiert très peu de ressources.

2.1.3.3 SMIL

Le langage *Synchronized Multimedia Integration Language* (SMIL) [11] est un standard XML déclaratif développée par le Consortium W3C pour la description des présentations multimédia interactives. Ne se focalisant pas sur le codage de données, SMIL permet leur intégration en spécifiant comment les composants sont reliés temporellement et spatialement. Sa spécification définit le comportement des documents via un ensemble de groupes fonctionnels, chaque groupe étant lui-même partitionné en modules et qui sont des collections d'éléments et d'attributs. L'implémentation d'un module implique la prise en charge de toutes ses spécificités. Divers *profils* fournissent des regroupements par défaut de modules ; la spécification SMIL 3.0³. définit les profiles suivants : LANGUAGE, UNIFIEDMOBILE, DAISY et TINY.

Un document SMIL comporte deux parties, une en-tête et un corps de document. L'en-tête est délimitée par les balises `<head>` et `</head>` et contient la définition des zones d'affichage (*layout*) et de leurs régions dans lesquelles les objets synchronisés dans la présentation sont rendus. Ces zones peuvent être définies de manière imbriquées et selon des modes de positionnement complexes.

Le corps du document comporte les définitions des objets à synchroniser, de leur synchronisation, ainsi que les animations, liens et transitions sur les objets. Les données étant référencées et non comprises dans le document SMIL, elles peuvent être partagées et réutilisées entre plusieurs documents.

La synchronisation des objets se fait principalement à travers des balises qui définissent un conteneur temporel, les fils de cet élément étant les éléments encapsulés dans le conteneur ; trois balises sont définies dans SMIL :

- `<seq>` pour la présentation en séquence ;

3. La nouvelle spécification SMIL 3.0 est disponible à l'adresse :<http://www.w3.org/TR/SMIL3/>

- <par> pour la présentation en parallèle ;
- <excl> pour la présentation en mode exclusif, un seul des fils étant présenté à la fois avec la possibilité d'indiquer des priorités.

Le comportement d'un scénario SMIL repose sur une sémantique à base d'intervalles. Chaque élément, objet média ou bien conteneur temporel, possède deux listes indiquant les occurrences possibles des débuts et des fins, listes qui peuvent changer dynamiquement lors de l'exécution du scénario.

Le listing 2.1 présente un document écrit en SMIL. Dans cet exemple, trois régions *a*, *b* et *c* sont définies afin d'afficher trois données médias identifiées par les mêmes noms. La composition temporelle consiste en la mise en parallèle de l'audio *a* avec la séquence de l'image *b*, présentée cinq secondes, et la vidéo *c*, arrêtée par l'audio.

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

Listing 2.1 – Exemple d'un document SMIL

2.1.4 Discussion

La définition de la structure et du contenu d'un document passe par l'élaboration d'un modèle de données qui décrit les entités constituant le document et les relations entre elles. Un tel modèle permet de préciser les règles selon lesquelles les informations sont organisées dans l'espace informationnel et les mécanismes et paradigmes permettant les diverses manipulations des documents s'y conformant. Un tel modèle pour les hypervidéos n'existe pas encore.

Considérant l'hypervidéo comme sous-domaine de l'hypermédia, nous avons brièvement présenté les modèles de référence du domaine, Dexter et Amsterdam. Développés pour couvrir tous les concepts pertinents des systèmes de l'hypertexte et de l'hypermédia, actuellement il n'y a aucune méthodologie de modélisation qui répondent entièrement à leurs spécifications [104]. En plus de ce manque d'applicabilité, la littérature hypertextuelle soulève le fait que ces modèles ne prévoient pas de mécanisme d'accès et d'adressage des composants atomiques qui sont considérés comme boîtes noires (*black boxes*) [96], empêchant l'ordonnancement des objets qui les composent - caractéristique prédominante dans les hypervidéo. Le comportement dynamique des documents est spécifié uniquement par des liens, gérés au niveau de la couche *runtime* [96], sans considérer d'autres mécanismes de changement dynamique de la présentation tels que les événements.

Ce constat établi par rapport à l'applicabilité de ces modèles s'ajoute au fait que les concepts de ces deux modèles sont trop généraux et leurs exigences trop restrictives pour être adoptés comme modèles hypervidéos. L'hypervidéo en tant que spécialisation de l'hypermédia et telle qu'expérimentée aujourd'hui ne peut être pleinement et efficacement théorisée par ces modèles bien qu'elle puisse être décrite sommairement d'une manière peu précise. Une telle démarche n'est pas solide et nous croyons qu'un modèle plus spécifique à même de propulser le développement d'applications et de nouvelles techniques de consommation de l'information audiovisuelle est encore nécessaire.

Les documents hypervidéos véhiculent des données qui sont retransmises à la présentation suivant une organisation spatiale et temporelle. Par conséquent, la modélisation des hypervidéos doit prendre en compte ces deux dimensions qui sont inhérentes à tout document multimédia. Actuellement le multimédia a fait objet de plusieurs travaux qui lui a permis de gagner en maturité avec des modèles spatiaux et temporels adaptés et adoptables à diverses exigences.

De tous les standards et langages multimédia disponibles, SMIL nous paraît un candidat potentiel, voire idéal, pour la définition des hypervidéos. En effet, il présente un modèle multimédia très complet et très robuste permettant la composition de documents hypervidéos d'une manière consistante. Néanmoins, les concepts que SMIL introduit sont

européennes aussi très généraux et ne décrivent pas précisément les propriétés intrinsèques des vidéos interactives et encore moins celles des hypervidéos. En outre, SMIL est un langage assez volumineux dont la spécification elle-même [11] ne contient aucune définition formelle de son modèle [43]. Comme indiqué dans [21], la manipulation directe de langages trop bien établis (sur-spécifiés) tels que SMIL est très compliquée pour la plupart des utilisateurs, ce qui nécessite une profonde compréhension de la sémantique du langage.

2.2 Hypervidéos et Annotations

2.2.1 Standards d'annotation multimédia

L'annotation vidéo requiert un effort considérable et fait émerger le besoin d'offrir à l'utilisateur un moyen de réaliser ses propres annotations d'une manière qui a une certaine valeur durable (pour réutilisations, maintenance, diffusion, etc.). Plusieurs initiatives visent le développement de normes répondant à de tels critères ; on peut en citer à titre indicatif les exemples de *Dublin Core*, *MPEG-7*, *AnnodeX* et *Cinelab*.

2.2.1.1 Dublin Core

La multiplication de besoins et la diversité des structures et des nomenclatures de métadonnées ont motivé l'élaboration d'un vocabulaire minimal pour décrire des annotations. Ce standard de métadonnées appelé Dublin Core⁴ est défini sous l'égide de l'*Online Computer Library Center* (OCLC) et fait l'objet du groupe de travail «Métadonnées de l'Information Multimédia - Dublin Core» au sein du Comité Européen de Normalisation. L'ensemble d'éléments du Dublin Core est une version très réduite des notices MARC qui ne comprend que quinze éléments (*title*, *creator*, *subject*, *description*, *publisher*, *contributor*, *date*, *type*, *format*, *identifier*, *source*, *language*, *relation*, *coverage*, *rights*). La signification de certains d'entre eux peut être précisée, étendue ou raffinée.

Comme beaucoup d'autres standards, Dublin Core traite de l'interopérabilité en permettant le partage de ressources entre différentes plateformes et différents domaines d'application. Étant dominés par leur orientation *ressources*, de tels standards ne font souvent que décrire un contenu au niveau de l'entité sans pouvoir pointer des parties de ces entités[56]. Bien que ce niveau soit très important, il n'englobe pas les besoins d'annotation plus fine d'un contenu audiovisuel et ces standards n'ont pas de modèles assez robustes pour aller au-delà de cet adressage [56].

4. <http://dublincore.org/>

2.2.1.2 MPEG-7

MPEG-7, connu aussi comme *Interface de description standard du contenu multimédia* [63], est un standard XML développé par le groupe *ISO/WG11 (MPEG- Moving Picture Expert Group)* en 2001. Il a pour objectif de répondre aux exigences opérationnelles qu'elles soient ou pas de nature temps réel, tout en étendant les capacités limitées des solutions propriétaires spécialisées dans la description du contenu, notamment par l'inclusion de nouveaux types de données.

Le standard MPEG-7 fédère et supporte différentes descriptions telles que les caractéristiques du contenu de niveau bas (forme, taille, texture, couleur, mouvement, position, etc.), les informations sur le contenu sémantique (personnage, lieu, action, etc.), ou encore les descripteurs culturels (auteur, date de création, format, etc.). Il est également possible, en utilisant ce standard, de décrire les relations de nature temporelle ou spatiale entre les objets du média.

Le format vise à standardiser un ensemble de mesures quantitatives pour représenter les caractéristiques des objets audiovisuels appelé *Descripteurs* (D), la structure des descripteurs et leur relations appelées *Schémas de Description* (DS). Un autre travail de *MPEG-7* concerne la standardisation d'un langage appelé *Langage de Définition de la Description* (DDL) qui permet de spécifier la description des schémas. Les considérations de rendus et de visualisations ne sont pas abordées par ce standard.

2.2.1.3 Annodex

Annodex [71] est un format développé par le *Commonwealth Scientific and Industrial Research Organisation (CSIRO)* pour l'annotation et l'indexation des médias continus (l'audio et la vidéo notamment). Ce standard essaie de faire de la vidéo ce que les liens Web (URL/URI) ont réussi à faire avec le texte et l'image, dans le contexte Web. Le standard a pour objectif de permettre l'adressage d'un instant d'une ressource Web continue (temporelle) via des pointeurs et des liens. Le projet *MetaVid*⁵ est une démonstration des capacités d'*Annodex* appliquées à des vidéos de U.S.Congress[56].

2.2.1.4 Format CineLab

Le modèle Cinelab⁶ tient ses origines de l'évolution du projet Advène dans la perspective de proposer un format d'échange entre les partenaires du projet Cinelab. Le modèle est construit en couches. La première couche nommée *modèle noyau* est assez générale

5. <http://metavid.ucsc.edu/>

6. <http://liris.cnrs.fr/advene/cinelab.html>

pour répondre à une grande variété d'usages, aussi indépendamment que possible des évolutions technologiques. La deuxième couche, nommée *modèle applicatif*, a vocation à concrétiser le modèle noyau par un certain nombre de choix techniques qui le rendent directement implémentable. Plusieurs modèles applicatifs peuvent exister au dessus du modèle noyau bien que la vocation du projet Cinelab est de trouver un modèle applicatif commun à tous les partenaires. Une application peut aussi répondre à un modèle applicatif plus large que celui proposé.

Suivant le modèle CineLab, les annotations sont considérées au delà de leur contenu en leur définissant la manière suivant laquelle elles sont contraintes, mises en relation et rendues dans les documents finaux. Les annotations sont classées suivant des types définis par l'annotateur, pouvant être contraintes par des schémas d'annotation, et sont des données attachées à des fragments du flux audiovisuel. Elles peuvent être en relation entre elles et un contenu spécifique peut être associé à ces liens entre les annotations.

Les *vues* sont définies comme des manières de présenter le contenu du document audiovisuel annoté au lecteur. Définies par l'auteur, elles permettent de générer facilement des représentations hypervidéos, d'où l'intérêt que nous portons à ce modèle.

2.2.2 Outils d'annotation vidéo

Plusieurs systèmes ont été développés dans le but de permettre l'annotation manuelle de la vidéo. **VIA**(Video and Image Annotation)⁷ est un outil permettant d'annoter des images et vidéos. **EXMARaLDA** (EXtensible MARkup Language for Discourse Analysis)⁸ permet de transcrire les interactions verbales entre de nombreux interlocuteurs. Les interactions de comportements non-verbaux sont considérées comme des annotations des énoncés verbaux. L'outil de transcription **Anvil** [46] a pour objectif de faciliter l'annotation et l'analyse des données verbales et non verbales dans le domaine des sciences du langage. Son interface ergonomique permet l'alignement temporel de divers types d'annotations et de graphiques avec le signal sonore et la vidéo. *Anvil* permet l'export sous forme textuelle de chaque piste d'annotations.

Le prototype **Elan** (EUDICO Linguistic Annotator)⁹ a pour objectif de répondre à des besoins d'annotation et d'exploitation d'enregistrements multimédia. Spécialement conçu pour l'analyse du langage oral, du langage des signes, de la gestuelle, il peut également être utilisé pour toute annotation, analyse et documentation de corpus médiatiques. Contrairement à *ANVIL* et *EXMARaLDA*, il ne permet pas de mettre en relation les

7. <http://mklab.iti.gr/via/>

8. <http://www.exmaralda.org/>

9. <http://www.lat-mpi.eu/tools/elan/>

niveaux d'annotation.

D'autres outils d'annotation exploitent les ontologies pour l'annotation audiovisuelle. **Ontolog** [37] est un de ces système. Ses métadata sont fournies grâce à deux ontologies par défaut : *Dublin Core Element Set* et *Dublin Core Qualified Element Set*.

Des systèmes s'intéressant à l'annotation collaborative tels que le prototype **Vannotea** [85] permettent d'une manière collaborative l'indexation, le parcours, la description, l'annotation et la discussion de film ou de contenu vidéo à une très grande échelle.

Diver (Digital Interactive Video Exploration and Reflection) [68] permet de créer, modifier et partager des *DIVES* qui sont des perspectives annotées d'une vidéo. DIVER pour Desktop permet à l'utilisateur d'importer des sources de vidéo et de créer des nouveaux *chemins* annotés à travers la vidéo source. Le nouveau film annoté devient propriété de l'auteur de la *DIVE*. *WebDIVER* permet aux *DIVERs* de partager une *DIVE* et d'autres utilisateurs peuvent y apposer leurs commentaires.

Advène(Annotate Digital Vidéos, Exchange on the NET) [4] est un système de création et de gestion d'hypervidéos basées sur les annotations. Le système permet à un lecteur d'annoter une vidéo, puis d'utiliser ces annotations pour contrôler le flux temporel et le présenter suivant différentes vues. La plateforme Advène permet d'une part de créer et d'éditer des hypervidéos, en annotant la vidéo d'origine et en définissant des vues basées sur ces annotations, et d'autre part de visualiser les hypervidéos ainsi créées, composées du document et de ses annotations présentés suivant ces vues.

La technologie MPEG-7 est très utilisée dans les systèmes d'annotation. Par exemple, l'outil d'IBM **VideoAnnEx**¹⁰ permet d'annoter les vidéos avec des descripteurs MPEG-7 en imposant aux annotations des exigences exprimées en terme de schéma de description (DS) MPEG-7 correspondant.

D'autres systèmes développés pour des besoins spécifiques existent. Par exemple, **ViPER-GT** (Video Performance Evaluation Resource Kits Ground Truth)¹¹ a été créé dans le but d'aider à l'évaluation des approches traitant de l'automatisation de l'analyse sémantique de la vidéo. Il permet la création et l'édition des annotations *frame-par-frame* au niveau de la scène et de l'objet.

Comparison entre les différents outils : Le tableau 2.1 résume les caractéristiques de certains de ces systèmes selon plusieurs critère :

- le *format* vidéo que prend en charge l'outil ;
- le format d'annotation résultant comme *sorties* à la fin du processus d'annotation ;

10. <http://www.research.ibm.com/VideoAnnEx/>

11. <http://viper-toolkit.sourceforge.net/>

Outil	Format	Sorties	OS	Schéma	Granularit	Type	Type d'annotation
VIA	MPEG-1, MPEG-2	Projet VIA (annotations + ontologie)	Tous	OWL, texte libre	video, segment, frame	Annot.	général, descriptif, structurel
EXMARaLDA	Formats supportés par JMF	XML	Tous	Fixe	video, partition	Annot.	descriptif (transcription)
Anvil	MPEG-1, MPEG-2, MOV, AVI	XML Schema	Tous	Libre	video, points, video segment	Annot.	général, descriptif, structurel
Elan	MPEG, MOV, Wav	XML (EU-DICO)	Tous	Fixe	video	Annot.	descriptif
Ontolog	Formats supportés par JMF ¹² ou Quick-time	domain ontology (RDFS), Dublin Core ES, Ontolog Schema ontology	Tous	Ontologies	video, segment	Annot.	général, descriptif, structurel
Vannotea	MPEG-1, MPEG-2, MPEG-4	MPEG-7	Win32	Fixe	video	Annot.	général, descriptif, structurel
VideoAnnex	MPEG-1, MPEG-2	MPEG-7	Win32	Fixe	video, frame, shot, scène	Annot.	général, descriptif, structurel
Advene	Formats supportés par Vlc	recueil XML et différents formats d'export	Win32, Linux, MacOs	Libres	video, segment	Annot. Visual.	général, descriptif, structurel
Diver	Formats supportés par Quick-time	DIVE format	Tous	Aucun	video	Annot. Visual.	général, descriptif, structurel
ViPER-GT	MPEG-1	Viper (XML)	Tous	Fixe	video	Annot. Visual.	général, descriptif, structurel

Tableau 2.1 – Comparaison des outils d'Annotation de la Vidéo

- les systèmes d’exploitation (*OS*) sur lesquels l’outil peut fonctionner
- les possibilités de définitions de *schémas* de description pour organiser les annotations : *fixe* si les schémas ne sont pas modifiables ; *aucun* s’il n’y a pas de possibilité de définir de schémas ; *ontologies* si les schémas doivent être définis comme des ontologies et *libres* si les schémas sont construits par l’utilisateur ;
- la *granularité* des unités informationnelles audiovisuelles sur lesquelles les annotations peuvent être définies.
- *type* de l’outil : annotation ou visualisation. Certains systèmes sont dédiés à l’annotation uniquement alors que d’autres permettent la visualisation et la navigation.
- le *type d’annotation* ou l’objectif de l’annotation. On peut identifier plusieurs types d’annotations : (1) les *annotations descriptives* qui renseignent du sujet traité ; (2) les *annotations structurelles* qui combinent les aspects spatial et temporel et la décomposition spatio-temporelle et (3) les *annotations générales* qui sont des données génériques de type date de création de l’annotation, créateur, droits d’accès, etc.

A partir de ce tableau, plusieurs remarques peuvent être faites. Notons d’abord que la plupart des systèmes se focalisent sur l’ajout de l’information comme moyen d’enrichir la vidéo et peu d’outils, à part *Advène* et *Viper-Gt*, combinent à la fois l’ajout d’information sur le document audiovisuel par annotation suivant des schémas de description, et la visualisation de ces documents enrichis par les informations ajoutées.

Les technologies du Web sémantiques ne sont pas utilisées à grande échelle, XML étant le choix le plus commun pour la saisie et la représentation des annotations produites. L’utilisation des descriptions *MPEG-7* peut en effet constituer une solution pour normaliser la description vidéo bien que cela soulève encore d’autres questions relatives par exemple au traitement automatisé des annotations, en particulier les annotations descriptives, à un niveau plutôt sémantique.

À cause du peu de performance des algorithmes de détection automatique des unités sémantiques de la vidéo, les segments temporels sont souvent détectés manuellement, en délimitant leurs bornes temporelles. Seuls *Advène* et *VideoAnnex* permettent d’effectuer une analyse de flux pour la détection automatique des unités d’informations (*shots*). Depuis peu, *Anvil* permet d’annoter une région spatio-temporelle, mais actuellement ces régions sont représentées par certains points précis d’une *frame* vidéo. La représentation de l’information structurelle, et par conséquent temporelle, d’une manière efficace minimisant la taille des structures d’annotation est encore un point qui n’est pas largement discuté. Enfin, signalons que jusqu’à présent, le support du processus d’annotation collaborative est négligeable et la plupart des outils sont des applications autonomes, mis à la disposition du public pour un usage non-commercial.

2.2.3 Discussion

L'idée d'intégrer la vidéo interactive en utilisant des modèles à base d'annotations a motivé notre présentation de quelques standards et outils d'annotation. Plusieurs standards et formats de données peuvent être utilisés pour une telle description des documents audiovisuels. Le modèle Cinelab nous paraît très intéressant car résultant de travaux antérieurs ayant pour objectif d'annoter les vidéos, de générer des présentations et d'échanger des données entre différents acteurs de l'audiovisuel. La description en elle-même peut être réalisée en langage plus commun tel que MPEG-7 et le projet Advene montre plusieurs possibilités d'annoter suivant ce schéma Cinelab et d'exporter en divers autres formats. Le mérite du prototype Advene est donc de proposer des outils permettant d'une part d'annoter des vidéos et d'autre part de générer des hypervidéos à base de structures d'annotations résultant. Le problème principal du modèle d'Advene - comme nous le verrons dans les prochains paragraphes - est relatif au fait que ses hypervidéos sont traitées comme des vues riches du média audiovisuel et le projet ne propose pas de représentation théorique assez générique des documents hypervidéos.

2.3 Systèmes et Environnements Hypervidéos

2.3.1 Premiers systèmes hypermédias à forte composante vidéo

Historiquement, les origines des hypervidéos remontent aux premiers jours de l'hypertexte, lorsque Ted Nelson - l'*inventeur* de l'hypertexte dont il a proposé le terme en 1965 - a étendu son modèle pour inclure la notion du «*branching movie*»[65]. Le premier véritable système hypermédia est créé par à MIT en 1979 [6]. Ce système, baptisé **Aspen Movie Map**, permet l'exploration virtuelle et interactive de la ville de Colorado (Aspen), au moyen de cartes géographiques, de vidéos de la ville (rues, bâtiments, etc.) et de liens hypertextuels. Ce projet a inspiré le développement du système *HyperPlant* chez Hitachi Research Laboratory en 1992 [92] pour le contrôle à distance de machines d'une centrale électrique grâce à des vidéos prises en temps réelles. Tous les objets de la vue caméra peuvent être identifiés et manipulés, des contrôles virtuels peuvent être effectués sur eux. Ces contrôles seront réellement traduits en actions concrètes sur la machine.

En 1989, au sein de MIT Media Lab est réalisée *Video Finger* [98], l'une des premières démonstrations de la video hyperliée avec des «objets segmentés». Similaire à l'utilitaire **finger** sous Unix, il permet de choisir n'importe quelle partie de l'écran contenant des personnes en mouvement pour afficher des informations détaillées sur l'identité de la personne sélectionnée.



Figure 2.2 – Prototype HyperCafe

Créé par Interctive Cinema Group au sein MIT Media laboratory en 1989, *Elastic Charles* [64] est une édition hypermédia du journal «American River Magazine», introduisant la notion de lien *video-à-video* grâce à des *Micons* (*moving icons*) qui apparaissent lors de la lecture d'une vidéo.

La technologie de l'hypermédia à base audiovisuelle a mis du temps pour porter ces idées à pleine réalisation [15, 16]. Les expériences qui ont suivi ont contribué à l'émergence et l'évolution de l'hypervidéo. Le concept et l'approche hypervidéo n'ont été réellement présentés et discutés comme tels qu'avec l'application de la vidéo hyperliée *HyperCafe* [82].

2.3.2 Systèmes d'édition et de présentation des hypervidéos

Hypercafe : Hypercafe [82] est un projet expérimental illustrant des concepts généraux de l'hypervidéo. Il place l'utilisateur dans un café virtuel dans lequel ont lieu plusieurs conversations, et lui offre la possibilité de naviguer entre celles-ci. Le spectateur peut donc suivre une conversation (vidéo) puis en choisissant de suivre un lien hypertexte temporellement situé, il peut partir vers une autre vidéo.

Le lecteur n'a réellement aucun contrôle sur le flux temporel de la vidéo, il se contente de le suivre et d'activer ou d'ignorer les liens qui lui sont proposés. Les liens sont présentés au lecteur comme des propositions événementielles de manière opportuniste - des *opportunités de liens* -, pouvant être temporels, spatiaux ou spatio-temporels, en ajoutant un flux et en augmentant le volume sonore pour exprimer le fait qu'un lien entre le flux principal et celui-ci est disponible. Si le lecteur décide de suivre ce lien, le flux principal devient inactif, c'est à dire qu'il continue de se dérouler dans le temps mais son niveau sonore baisse et il devient moins important, tandis que la cible du lien devient à l'inverse, active. Si l'utilisateur souhaite suivre un lien, il se retrouve dans une autre situation, suivant une autre conversation, sinon le lien disparaît au bout de quelques instants. Entrer dans



Figure 2.3 – Prise d'écran du prototype Hyper-Film

hypercafé est donc similaire à pénétrer dans un café et tenter de suivre les conversations aux différentes tables.

Les auteurs du prototype ont proposé plusieurs nouveaux concepts relatifs à la structure et à la navigation au sein de ce genre de document, définissant un modèle assez général pour les pratiques de l'hypervidéo [100]. Les scènes sont considérées comme étant les unités structurelles et les *séquences narratives* sont un chemin potentiel à travers un ensemble de scènes liées.

Les technologies et méthodologies décrites pour la navigation dans un «espace vidéo» sont expérimentales et non testées[100] mais elles ont permis d'introduire une expérience nouvelle pour l'utilisation de l'hypermédia à forte composante audiovisuelle.

Hyper-Film : Le système Hyper-Film [53] a été développé dans le but d'enrichir les hyperstructures vidéo avec d'autres unités d'information (images, vidéos, fichiers, pages HTML, texte,...). Écrite en Flash, son interface est conçue comme application Web. La structure du document est présentée sur une région de la fenêtre appelée *table des liens*. Cette table présente la structure interactive du document mettant à la disposition de l'utilisateur un moyen de naviguer au sein de ce dernier.

Les régions sensitives (hotspots) ne peuvent être définies que sur quelques images (*frames*) de la vidéo. Ainsi, même si le système permet qu'une région sensible soit attribuée à un emplacement précis dans une frame d'une séquence vidéo, aucun comportement temporel ne peut lui être attaché. Avec cette restriction, l'utilisateur ne peut interagir qu'avec ces frames particulières, correspondant à certains objets de la vidéo. Il est impossible de sélectionner des objets directement d'une vidéo en cours. La visualisation des régions sensitives grâce aux possibilités offertes comme couleur et apparence est un trait très intéressant de l'application.

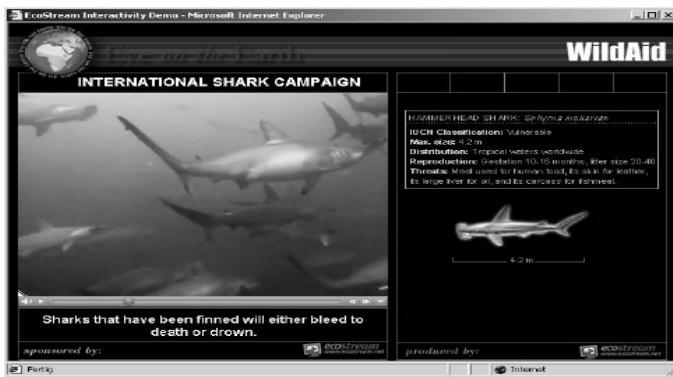


Figure 2.4 – Interface utilisateur du système VideoClix

VideoClix : VideoClix¹³ est une application Web pour l'édition hypervidéo publiée en 2001 par la société eLine Technologies¹⁴ (fondée en 1999 et rachétée en 2006 par la société VideoClix Inc). Avec l'avantage que ses vidéos peuvent être lues grâce à des lecteurs populaires tels que QuickTime et Flash, ce produit s'est avéré être un succès commercial.

Comme Hyper-Film, VideoClix est basé sur des vidéos prétraitées. Les outils d'édition de VideoClix offrent beaucoup d'options et couvrent une partie très large du processus de production des documents hypervidéos. VideoClix permet qu'une région sensitive soit définie par des paramètres temporels sur une séquence vidéo. Ces régions ne sont pas représentées sur l'interface de présentation. Le seul moyen pour les localiser est via le mécanisme *roll-over* qui implique un changement du curseur de la souris au passage sur une telle région.

Les données additionnelles utilisées pour l'édition hypervidéo sont incrustées avec le flux vidéo de telle manière que le contenu vidéo ne soit plus disponible séparément du reste de la structure hypervidéo. Par conséquent, la création d'une nouvelle région sensitive et d'une référence influe sur la structure interne du film rendant ce dernier inexploitable en dehors des outils fournis avec VideoClix.

Hyper-Hitchcock : L'environnement hypervidéos Hyper-Hitchcock[87] s'appuie sur le modèle *detail-on-demand hypervideos* décrit dans [30] et qui introduit une forme simplifiée de l'hypervidéo, permettant la définition d'un seul lien pouvant être activé à un instant donné de la présentation. Les fonctionnalités de base de l'environnement Hyper-Hitchcock sont la définition de liens de navigation entre les composants vidéo structurés hiérarchiquement et la définition de propriétés pour les liens leur associant des étiquettes ainsi que des comportements de retour et de la reprise.

13. <http://www.videoclix.tv/>

14. <http://www.elinetek.com/>



Figure 2.5 – Interface utilisateur du système Hyper-Hitchcock

Les fragments vidéo représentent les unités de base du système et peuvent être regroupés pour former des vidéos composites qui, à leur tour, peuvent composer d'autres vidéos composites de plus haut niveau. Des liens peuvent exister entre les différents objets de la vidéo, des fragments ou des composites. Les sources des liens sont définies par l'intervalle de temps pendant lequel le lien est disponible au lecteur alors que la cible est la séquence vidéo qui sera présentée si le lien vient à être activé.

Un seul lien peut être attaché à un élément donné et un seul lien est disponible au lecteur à chaque instant de la présentation. Un élément, par contre, peut être cible de plusieurs liens. Si plusieurs niveaux d'une hiérarchie ont des liens qui se chevauchent, celui attaché à l'élément le plus en bas (le plus profond) est rendu disponible.

Les outils d'édition sont séparés du système de présentation dans Hyper-Hitchcock, l'utilisateur ne pouvant éditer le document lors de sa présentation. D'autre part, l'interaction avec la vidéo n'est jamais directe, elle passe par un espace de navigation séparé. En effet, le système ne prend pas en charge la notion de région sensitive et ne permet pas d'agir sur un objet de la vidéo ni d'adresser une partie du segment spatialement.

HyperSoap : HyperSoap [8] est un système dans lequel le spectateur peut interagir grâce à une télécommande augmentée (figure 2.6). Il permet d'afficher des informations contextuelles sur des objets vidéo prédéfinis, notamment la façon dont ils peuvent être achetés. Ce projet permet notamment d'affiner considérablement la technique des régions sensitives (hotspots) interactives.

HyperSoap se distingue des autres systèmes par le développement d'algorithme permettant le suivi d'objets d'une vidéo, son système auteur implémentant des techniques de suivi très avancées permettant la définition de régions sensitives ambulantes. La qualité d'affichage de ces régions est très appréciable. Cependant, cette particularité fait qu'il est moins adapté pour une utilisation sur le Web. Pour une utilisation commerciale, un dérivé



(a) Affichage d'information à propos d'un élément sélectionnée avec HyperSoap (b) Interaction avec HyperSoap par un pointeur laser durant une projection vidéo

Figure 2.6 – Interface utilisateur du système HyperSoap

du système appelé *WatchPoint* est fourni, permettant l'utilisation de HyperSoap dans les applications de commerce électronique.

Non utilisable sur réseau, le système de visualisation consiste en un lecteur DVD, un projecteur vidéo, un écran et un pointeur laser qui permet la sélection d'objets sur l'écran pour affichage d'information sur ce dernier.



(a) Outil d'édition de VisuaSHOCK Movie (b) Présentation VisuaSHOCK Movie

Figure 2.7 – Interface utilisateur du système VisuaSHOCK Movie

VisualSHOCK MOVIE : VisualSHOCK MOVIE [40] est un système d'édition et de présentation des vidéos hyperliées (figure 2.7) qui comporte un outil d'édition appelé *Movie Map Editor* et un outil de présentation qui est aussi disponible comme contrôle Active-X, un plugin Netscape destiné exclusivement pour l'environnement Windows.

Le système d'édition permet de créer et de modifier des régions sensitives et de les représenter sous forme de rectangles ou de cercles dont la visibilité est paramétrable. Pour suivre le mouvement d'un objet, la frame de début et celle de la fin doivent être délimités et définies comme ancrés. Le mouvement de l'objet peut aussi être suivi manuellement.



Figure 2.8 – Prise d'écran de la démonstration Hot Norman

HotVideo : HotVideo [24] est un système qui permet la définition de lien hypertextuels appelés *hotlinks* sur la vidéo en attachant des segments vidéo à différentes sources de données textuelles, sonores, graphiques ou autres. Avec un système auteur très flexible, l'apparence des hotlinks et le comportement de l'interface peuvent être personnalisés par l'utilisateur. La figure 2.8 représente une célèbre démonstration de HotVideo appelée *Hot Norman*.

HVet : HVet (Hipervídeo Veterinário) [93] est un environnement hypervidéo d'apprentissage interactif sur le Web développé en HTML, CSS, PHP, JavaScript et MySQL et ayant pour objectif de constituer un dépôt de vidéos traitant de la chirurgie vétérinaire. Contrairement à la plupart des systèmes hypermédias existant, HVet reprend l'idée de *Multilink* [49] pour définir des liens multi-destinataires. L'activation d'un tel lien passe par l'affichage d'une fenêtre de sélection à l'utilisateur pour qu'il choisisse la destination désirée. D'autre part, s'appuyant sur l'étude des modes, les phases et les styles cognitifs de l'apprentissage ainsi que l'étude des théories de la cognition (*Dual Code Theory* et *Cognitive Theory of Multimedia Learning*) pour proposer un ensemble de choix de présentation. Les annotations dans HVet permettent d'ajouter des notes textuelles à une hypervidéo qui seront visibles et pérennes sur le document. Alors que des liens peuvent être définis vers des instants de la présentation, des hotspot peuvent être définies à partir d'un segment. HVet utilise *VideoClix*¹⁵ pour créer les objets liés de la vidéo et pour suivre leur mouvement.

Advène : Advène [2] (voir la figure 2.9) est un système qui permet au lecteur l'annotation de la vidéo et l'utilisation des annotations pour contrôler le flux temporel de la vidéo et le présenter suivant différentes vues.

15. www.videoclix.tv

2.3. Systèmes et Environnements Hypervidéos

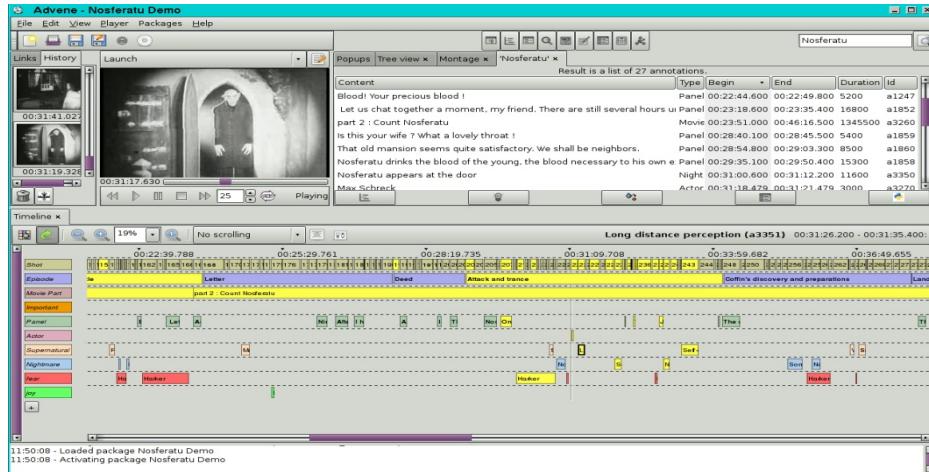


Figure 2.9 – Interface du prototype Advene

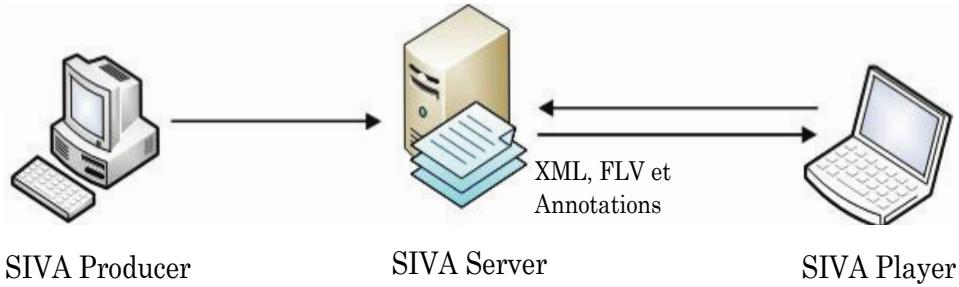


Figure 2.10 – Modules de l'environnement SIVA Suite

Advene est une plateforme permettant d'une part l'annotation de documents audiovisuels suivant des schémas d'annotation et d'autre part la visualisation sous forme d'hypervidéos de ces documents annotés. La notion centrale du modèle d'Advene est celle de recueil : Un recueil est le regroupement de l'ensemble des annotations d'un document audiovisuel, classées suivant des types et organisées entre elles par des relations, ainsi que des vues. Celles-ci représentent des ensembles de fragments du flux annotés et organisés pour produire des hypervidéos. L'ensemble formé du document audiovisuel, des annotations et des relations entre celles-ci constitue la couche de stockage. La partie visualisation est assurée par les vues qui définissent un angle de perception de l'hypervidéo.

SIVA Suite : L'environnement SIVA Suite [57] est composé de trois composants (voir figure 2.10) : *SIVA Producer*, *SIVA Player* et *SIVA Server*. L'objectif de SIVA Player est le rendu de vidéos interactives réalisées avec SIVA Producer et fournies par SIVA Server.

SIVA Producer permet d'éditer la vidéo et de lui ajouter des annotations textuelles

ou graphiques ou des liens à des scènes prédéfinies ou pour toute la vidéo. Le système fournit un utilitaire de détection et de découpage de la vidéo en scènes, un diagramme de scènes pour spécifier un montage non linéaire ainsi qu'un éditeur HTML pour créer des annotations textuelles et des liens.

Une vidéo interactive est décrite par un document XML qui spécifie les interactions, annotations et relations des scènes vidéo. Le contenu vidéo est fourni via un format vidéo Flash (flv). Un projet SIVA peut être exporté vers un fichier XML formaté et valide schéma ainsi que des vidéos flash, pouvant être interprété par le Player.

2.3.3 Comparaison entre les systèmes

Les systèmes, environnements et langages de création et de présentation des hypervidéos dont nous avons présenté un échantillon, ont permis l'émergence de l'hypervidéo comme nouvelle forme de documents audiovisuels et des hypermédiyas centrés sur la vidéos. Ces systèmes partagent plusieurs caractéristiques et diffèrent sur d'autres :

Orientation Web : pour le besoin d'une meilleure diffusion de l'information, certains systèmes tels que *HyperFilm*, *VideoClix* et *VisualShock Movie* sont développés comme des applications Web qui implémentent des architectures client-serveur ; ils s'exécutant sur navigateurs moyennant des greffons (plugins) appropriés.

Suivi d'objets : la référence à des objets en mouvement est possible avec *VideoClix*, *HyperSoap* et *VisualSHOCK MOVIE*. L'algorithme de suivi de mouvement dans *HyperSoap* est l'un des plus puissants ; ceux utilisés par *VideoClix* et *VisualSHOCK MOVIE* sont plutôt semi-automatiques, l'auteur décidant, sur une certaine période de temps, l'emplacement de la région dans une séquence vidéo. La complexité technique de tels systèmes est très élevée, surtout à cause des régions définies comme ancrés au sein du flux vidéo.

Représentation des ancrés hypervidéos : la représentation des ancrés hypervidéo est souvent très mal gérée. L'affichage des régions correspondant, quand cette option existe, est souvent fait d'une manière inopportunne perturbant l'activité de lecture. Cependant, sans ces régions, l'utilisateur n'a pratiquement aucun moyen d'interactivité et de contrôle sur la présentation. Le comportement par défaut quant à la visibilité ou non de ces régions est souvent géré par l'auteur et parfois dépend des préférences du lecteur. *VideoClix* par exemple ne fait que signaler l'existence d'une telle région au lecteur quand il la survole pour décider de l'afficher ou pas. Toutefois, cela requiert de l'utilisateur qu'il surveille constamment et balaie souvent l'interface de présentation de la vidéo.

Annotations : bien que la notion d'annotation prise dans son sens le plus général - toute donnée ajoutant de la connaissance à un flux audiovisuel - soit supportée par la majorité des systèmes, il n'y a que *Advène* qui permet de générer des hypervidéos à base de structures d'annotation bien formalisées.

Altération du flux : des systèmes tels que *HyperFilm* et *VisualSHOCK MOVIE* favorisent un stockage séparé des différentes unités d'information de la structure du document. La définition des régions sensitives ainsi que les liens est gardée séparément et la vidéo n'est pas modifiée. Toute modification ou extension de l'hypervidéo n'a aucune influence directe sur le contenu vidéo du document. *VideoClix* par contre adopte la démarche contraire. Toutes les données sont enregistrées sur le même fichier résultant en un format propre qui ne peut être exploité par d'autres applications.

2.3.4 Discussion

L'approche de la plupart des systèmes hypervidéos est axée sur le développement d'outils pour des besoins spécifiques mettant en jeu des documents audiovisuels munis des capacités d'interaction. Ces systèmes héritent des concepts et pratiques issus de l'expérience hypermédia. Ils ne discutent pas des bases théoriques sur lesquels les documents sont produits et ne fournissent un cadre théorique bien défini. De ce fait, ils présentent une grande hétérogénéité dans la façon de définir les hypervidéos, ce qui résulte en manque de conformité des réponses que ces systèmes réservent aux besoins et aux exigences des hypervidéos. Or, le volet préalable de modélisation formelle doit être considéré dans le but d'avoir des références théoriques bien définies. Actuellement, aucune réelle standardisation de ces concepts dans le cadre bien défini de l'hypervidéo n'existe réellement. La plupart des modélisations s'appuient sur des aspects pratiques, exprimées en termes de besoins pour des domaines spécifiques et cherchent un compromis combinant des considérations architecturales, techniques et conceptuelles atténuant ainsi la généralité des concepts associés et introduisant des restrictions le plus souvent relatives aux particularités de l'implémentation et de la mise en oeuvre.

En dehors des modèles génériques de l'hypertexte/hypermédia, peu de travail a été accompli dans le sens de mettre sur pied une modélisation propre à l'hypervidéo, si l'on excepte quelques tentatives telles que le modèle *detail-on-demand hypervideo* qui est plutôt restrictif. Le modèle de *Advène* permet de conceptualiser les hypervidéos comme vues et son modèle est beaucoup plus propriétaire que générique.

2.4 Hypervidéo sur le Web

2.4.1 Le multimédia sur le Web

Plusieurs efforts de standardisation ont eu lieu dans le domaine du multimédia sur le Web sans réel succès jusqu'à présent puisqu'aucun standard ou outil de création multimédia diffusé à grande échelle n'utilise leurs propositions.

2.4.1.1 Dimension spatiale des documents Web

Les documents Web, souvent exprimés en termes de document HTML (et XHTML), sont organisés spatialement d'une manière absolue ou relative sur l'interface de rendu. Le modèle spatial sous-jacent a des capacités très limitées pour l'expression de la disposition et de l'agencement spatial de documents électroniques. Les langages de spécification de style tels que CSS, XSL et DSSSL offrent un vocabulaire de plus grand degré d'expressivité, pour compléter le langage HTML et de le doter d'un modèle spatial plus puissant. Ceci inclut la définition des propriétés de mise en page et de style, tels que les types et tailles de polices, pour les différents types de composants du document. La sortie du traitement d'un document avec une feuille de style maintient le même flux de base du document original. La spécification de style modifie l'apparence de chaque objet dans ce flux, et l'apparition d'aspects globaux de la présentation du document dans son ensemble, mais ne modifie pas la progression fondamentale de ce flux.

2.4.1.2 Le temps sur le Web

Le contenu multimédia présente une opportunité et un défi à la communauté Web : des objets massifs de contenu opaque constraint temporellement doivent être définis, publiés, trouvés, transférés et rendus à travers une infrastructure qui est développée d'une manière '*time-insensitive*', insensible au temps et surtout utilisant des navigateurs, des interfaces et des outils qui sont optimisés pour des données atemporelles. Les créneaux les plus importants à investir doivent permettre le développement d'outils et des technologies qui assureront que le multimédia devienne plus utile sur le Web [42].

La définition d'un comportement temporel aux documents structurés permet de leur associer une sémantique temporelle [43]. La dimension temporelle étant absente du Web exprimé par le standard HTML, pour les documents nécessitant des fonctionnalités de synchronisation, ce format pour le rendu n'est clairement pas adapté. Comme conséquence de la nature atemporelle du Web actuel, les applications Web multimédia n'ont pu être développées à grande échelle. Ceci a motivé plusieurs travaux qui ont eu pour objectif de

proposer de nouveaux standards et/ou mécanisme de temporalisation du Web.

Le langage SMIL [11, 12] a été dès sa première version proposé comme standard d'expression multimédia sur le Web. XHTML+SMIL [83], connu aussi comme HTML+SMIL, est une note W3C qui décrit l'intégration des concepts SMIL avec XHTML et CSS. Le langage partage plusieurs modules avec SMIL (modules de temporalisation, de définition des objets, de liens, etc.). Le module spatial de SMIL est remplacé par celui de HTML. L'implémentation la plus connue du langage est celle réalisée par MicroSoft portant le nom de HTML+TIME [84]. Le support de la dimension temporelle est fourni à titre accessoire pour le document atemporel. HTML+TIME utilise un moteur HTML, fournissant un support étendu pour la mise en style avec CSS. Le contrôle du temps est fourni grâce à un modèle de conteneurs temporels proche de celui utilisé dans DFXP.

2.4.2 Outils de diffusion multimédia sur le Web

Plugins : Le Web à l'origine ne permet pas de présenter un contenu riche et continu tel que celui délivré par l'audiovisuel. Grâce aux mécanismes des plugins des navigateurs, une plus grande intégration du multimédia est possible. Ces applications tierces sont capables de traiter un certain nombre de types de contenu multimédia. De nombreux plugins permettent d'ajouter du son, des animations et des vidéo aux documents Web.

Un plugin est un composant binaire destiné à apporter une fonctionnalité additionnelles à une application - dans ce cas, aux navigateurs - qu'elle ne sait pas gérer nativement. Par exemple, le plugin Adobe Reader¹⁶ permet d'ouvrir directement des fichiers PDF dans le navigateur, et les plugins Adobe Flash¹⁷, Quicktime¹⁸, Silverlight¹⁹ et RealPlayer²⁰ sont utilisés pour afficher entre autres des composants multimédia dans une page Web.

Exemple de Flash : Flash est un environnement de développement intégré (IDE) développée à l'origine par la société Macromedia(rachetée en 2005 par Adobe Systems). Le terme «Flash» peut se référer à un lecteur, un environnement ou à un fichier d'application. Il offre la possibilité aux concepteurs de créer des présentations interactives avec le lecteur. Les projets réalisés en flash peuvent inclure des animations, du contenu vidéo, du texte animé, du son, etc. Les éléments de contenu sont appelés généralement «applications». Les fichiers Flash, généralement appelés *animation Flash* portent l'extension *.swf*. Ils peuvent être inclus dans une page web et lus par le plugin Flash du navigateur, ou bien interprétés

16. <http://www.adobe.com/products/reader/>

17. <http://www.adobe.com/products/flash/>

18. <http://www.apple.com/quicktime>

19. <http://www.silverlight.net/>

20. <http://www.real.com/realplayer>

indépendamment dans le lecteur Flash Player. Un document flash se compose de trois parties : la bibliothèque, le scénario et le code «*ActionScript*».

Le multimédia et HTML5 : *HTML5* (HyperText Markup Language Version 5)²¹, le nouveau standard du Web, porte une révision majeure à ses prédecesseurs *HTML4.01*, *HTML1.0* et *DOM Level 2* du HTML et s'emploie à intégrer le multimédia en natif pour ne plus le considérer comme ensemble d'entités externes. *HTML5* propose que les pages Web soient traitées comme des conteneurs atemporels d'éléments continus `<video>` et `<audio>`. Ces nouveaux éléments multimédia ont des attributs, un comportement, une spécification DOM et une mise en style standardisés. Malgré un relatif succès - la part de la vidéo *HTML5* disponible sur le Web est passée de 10% en Janvier 2010 à 63% en Février 2011, selon la firme MeFeedia's qui a réalisé une enquête-audit sur environ 33 000 sources de données vidéo²² -, grâce notamment à son adoption par les principaux acteurs de la vidéo sur Internet comme YouTube, DailyMotion et Vimeo, plusieurs insuffisances préviennent *HTML5* de devenir un format pleinement multimédia.

SMIL Timesheets 1.0 : ce standard²³ est introduit par la spécification SMIL pour la définition des «feuilles de temps (Timesheets)». Les Timesheets est une spécification W3C qui définit un langage de temporalisation XML permettant de rendre disponible les éléments et attributs des modules Temps et Synchronisation de SMIL 3.0, à une plus grande panoplie de langages à base de XML, y compris les langages absolument atemporels comme HTML. Alors que les langages de feuilles de style comme CSS ajoutent du style et la gestion spatiale aux documents XML, les Timesheets ajoutent un comportement temporel à ces langages.

Comme les feuilles de style, la définition de Timesheet peut se faire à l'intérieur du document HTML ou sur un fichier externe permettant leur réutilisabilité. Néanmoins, les Timesheets diffèrent sur le fait que les éléments des Timesheets pointent directement sur les éléments HTML, alors que c'est un élément HTML qui pointe sur un élément CSS. En plus des conteneurs de temps du langage SMIL (par, seq et excl), deux éléments particuliers sont définis : l'élément `<timesheet>` détermine le début et la fin de la feuille de temps alors que l'élément `<item>` permet avec l'attribut *select* d'associer un élément Timesheet à un nœud du document HTML, remplaçant les éléments `<video>`, `<audio>`, `` et `<text>` du langage SMIL.

21. <http://www.w3.org/TR/html5/>

22. Enquête disponible à l'adresse : <http://blog.mefeedia.com/html5-feb-2011>

23. <http://www.w3.org/TR/timesheets/>

Groupes de travail W3C : Le consortium W3C (World Wide Web Consortium) comporte un ensemble de groupes de travail pour proposer des standards pour la vidéo sur le Web. Le but des activités des groupes de *Video in the Web Activity*²⁴ est de propulser l'utilisation de la vidéo sur le Web. Le groupe *MFWG : W3C Media Fragments Working Group* a pour objectif de développer un mécanisme unique à base de URI (Uniform Resource Identifiers), permettant d'identifier des fragments spatiaux et temporels des objets média (image, vidéo, audio) sur le Web. Le groupe de travail *TTWG : Timed Text Working Group* examine les possibilités de représenter sous une format XML un contenu textuel auquel un comportement temporel et un style de présentation ont été ajoutés. Le groupe s'est intéressé à la fin de 2009 à produire la spécification du format DFXP (Distribution Format Exchange Profile). Enfin, le groupe *MAWG : The Media Annotations Working Group* a pour but de fournir une ontologie et une API pour faciliter l'intégration de données inter-communautaires à propos d'objets média sur le Web. D'autre part, l'initiative *Web Accessibility Initiative (WAI)*²⁵ a pour objectif d'étudier et d'améliorer l'accessibilité Web aux utilisateurs avec handicaps.

2.4.3 Vidéo enrichie sur le Web

Actuellement, les services vidéo en ligne permettent beaucoup de pratiques enrichissantes du contenu audiovisuel. Il est par exemple possible avec ces services de poster un commentaire sur un certain contenu vidéo, ajouter une évaluation (*rating*) et partager un certain contenu par l'intégration d'un code HTML qui inclut le contenu audiovisuel.

Des sites tels que Youtube²⁶ et DailyMotion²⁷ ont gagné une audience incontestable grâce à leurs efforts pour vulgariser la publication de l'information audiovisuelle. Ces environnements ont montré que la vidéo peut effectivement servir pour le partage de la connaissance du message le plus simple au plus complexe sur Internet [48].

La figure 2.11 montre quelques possibilités offertes par Youtube. La présentation de la vidéo est enrichie de diverses manières : des liens internes sont proposés pour aller vers des fragments bien définis et d'autres liens vers d'autres vidéos peuvent être aperçus. D'autre part, on peut distinguer un texte publicitaire du fournisseur de contenu ainsi que des portions textuelles incrustées au flux. Youtube et ses récents développements permettent d'autres possibilités et offrent des services très attrayants et attractifs.

24. <http://www.w3.org/2008/WebVideo/>

25. <http://www.w3.org/WAI/>

26. <http://www.youtube.com/>

27. <http://www.dailymotion.com>

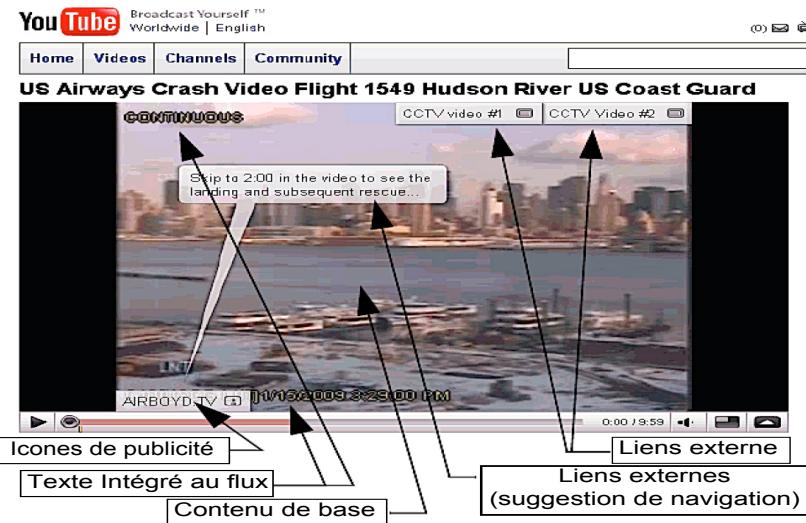


Figure 2.11 – Interface riche de YouTube

2.4.4 Hypervidéo et Web

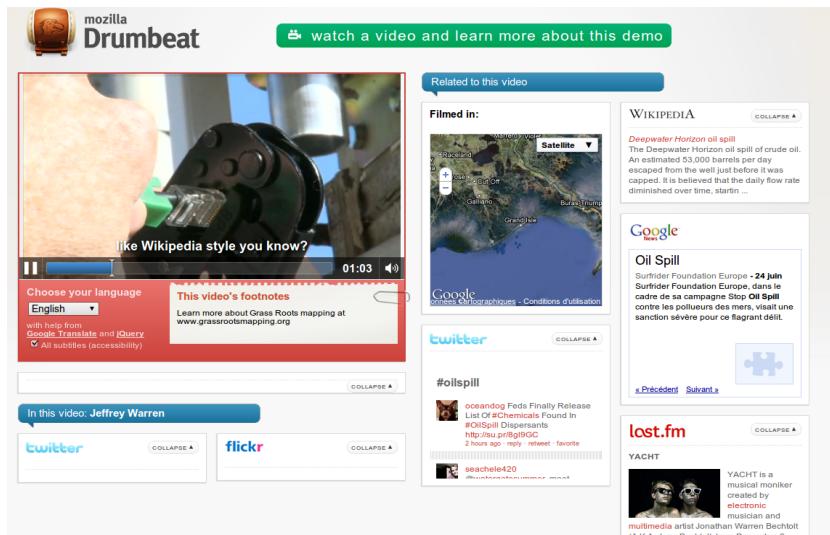


Figure 2.12 – Prototype Popcorn

Des formes primitives de l'hypervidéo sur le Web ont été expérimentées dans plusieurs projets tels que *VideoClix*²⁸, *Asterpix*²⁹, *Adivi*³⁰, *wireWax*³¹ et *hotclicktv*³². Avec le nouveau standard HTML5, de nouvelles expériences de la vidéo interactives commencent à

- 28. <http://www.videoclix.tv/>
- 29. <http://www.asterpix.com>
- 30. <http://www.adivi.net/>
- 31. <http://www.wirewax.com/>
- 32. <http://www.hotclicktv.com/>

émerger. L'exemple du projet *Popcorn.js*³³ (figure 2.12), est très intéressant. Popcorn.js est un projet Mozilla Drumbeat Web Made Movies concernant l'ajout de métadonnées à la vidéo HTML5. Il permet à la vidéo d'être liée au reste du Web, en définissant des passerelles avec l'espace hypermédia via des données d'origines diverses : Google Maps, Twitter et Wikipedia.

2.4.5 Discussion

Avec les techniques actuelles, l'hypervidéo - à l'image des autres applications multimédia - ne peut être pleinement et correctement exprimée sur le Web qui, comme souligné dans [70], ne permet pas encore l'émergence de pratiques vidéo à même de faire de cette dernière une entité de première classe, «*a first class citizen*». En effet, la vidéo sur le Web n'est pas facilement *recherchable*, navigable, recomposable et mise en cache. Il n'y a aussi aucun moyen de pointer des fragments ou des objets définis dans sa structure ou dans son flux. Malgré les développements actuels, beaucoup de modèles et de fonctionnalités restent à développer[14] si ce n'est à inventer. Par exemple, le partitionnement (ou fragmentation) de la vidéo en temps et en espace n'est pas encore possible, la vidéo est encore considérée comme unité atomique. Par conséquent, la navigation intra-objet n'est pas possible et des portions d'objets ne peuvent être partagées par les utilisateurs.

Le Web enrichi grâce aux plugins a permis de s'affranchir de plusieurs insuffisances des standards Web en offrant la robustesse des langages de programmation aux développeurs et aux navigateurs. Les mécanismes de plugin ont néanmoins plusieurs insuffisances et failles [42]. L'accès aux services offerts par les navigateurs étant basé sur une API (*Application Programming Interface*), ces interfaces sont généralement limitées en terme de types d'interaction possibles avec le navigateur mais aussi avec le reste du contenu, éventuellement présenté via des plugins. Comme résultat, les plugins donnent en général un contenu opaque incrusté sur la page avec des possibilités d'interaction très limitées. Le format des plugins n'étant pas normalisé, des problèmes d'interopérabilité se posent : les moteurs de recherche n'indexaient pas leur contenu, les feuilles de style CSS ne leur sont pas applicables, les fonctions de recherche du navigateur n'y accèdent pas, l'historique de navigation ne les inclut pas. En plus, l'utilisation des plugins dans une page ajoute une charge supplémentaire sur la CPU, ralentissant les navigateurs et accentuant les vulnérabilités sécuritaires.

Bien que le standard HTML5 ait introduit la possibilité de support des médias continus (**audio** et **video**) en natif, ce qui décharge l'utilisateur et les navigateurs des plugins, la

33. <http://popcornjs.org>

temporalité de ces éléments ne dépasse pas leur propre portée. En effet, HTML5 ne fournit pas de conteneur temporel pour ses documents, ce dernier est de ce point de vue atemporel. Les éléments média qu'il introduit ont une ligne de temps continue et propre à l'objet présenté. Or, une telle définition du temps diffère de celle suggérée par les modèles de documents multimédia interactifs. Au sein de ces documents, plusieurs lignes de temps sont définies et reliées, basées sur des conditions, des événements, des interactions et tout type d'interruptions, dans le but d'offrir une expérience multimédia complète.

Notre intérêt pour le Web nous dissuade d'adopter SMIL comme modèle ou comme langage. A l'exemple des plugins tels que Flash, aucun navigateur ne le prend en charge en natif et les quelques systèmes le manipulant sont pour la plupart développés à des fins expérimentales. Le désintérêt qui a accompagné le développement de SMIL a fini par dissuader quant à son utilisation. Ce constat a fait objet de plusieurs réflexion et c'est dans ce contexte que la spécification *SMIL Timesheets* par les concepteurs de SMIL 3.0[11] a été élaborée pour tirer profit du modèle temporel et de synchronisation très robuste de SMIL. L'utilisation des *Timesheets* serait une solution idéale pour la temporalisation des langages XML du moment qu'une information temporelle peut être attachée à n'importe quel élément d'un langage XML [59].

2.5 Conclusion

Nous avons fait un tour d'une partie relativement importante de modèles, langages, standards et outils décrivant divers aspects du domaine de l'hypervidéo.

La revue des travaux existant et les insuffisances que nous avons pu relever montrent clairement qu'au moment où des domaines généraux tels que l'audiovisuel, le multimédia et l'hypermédia sont bien modélisés et leurs technologies offrent des choix divers qui couvrent les besoins actuels, l'hypervidéo n'a pu se détacher de ces différents domaines et elle est encore aujourd'hui considérée comme l'intersection de ces derniers, soulevant des considérations beaucoup plus techniques alors que nous croyons que les besoins sont d'abord formels. Le besoin d'un modèle expressif et dédié motive notre présent travail. Dans le prochain chapitre, nous développons notre proposition de modèle de données pour caractériser les hypervidéos. Nous essayerons de tirer au mieux profit de l'existant tel que les possibilités offertes par les annotations et les différents modèles décrivant les dimensions multimédia des documents, pour proposer une nouvelle approche permettant de mieux étudier le domaine et d'ouvrir la voie à de nouvelles spécifications et pratiques audiovisuelles.

Chapitre 3

CHM : Modèle de Documents Hypervidéos

AFIN de répondre au besoin de modélisation des hypervidéos exprimé dans le chapitre précédent, et dans l'objectif de contribuer à mieux cerner leurs spécificités et d'asseoir des bases théoriques qui tiennent en compte les différents aspects qu'elles présentent, un modèle de document est décrit dans ce chapitre. L'hypervidéo aligne les capacités de l'hypertexte, la richesse du multimédia avec les particularités de l'audiovisuel. Par conséquent, le modèle proposé est décrit suivant les trois dimensions : *audiovisuelles*, *multimédia* et *hypermédia*. Le modèle *CHM - Component-based Hypervideo Model* - se propose comme cadre théorique qui introduit un formalisme abstrait indépendant de toute technologie dans le but d'exposer un cadre unificateur pour la description de tout ce que l'hypervidéo d'aujourd'hui permet de réaliser.

3.1 Principes du modèle

3.1.1 Concept d'hypervidéo dans CHM

L'hypervidéo est définie dans le modèle comme étant une spécialisation de l'hypermédia centrée sur un ou plusieurs documents audiovisuels (généralement une vidéo ou un corpus de vidéos) annotés. La structure d'annotations associée à la composante audiovisuelle permet d'une part de la délinéariser et lui définir des comportements interactifs, et d'autre part de lui associer différents rendus multimédia riches, comportant divers artefacts de visualisations comme les sous-titres, les illustrations et les enrichissements. Le contenu associé enrichit et explique la partie audiovisuelle principale autour de laquelle la présentation est organisée en espace et en temps.

Les nouvelles formes d'interactivité - celles de la vidéo combinée à celles de l'espace hypermédia - supposent l'intégration complète et effective de la vidéo interactive pour procurer à l'utilisateur des alternatives de visualisation, d'interaction et d'utilisation attractives non-conventionnelles, des moyens de transmission et d'acquisition de connaissance robustes ainsi que la possibilité d'organiser et d'explorer l'espace documentaire suivant des choix et des parcours non linéaires personnalisables [34].

3.1.2 Les Annotation comme infrastructure de conception

La vidéo n'ayant pas de structure de données explicite, le besoin de la décrire, de la structurer et de la délinéariser a motivé la proposition d'un modèle à base d'annotations. Plusieurs auteurs considèrent l'avènement des hypervidéos comme conséquence de l'intérêt que portent les systèmes hypermédias/multimédia conventionnels pour les modèles et les techniques d'annotation. Dans ce sens, l'hypervidéo peut être vue comme le résultat de l'adoption des méthodes de l'*hypermédia* à la *vidéo numérique annotée* [24].

Avec les modèles à base d'annotations, la linéarité de l'information audiovisuelle est brisée au profit d'un espace d'information non linéaire [60] et une nouvelle couche de contenu est définie au dessus du flux audiovisuel [47]. Cette couche introduit une infrastructure capable d'augmenter le média avec de la structure et de l'interactivité, et l'enrichir avec de l'information et du contenu, induisant la possibilité de construire diverses représentations et applications du document. La composition de présentations hypervidéos peuvent être conçues à base de ces structures comme des vues riches voire complexes du document, et qui héritent des techniques audiovisuelles, rehaussées par les capacités avancées et avérées de l'hypermédia et du multimédia.

3.1.3 Modélisation logique à base de composants

La modélisation logique à base de composants des documents hypermédias est une représentation usuelle que l'on peut retrouver dans plusieurs modèles tels que ceux de Dexter et de Amsterdam. Dans ce dernier, un composant peut désigner un nœud hypermédia, un lien ou une composition de composants.

Le modèle CHM ou *Component-based Hypervideo Model* que nous proposons définit le composant comme concept clé dénotant une entité abstraite servant de réceptacle de données et/ou pouvant être utilisée pour la définition d'une représentation homogène d'un nombre de concepts du modèle. CHM utilise le principe d'héritage et d'imbrication de composants pour la création et la représentation hiérarchique de documents. La technique d'imbrication de composants - en particulier ceux de haut niveau - pour la composition

d'applications est très utilisée dans le génie logiciel et représente un des processus les plus robustes dans le *software engineering* [79]. De ce fait, à partir d'entités élémentaires, des composants de base sont définis permettant à leur tour l'introduction d'autres composants, ce qui suggère une décomposition modulaire des hypervidéos. Alors que la proposition d'un ensemble de composants de haut niveau simplifie les tâches de conceptualisation, voire de codification - pour la mise en œuvre du modèle - , CHM est extensible car permettant que d'autres composants de haut niveau soient créés selon le besoin à travers ceux existants (de bas et de haut niveaux).

3.1.4 Adaptation des paradigmes de l'hypermédia

Un des buts principaux du modèle est de proposer des définitions formelles concises des paradigmes rencontrés dans les documents hypervidéos. Les concepts clés de l'hypertexte/hypermédia tels que les noeuds et les liens comme définis par les modèles hypermédias ne sont pas assez expressifs et/ou n'ont pas toujours la même signification pour les hypervidéos. Par exemple, durant une navigation, le retour à un noeud déjà visité constitue une pratique très usuelle avec la navigation Web. Un tel retour en arrière dans le contexte d'une hypervidéo ne peut pas exprimer le fait de "*rembobiner*" la présentation alors qu'en réalité un retour en arrière peut signifier effectivement rembobiner la présentation. Pour tenir compte de tels cas courants, le modèle recense un ensemble minimal de concepts qu'il discute et modélise via une éventuelle redéfinition. Dans le modèle, cette redéfinition est décrite avec des composants de niveau utilisateur (des composants de haut niveau).

3.1.5 Dimensions multimédia de l'hypervidéo

Une hypervidéo est nécessairement un document multimédia constitué de média discrets et d'autres continus, organisés en espace et en temps. Par conséquent, le modèle hypervidéo doit pouvoir spécifier et gérer les différentes dimensions du multimédia. Un modèle multimédia sert à décrire des aspects différents des documents incluant des structures et des représentations : logique, spatiale, temporelle et hypertextuelle.

3.1.6 Gestion dynamique à base d'événements

La modélisation des hypervidéos par CHM implique deux niveau fonctionnels : le niveau *déclaratif* et le niveau *opérationnel*. Le niveau déclaratif est constitué de l'ensemble de composants constituant la structure stable du document, c'est-à-dire l'ensemble de composants dont la définition ne dépend d'aucune dynamique extérieure ; ces composants

existent dès leur instanciation. Le niveau opérationnel, par contre, implique la création et la gestion à l'exécution des composants éventuellement créés au niveau déclaratif ; la manipulation de ces derniers peut avoir un impact sur leur contenu et leur comportement. Pour maintenir côté à côté ces deux niveaux, le modèle les combine grâce à une spécification à base d'événements, de sorte que la définition déclarative des composants peut être augmentée par le biais d'événements, créés d'une manière déclarative, impliquant des traitements à l'exécution.

3.2 Modèle d'annotations

Le modèle d'annotation dans CHM [78] tient ses origines de la conceptualisation très forte introduite par le modèle *Cinelab*³⁴ qui est une évolution du projet Advène pour l'échange de données avec différents partenaires. Notre concept d'annotation est donc conforme à ce modèle et par souci de généricité, on se restreint à un schéma de description très simplifié mais extensible. Le modèle Cinelab a été déjà présenté (voir § 2.2.1.4).

Concept d'annotation : Le concept d'annotations dans CHM englobe toute *association* d'une nouvelle donnée/information (méta-data, ressource, vue, lien, etc.) à une vidéo (ou un corpus) dans le but de l'enrichir, l'expliquer, la structurer, lui définir des liens ou pour des besoins de visualisation.

Conformément à Cinelab, l'annotation est définie comme étant une donnée associée à un segment logique de la vidéo appelé *fragment* [4] qui lui permet d'avoir un intervalle de validité, une portée définie par des *timecodes* - instants de début et de fin - par référence au temps de la vidéo. Par exemple, une annotation définissant une ancre spatio-temporelle adresse un fragment qui couvre son intervalle d'activation ; un lien et son ancre source sont définis par des annotations qui adressent le même fragment.

Une annotation est constituée d'un contenu, attaché à un fragment d'un média audiovisuel. Ainsi, les attributs d'une annotation incluent son *type*, sa *référence média*, ses *timecodes début/fin* et son *contenu*.

Hypervidéo à base d'annotation : Une hypervidéo CHM présente une information audiovisuelle à laquelle est associée une structure d'annotations utilisée pour générer des présentations riches par des mécanismes de structuration, d'interaction, de représentation, de mise en forme, etc. L'information enrichissante véhiculée par les annotations est un ensemble de données pouvant être présentées suivant des schémas de placement et de

34. <http://liris.cnrs.fr/advène/cinelab.html>

synchronisation et reliées suivant la logique de l'auteur au média principal. Le contenu d'enrichissement ajouté sur le flux audiovisuel peut être de nature diverse. Ainsi, tout contenu supplémentaire et toute information nouvelle, présentés dans le document ou référencés par celui-ci, sont considérés comme des annotations ou des contenus d'annotations qui ajoutent de la connaissance à l'information primaire.

La nature basée sur les annotations du modèle CHM permet de séparer le contenu du document de sa forme et son contexte de présentation. Le but de la composition multimédia est de rendre compte et/ou d'enrichir certaines caractéristiques de l'information principale. Un document hypervidéo n'est finalement qu'une vue enrichie des informations contenues dans le corpus de vidéos annotées.

Création d'annotations : Selon les besoins de l'utilisateur, le modèle d'annotation proposé peut être facilement étendu ou spécialisé et adapté à des exigences spécifiques. Dans le présent travail, l'édition vidéo et l'extraction de caractéristique (*feature extraction*) n'est pas au cœur du modèle et le processus d'annotation peut être réalisé manuellement ou par des systèmes dédiés tels que Advene [4] ou Anvil [46].

3.3 Modèle logique à base de composants

Le modèle logique dans CHM introduit un formalisme à base de composants [77] pour structurer le document suivant plusieurs niveaux hiérarchiques. Un ensemble - non exhaustif - de concepts hypervidéos est défini par cette abstraction logique à plusieurs stades de représentation. Pour permettre des implémentations libres ou ciblées et moins restreintes suivant des logiques propres aux systèmes hôtes et aux environnements cibles, le modèle ne fournit qu'une description de haut niveau (*high level description*) des composants. Des spécialisations plus techniques seront introduites ultérieurement pour le besoin d'illustrer l'implémentation des concepts, présentée au prochain chapitre.

Description Générale du Modèle : Un schéma général de la structure du modèle est représenté sur la figure 3.1. Une hypervidéo est construite à partir de composants de bas et haut niveaux qui sont les briques qui représentent l'information formelle et les unités de composition. Ces composants sont reliés au flux audiovisuel principal continu par le biais d'un type particulier de composants, les *TimedMedia*. Le rendu de ces composants permet la définition d'une *référence de temps*, une horloge abstraite inférée à partir de la présentation du temps de la vidéo et utilisée pour synchroniser les composants du document. Les données présentées par un composant sont récupérées principalement à

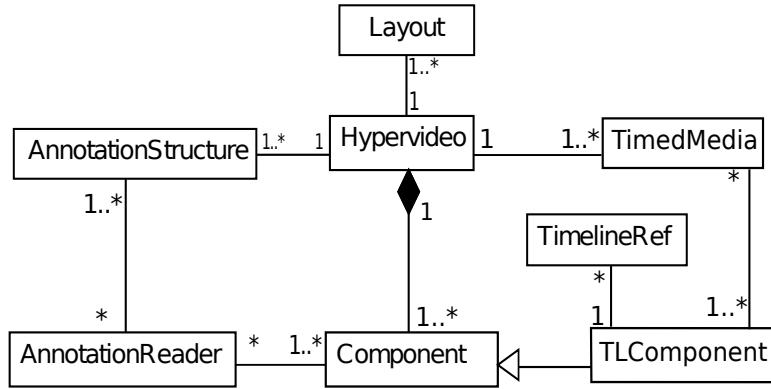


Figure 3.1 – Vue générale du modèle logique de CHM

partir d'un ensemble de structures d'annotation, accédées à travers des composants de lecture d'annotations.

3.3.1 Composants de base (CHM Core Components)

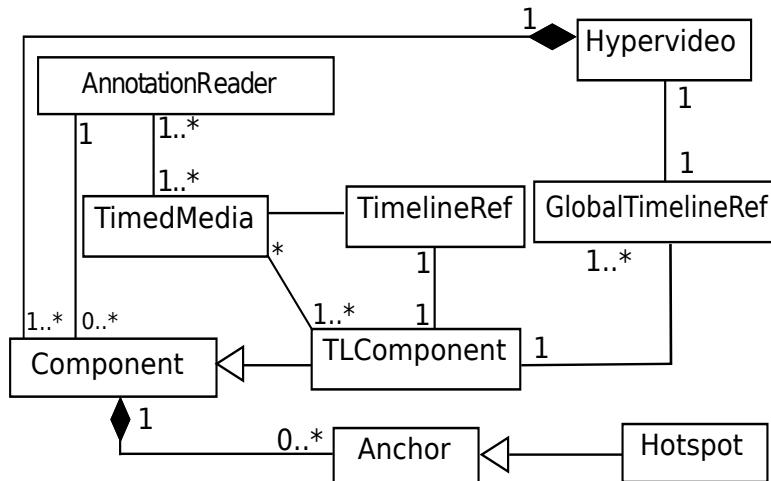


Figure 3.2 – Les composants de base de CHM

La représentation d'une hypervidéo exprimée par le formalisme CHM est obtenue par le biais de la spécialisation et la composition à plusieurs niveaux de représentation du composant de base **Component**, l'entité élémentaire principale permettant de définir l'hypervidéo, représenté sur la figure 3.2. Une hypervidéo dont l'élément racine est **Hypervideo** est formée d'un ensemble de composants temporalisés et non temporalisés, présentés lors de son exécution sur une interface physique de rendu. Pouvant avoir un intitulé (attribut *title*) et éventuellement une description (attribut *meta*), elle fait référence à au moins un lecteur d'annotations, exprimé par l'élément **AnnotationReader** et qui permet d'ac-

céder à une structure d'annotations et de renvoyer les données en réponse à une requête spécifique. Les annotations peuvent provenir d'une structure explicite définie au sein du document, fourni par un identificateur, ou à une ressource prétraitée accédée via une URL (`href`). Les ancrès et *hotspot* seront présentées ultérieurement. L'ensemble minimal des attributs d'une hypervidéo et de ses lecteurs d'annotation sont décrits très génériquement dans le tableau 3.1.

Elément	Attributs	Définition
<code>hypervideo</code>	<code>title</code>	titre du document hypervidéo
	<code>meta</code>	chaîne de caractères décrivant le document.
<code>AnnotationReader</code>	<code>id</code>	identificateur unique du lecteur d'annotations.
	<code>href</code>	URL ou identificateur de la structure d'annotations

Tableau 3.1 – Éléments CHM de base : Hypervidéo et lecteurs d'annotation

Elément	Attributs	Définition
<code>component</code> (et tous les composants qui en héritent)	<code>id</code>	identificateur unique du composant pouvant être assigné par l'auteur ou par le système.
	<code>src</code>	identificateur de la source de données (lecteur d'annotations)
	<code>filter</code>	chaîne de caractères permettant d'interroger la source de données.
	<code>volume</code>	valeur absolue ou relative fixant le niveau du son du composant audible.
<code>VisualComponent</code> (et tous les composants qui en héritent)	<i>(TL/)</i> <i>Component Attributes</i>	attributs hérités de <code>Component</code> ou de <code>TLComponent</code> si <code>timelineRef</code> est défini (voir§ 3.5.1).
	<i>Spatial Attributes</i>	attributs spatiaux du composant.
	<code>style</code>	style de présentation du composants.
	<code>anchors</code>	tableaux d'identificateurs ancrès/hotspots.

Tableau 3.2 – Éléments CHM de base : les composants atomiques

Composants de rendu atomiques : l'élément `Component`, un composant atomique est un élément de rendu générique présentant un contenu dont la source de donnée est renseignée par l'auteur ou extraite à partir de la structure d'annotations. Un composant comporte un identificateur unique au sein du document. Le contenu de cet élément peut être audible et du fait, des attributs sonores (volume) peuvent être spécifiés.

Le contenu avec un rendu visuel est présenté via les éléments `VisualComponent`. Un tel composant a des manifestations visuelles explicites et son message principal est véhiculé par son apparence sur l'interface utilisateur. Les propriétés de ces composants sont résumés dans le tableau 3.2.

Composants Temporalisés et de Temporalisation : Une hypervidéo étant un document multimédia à base audiovisuelle, le temps et la synchronisation de sa présentation sont définis par rapport à un flux audiovisuel principal. Ainsi, une hypervidéo fait référence à au moins un média audiovisuel accédé à travers un élément **TimedMedia** qui est un composant abstrait se référant à un flux temporelisé (audio ou vidéo) et pouvant être présenté par un composant de rendu.

Un composant **TimedMedia** a une durée intrinsèque qui, lors de sa présentation par un composant *player*, ajoute des capacités temporelles au document, exprimée par une référence virtuelle appelée *TimeLine Reference* (TLR) (élément **TimelineRef**) et qui s'occupe de la synchronisation des objets de rendu reliés à l'élément **TimedMedia** en cours de présentation. Plusieurs lecteurs, et du fait, plusieurs TLRs peuvent co-exister, définissant des sous-documents hypervidéos différents.

Les composants reliés à des TLR sont des éléments spécialisés appelés **TLComponent** contraints temporellement. Les composants **TimedMedia** sont des **TLComponent** spécifiques permettant de déclarer une référence de temps *TLR*. Les composants indépendants du temps peuvent être utilisés comme conteneurs ou pour des besoins d'illustration statique. Un élément **Component** générique d'une hypervidéo peut ne pas être lié à aucune TLR et dans ce cas, il est dit indépendant ou insensible au temps (*time-independent* ou *time-insensitive*).

Un élément global de synchronisation, **GlobalTimelineRef**, permet la synchronisation entre différentes TLRs et la définition d'une source de synchronisation pour des composants temporalisés mais non liés à un **TimedMedia** spécifique mais plutôt à tout le document. Quelques attributs de ces composants sont définis dans le tableau 3.3.

Récapitulatif : Les composants de rendu de base sont des conteneurs de données abstraits de plus bas niveau. Ils définissent des classes très génériques permettant de concevoir le document dans un langage simple et pertinent.

Une hypervidéo, suivant ce formalisme, est constituée d'un **TimedMedia** qui véhicule un flux audiovisuel principal et qui permet d'inférer un référentiel temporel au document. La structure et le contenu du document est défini par interrogation des structures d'annotations qui permettent de définir l'allure et le comportement des éléments définis. Dans le document, la plupart des éléments présents sont liés temporellement au flux, leurs instants de début et de fin étant inférés par la portée temporelle des fragments désignés par les annotations impliquées. Ces éléments sont détenus par des composants **TLComponent**. L'auteur du document peut aussi définir des objets non liés directement aux annotations ou détachées de celles-ci, par exemple un menu global, une zone de commentaires ou un

Elément	Attributs	Définition
TLComponent	Component Attributes	attributs hérités de Component.
	Temporal Attributes timelinref	attributs temporels du composant. identificateur de la référence de temps TimelineRef.
TimedMedia	(TL/)Component Attributes	attributs hérités de Component ou de TLComponent si timelineRef est défini.
TimelineRef	timedmediaId	identificateur du TimedMedia.
	position	position courante de l'horloge locale du TimedMedia utilisé.
	status	état de lecture du TimedMedia. Valeur possible : play, pause et stop.
	duration	durée complète nécessaire pour la présentation du flux.
GlobalTimelineRef	position	position courante de l'horloge globale correspondant au début de la présentation
	status	état de lecture de la présentation. Valeur possible : play, pause et stop.

Tableau 3.3 – Éléments CHM de base : les composants temporalisés et de temporalisation

lecteur rss. Ces objets sont atemporels et leur contenu est défini par des *Components* génériques. A un instant donné, plusieurs *TLR* peuvent être définis impliquant plusieurs sous document hypervidéos. La synchronisation entre les différentes lignes de temps est réalisée par rapport à la référence globale *GlobalTimelineRef*.

3.3.2 Composants d’interface simples (CHM Plain Components)

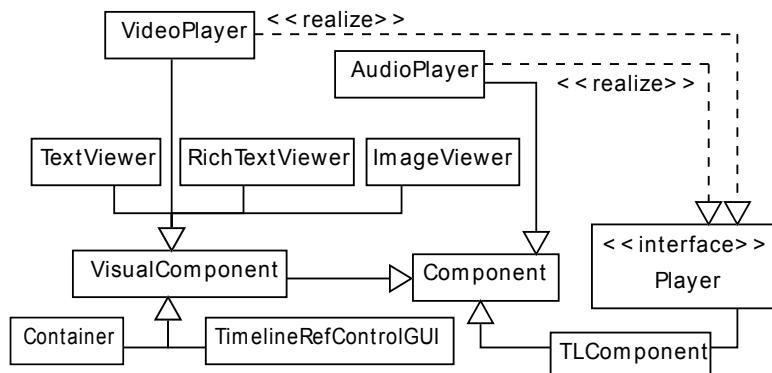


Figure 3.3 – Composants d’interface simples de CHM

Les composants d’interface (ou composants simples - *plain components*-) sont des éléments de niveau intermédiaire que l’on pourrait appeler niveau hypermédia/multimédia pouvant être utilisés et instrumentés directement par l’auteur car utilisables comme tels dans le document. Ils offrent un cadre abstrait suffisant pour l’édition hypervidéo, voire hypermédia. Néanmoins, l’intérêt principal de ces éléments est leur utilisation comme

briques pour dériver des composants hypervidéos plus spécifiques. Ainsi, étant eux-mêmes basés sur les composants de base qu'ils spécialisent, les *plain components* peuvent être utilisés pour former des éléments plus complexes.

La figure 3.3 présente ces composants simples. Les composants de rendu spécifiques synchronisés offrent des interfaces interactives pour la présentation de données temporelles, définies par des annotations. Des lecteurs de données, appelés *AnnotationReaders* (voir § 3.3.4) sont initialisés pour renvoyer un contenu à partir des structures de données et d'annotations. Ces structures peuvent être fournies par l'auteur ou extraites automatiquement par le système de rendu durant la présentation.

Remarques :

- Les composants de rendu de ce niveau sont des interfaces spécialisées pour recevoir et renvoyer des données de certains types. De ce fait, nous nous contenterons de décrire ces composants ; les attributs et méthodes spécifiques à ces composants sont déterminés par le type d'implémentation envisagé et ne sont pas considérés pertinents pour la compréhension du modèle. Les attributs qu'une implémentation minimale requiert sont déjà définis au niveau des composants de bas niveau
- La représentation des ressources médias en elle-même n'est pas pertinente pour le modèle et ne le renforce pas spécialement. Une ressource est accédée par un composant via une spécification rendue par le lecteur d'annotations. Par exemple, un serveur de ressources peut être défini pour récupérer le contenu des composants. Au niveau d'abstraction sur lequel nous nous plaçons, toute donnée est supposée assez bien renseignée par les structures et les lecteurs d'annotations.

Conteneurs statiques : Un conteneur statique, représenté par l'élément *Container*, est un composant offrant une interface graphique pour comprendre d'autres composants. Un conteneur ne peut recevoir des données ou interroger des lecteurs d'annotation. Il permet de simplifier le regroupement spatial et/ou sémantique de plusieurs composants et d'unifier leur traitement comme l'application d'un style de présentation commun.

Interfaces de rendu des médias : Le rendu du contenu multimédia d'un document requiert des visionneuses et lecteurs spécifiques avec interfaces interactives. Le contenu présenté est accédé par l'interrogation de lecteurs de données (lecteurs d'annotations).

La présentation de contenu via des visionneuses et des lecteurs permet un traitement homogène de tous les types de média manipulés, qu'ils soient visuels ou non, discrets ou continus. Pour les éléments n'ayant pas à priori de représentation physique (par exemple

une piste audio), la visionneuse fera le rendu audio et présentera une interface simple si l'implémentation le prévoit et si l'utilisateur le spécifie. Pour les média discrets, n'ayant pas de durée intrinsèque, une durée infinie leur est affectée par défaut.

Interfaces de rendu des flux audiovisuels : Les éléments `Player` sont des composants génériques définis dans le but de présenter un flux de données continu (ayant une durée intrinsèque). Pour représenter un média, un fragment ou un montage (défini par la mise bout à bout de fragments), la lecteur (*player*) fait le rendu du média avec un ensemble de contrôles et éventuellement une *timeline* (comme dans le cas classique des lecteurs audio). Plus spécialisés, les éléments `AudioPlayer` et `VideoPlayer` définissent des interfaces adaptées aux médias audibles uniquement et audiovisuels respectivement.

Un composant `AudioPlayer` permet la restitution de la piste audio d'un fichier multimédia (son, vidéo,...). L'interface de sortie de flux audio de l'`AudioPlayer` produit les buffers audio qui seront envoyés à la carte son.

Le lecteur vidéo (`VideoPlayer`) est un composant essentiel du document audiovisuel qui assure le rendu riche et personnalisable des données de flux continu. Alors que le *player* est une visionneuse minimale de flux, l'interface de rendu vidéo est plus riche et est adaptée à présenter le flux principal contenu dans le *TimedMedia*.

Interfaces de rendu des médias statiques : Les visionneuses et les lecteurs de contenu statique permettent de présenter des média n'ayant pas de durée intrinsèque, comme le texte et les graphiques.

Un lecteur de texte est représenté par l'élément `TextViewer`. Il permet de présenter des portions textuelles et d'appliquer un style d'affichage propre au composant. Parmi ses fonctionnalités, il permet la sélection et la copie de passages spécifiques.

La visionneuse graphique représentée par l'élément `ImageViewer` est spécialisée dans le rendu des images et autres graphiques selon des spécifications propres au composants (la taille, les palettes de couleurs, etc.).

Un contenu textuel plus complexe peut être présenté par le composant `RichTextViewer` qui est défini de telle manière à prendre en compte un certain nombre de types de document orienté XML avec définition d'un namespace et de feuilles de style tels que les documents HTML, RSS, XML+CSS (ou XSL), etc. Par exemple, un contenu synchronisé peut être lu à partir d'un lecteur de données ayant pour source l'encyclopédie en ligne *Wikipédia* et rendu dans un composant `RichTextViewer`.

Un composant plus spécifique au média principal est `TimelineRefControlGUI`. Ce dernier permet de représenter d'une manière graphique une référence de temps (TLR)

permettant de la contrôler et d'interagir avec elle.

3.3.3 Composants de haut niveau (CHM High Level Components)

Les composants de base ayant servi pour l'introduction de composants de rendu simple, ces derniers sont utilisés pour la spécialisation et la génération d'éléments de haut niveau qui regroupent les artefacts les plus usuels pour la présentation hypervidéo. Ces composants constituent un sorte de langage de description du niveau *lecteur* qui adapte les artefacts de présentation, représentation et navigation que l'on retrouve dans la plupart des documents hypermédias, tels que les menus et les tables de matière. Ces composants regroupent aussi des paradigmes bien propres aux hypervidéos tels que les *hotspots*, les *sous-titres* et les superpositions que l'on désignera par le terme *overlays*.

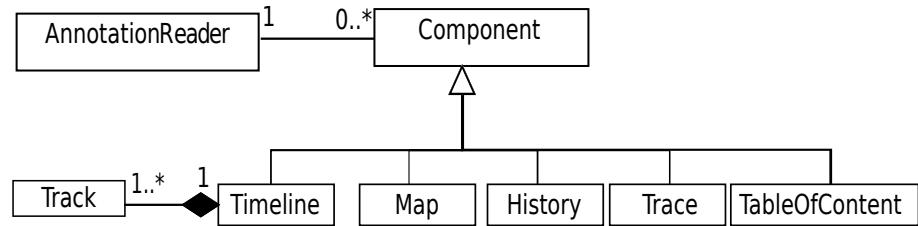


Figure 3.4 – Composants CHM de haut niveau

L'ensemble de ces composants sont présentés à la figure 3.4. Sans être exhaustif, cet ensemble est extensible et permet de simplifier les tâches de modélisation et de programmation lors de son implémentation. Quand un nouveau composant est nécessaire, il peut être construit de ceux déjà existants.

Historique et Parcours de lecture : Des graphes de navigation complexes sont induits par le nombre de possibilités de parcours du document hypervidéo, principalement de par sa composante audiovisuelle continue, résultant en une surcharge cognitive importante pouvant causer la désorientation du lecteur. Le mécanisme de retour en arrière implémenté par la plupart des navigateurs est une fonctionnalité intéressante devenue l'une des plus indispensables pour la navigation Web. De telles fonctionnalités n'existent pas dans la plupart des systèmes hypervidéos actuels [39]. L'idée d'un historique de navigation est étroitement liée à celle du contexte. En effet, partir vers un point du parcours déjà visité suppose que l'utilisateur retrouve la configuration exacte de la présentation à l'instant correspondant. Pour ce faire, l'élément *context* (voir § 3.5.1) est utilisé pour la sauvegarde et la restitution d'un contexte déjà visité. Lors d'une restitution, nous pouvons distinguer deux cas :

1. La présentation redémarre à partir du point sauvegardé et continue une lecture linéaire jusqu'à ce qu'un événement vient modifier le chemin narratif sauvegardé. Ce comportement ressemble à l'historique de navigation classique et est utilisé par l'élément **history**.
2. La présentation redémarre à partir du point sauvegardé et continue en réempruntant le chemin narratif qui a été suivi avant la restitution de contexte en simulant les mêmes événements et interactions. Cela introduit la notion de trace utilisée par l'élément **trace**.

Les attributs de ces composants sont décrits sur le tableau 3.4.

Elément	Attributs	Définition
History	<i>VisualComponent Attributes</i>	attributs hérités de <i>VisualComponent</i> .
	numberOfContextToSave	nombre de contexte à sauvegarder.
	Contexts	pile de contextes sauvegardés.
Trace	<i>VisualComponent Attributes</i>	attributs hérités de <i>VisualComponent</i> .
	numberOfContextToSave	nombre de contexte à sauvegarder.
	Contexts	pile de contextes sauvegardés.
	CurrentContext	indice du contexte courant.

Tableau 3.4 – Composants de Haut Niveau : Historique et Trace

Historique de Navigation : Le composant introduit par l'élément **History** permet d'enregistrer l'évolution de la présentation et des interactions de l'utilisateur et de sauvegarder le chemin narratif induit par le parcours de lecture suivi. Des points de sauvegarde et de récupération sont automatiquement enregistrés par la lecture du contexte approprié. Présenté graphiquement à l'utilisateur, ce dernier peut restaurer un état antérieur de la présentation et construire un nouvel historique.

Trace de lecture : La trace de lecture, renseigné par un composant **Trace**, permet d'enregistrer tout le chemin narratif, exactement comme c'est fait avec le composant *history*. La différence avec ce composant est que, lors de la restitution, le moteur hypervidéo se charge de suivre le même chemin que celui emprunté lors du dernier passage, en dehors de toute intervention utilisateur, impliquant une lecture plutôt cinématographique. Le lecteur peut interrompre cette re-lecture par une interaction qui changerait le chemin narratif sauvegardé et annulera les points non encore revisités.

Indexes de Navigation : Les composants d'index de navigation tels que les tables de matière permettent de guider le lecteur en lui offrant des entrées lui révélant la structure de l'hypervidéo, élément important pour prévenir la désorientation tout en permettant à

l'auteur de structurer son récit en définissant des raccourcis pour munir le lecteur d'un moyen d'explorer l'hyperespace d'une manière guidée.

Table de Matières : Un composant `TableOfContent` (ToC) définit une table de contenu interactive de l'hypervidéo. Elle révèle la structure du document suivant des paramètres particuliers exprimés en termes d'annotations ou de types d'annotations. Plusieurs tables peuvent être définies, présentées et représentées d'une manière plate ou suivant un arbre hiérarchique.

Plan : Le plan (ou la cartographie) d'un document hypervidéo, exprimé par un composant `Map`, est une représentation succincte ou un aperçu général du contenu donnant accès direct à un certain point de la progression narrative du document par un mécanisme de branchement vers un point de vue particulier. Alors que les ToCs peuvent être vues comme spécialisation des maps axant sur la structure, les maps peuvent être vues comme des résumés exprimés par exemple par des clichés (*snapshots*) ou des descriptions textuelles des cibles des entrées. Les maps peuvent être construites à partir de composants textuels ou graphiques, voire même audiovisuels. Plusieurs maps peuvent être définies dans le document pour illustrer les différentes caractéristiques.

Quelques attributs de ces composants d'index sont présentés sur le tableau 3.5.

Elément	Attributs	Définition
<code>TableOfContent</code>	<i>Container Attributes</i>	attributs hérités de <code>Container</code> .
	<code>title</code>	titre de la table de matière.
	<code>filter</code>	filtre complexe d'interrogation de la structure d'annotations permettant la génération de plusieurs niveaux.
	<code>entries</code>	liste ordonnées de <code>TextViewer</code> contenant chacun une entrée affichée textuellement.
<code>Map</code>	<i>Container Attributes</i>	attributs hérités de <code>Container</code> .
	<code>title</code>	Titre du plan.
	<code>filter</code>	filtre complexe d'interrogation de la structure d'annotations permettant la génération de plusieurs niveaux.
	<code>entries</code>	liste ordonnées de <code>VisualComponent</code> contenant chacun une entrée.

Tableau 3.5 – Composants de Haut Niveau : Index de navigation

Les lignes de Temps : Une ligne de temps (composant `Timeline`) est une représentation graphique, chronologique et interactive du temps de l'hypervidéo permettant de représenter et organiser spatialement des caractéristiques particulières survenant au cours

du temps. Le timeline fournit un *slider* (glisseur) pour indiquer et agir sur le temps courant de la vidéo. Les timelines placent les éléments média, les événements importants ainsi que les liens sur un axe chronométré, sur différentes pistes ou *track*.

Un composant **Track** (piste) est la représentation temporelle des médias, indiquant la période active de l'annotation correspondante. L'axe du temps est représenté d'une manière relative puisque les instants de début et fin effectifs d'un piste peuvent ne pas être connus, par exemple quand ils sont basés sur des événements.

Le tableau 3.6 présentent quelques attributs génériques des lignes de temps et des pistes les composant.

Elément	Attributs	Définition
Timeline	<i>Container Attributes</i>	attributs hérités de Container.
	title	titre de la table de matière.
	filter	filtre complexe d'interrogation de la structure d'annotations permettant la génération de plusieurs niveaux.
	entries	liste ordonnée de Track contenant chacun une entrée.
Track	<i>Container Attributes</i>	attributs hérités de VisualComponent.
	src	source de données.
	filter	filtre d'interrogation de la structure de données.

Tableau 3.6 – Composants de haut niveau : ligne de temps

Sous-titres (*Sbtitles*), Légendes (*Captions*) et Overlays : L'utilisation des composants de haut niveau est très variée et peut déboucher sur d'autres artefacts de présentation très intéressants. Dans le cadre de l'hypervidéo, les overlays sur la fenêtre de la vidéo principale est un exemple typique de ces utilisations. Ainsi, pour améliorer l'accessibilité vidéo, plusieurs composants peuvent être utilisés. Les légendes textuelles *Text Captions* tels que sous-titres, traduction, commentaire, etc., des superpositions graphiques (*Graphic Overlays*) et d'autres *Multimedia Overlays* dont les données sont contenus d'annotations textuels ou multimédia peuvent être placés sur les lecteurs vidéo. Leurs conteneurs sont des instances de visionneuses et lecteurs ayant des spécifications spatiales appropriées. Le modèle CHM ne spécifie pas explicitement ces composants car ils peuvent être facilement instanciés à partir des composants simples qu'il définit et en plus, tout composant visuel peut être superposé par n'importe quel contenu multimédia à travers la définition adéquate de leur disposition spatiale.

3.3.4 Composants d'accès aux données (CHM Data Access Components)

Au niveau d'un composant, certaines données peuvent être extraites d'autres informations et ressources décrites par les structures d'annotations. L'accès à une donnée externe

n'est possible que par une interaction qui elle-même dépend de la création d'une liaison par interrogation de composants de communication spécifiques appelés *composants d'accès aux données*, présentés à la figure 3.5. Ces composants sont des *middlewares* avec interfaces fonctionnelles d'interrogation des structures de données (annotations et ressources) suivant une procédure unifiée.

L'élément `AnnotationReader` représente le gestionnaire d'accès aux données le plus standard. Composant virtuel, il n'est propre à aucun format de données et du fait, d'autres types de lecteurs sont introduits, héritant de ses caractéristiques en se spécialisant à des formats plus particuliers. Le modèle d'annotation que nous avons proposé peut être efficacement exprimé par le modèle d'Advène ; ainsi le format par défaut reconnu dans CHM est celui-ci et il peut être lu par le lecteur spécialisé `AdveneReader` ; ce lecteur suppose que les structures d'annotation ont été réalisées avec Advène et du fait, ses méthodes permettent de lire des données d'annotations à partir de ces structures explicites.

Le lecteur `JsonReader` se charge des structures d'annotation décrites sous le format JSON (JavaScript Object Notation). Ce dernier est un format textuel, générique, dérivé de la notation des objets du langage ECMAScript. Nous discuterons plus en détail de ce format lors de notre implémentation.

Beaucoup de formats existent pour la définition de texte temporalisé. L'accès à certains de ces formats peut utiliser le lecteur de données `TimedTextReader` qui permet d'interroger des ressources sauvegardé sous des formats *SubRip* (srt)³⁵, MPEG-4 Timed Text³⁶, SmilText³⁷ et W3C TTML (DFXP)³⁸.

D'autres lecteurs de divers formats peuvent être mis en œuvre. Le format de description multimédia MPEG-7³⁹ est utilisé par le lecteur `MPEG7Reader` alors que le lecteur `RDFReader` peut être utilisé pour ajouter des capacités de requête sémantique exprimées en RDF⁴⁰ sur les documents hypervidéos.

3.4 Dimension spatiale des hypervidéos CHM

Dans le modèle proposé, le contenu de rendu est représenté par l'aspect visuel des composants, représenté par une boîte-conteneur. Ce conteneur peut être rapproché du concept de *Channel* dans *AHM* ou de *Region* dans *SMIL*. La description du modèle est

35. <http://zuggy.wz.cz/>

36. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39478

37. <http://www.w3.org/TR/smil/smil-text.html>

38. <http://www.w3.org/TR/ttaf1-dfxp/>

39. <http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>

40. <http://www.w3.org/TR/rdf-primer/>

3.4. Dimension spatiale des hypervidéos CHM

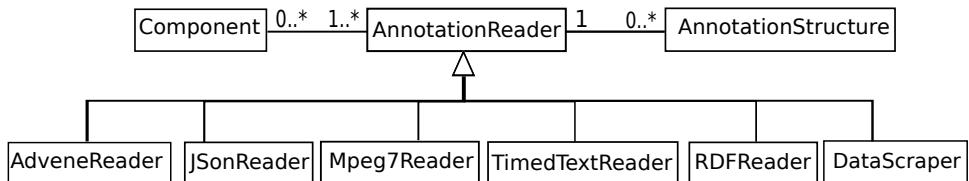


Figure 3.5 – Composants CHM d'accès aux données

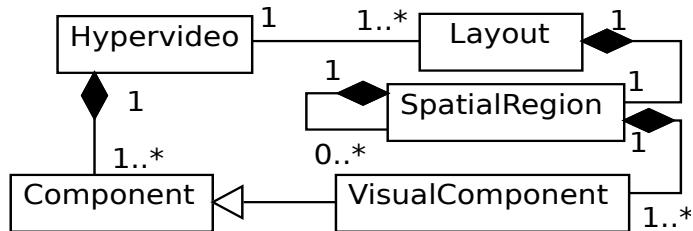


Figure 3.6 – Model spatial des hypervidéos CHM

représentée à la figure 3.6. Faisant partie de la famille *des modèles absolus*, il s'articule sur la définition des éléments *layout*, *régions spatiales* et *éléments visuels*.

Comme toute application à interface utilisateur, le système permettant le rendu d'une présentation hypervidéo exécute le fichier sur une fenêtre de présentation. Le conteneur parent de la présentation est défini par l'élément **Layout** qui maintient la configuration spatiale du document. Il correspond à la fenêtre de présentation principale de l'hypervidéo et est automatiquement instancié à l'exécution, et sa destruction correspond à la terminaison de la lecture du document hypervidéo. Ce dernier peut contenir une ou plusieurs spécifications représentées par plusieurs éléments **Layout**. Une seule layout, et donc une seule configuration spatiale, peut être active à un moment donné.

Une *layout* est composée d'un ensemble de régions spatiales, exprimées par des éléments **SpatialRegion**. Une région spatiale peut contenir un ensemble de composants visuels **VisualComponent** et éventuellement d'autres régions spatiales.

Attribut	Signification
<i>top</i>	position supérieur du composant.
<i>bottom</i>	position inférieure du composant.
<i>left</i>	position gauche du composant.
<i>right</i>	position droite du composant.
<i>width</i>	largeur du composant. Cette largeur est par défaut celle du média graphique intrinsèque.
<i>height</i>	hauteur du composant. Cette hauteur est par défaut celle du média graphique intrinsèque.
<i>zIndex</i>	la position en profondeur du composant. Un composant recouvre tous les composants avec valeurs de <i>zIndex</i> inférieures à la sienne.
<i>spRelativeTo</i>	la référence de positionnement et de dimensionnement. Les valeurs possibles sont (<i>container</i> , <i>region</i> , <i>layout</i> - ou <i>absolute</i> -)

Tableau 3.7 – Attributs spatiaux

Le modèle définit un ensemble d’attributs, présenté sur le tableau 3.7, permettant de spécifier les coordonnées et les dimensions des objets visuels. Par défaut, un élément **VisualComponent** a les mêmes valeurs de dimensionnement que celles de son contenu. L’auteur peut lui assigner d’autres valeurs, d’une manière absolue ou relatives ; par exemple, spécifier 50% pour une hauteur veut dire que le composant va occuper 50% de la hauteur de son conteneur ou de l’élément de référence renseigné par l’attribut *spRelativeTo*. Ce dernier attribut détermine aussi la référence de positionnement dont les valeurs peuvent elles aussi être relatives ou absolues. La référence peut être le conteneur du composant s’il est contenu par un autre composant visuel, la région spatiale du composant ou toute la *layout*, ce qui revient à une spécification absolue. L’attribut *zIndex* détermine le degré de profondeur du composant, permettant de mieux gérer les chevauchements et les *overlays*.

3.5 Dimension temporelle des hypervidéos CHM

La spécification temporelle d’un document multimédia permet de définir l’ordre temporel de ses composants. Dans le modèle proposé, cette spécification s’articule sur la notion bien connue de *ligne de temps* adaptée au contexte hypervidéo et utilisant une échelle temporelle explicite comme référence d’activation commune. La synchronisation, quant à elle, définit l’occurrence d’événements simultanés. Le modèle temporel proposé s’articule sur la notion bien connue des *lignes de temps*.

3.5.1 Paradigme TimeLine Reference (TLR)

Dans CHM, tout objet actif véhicule une information relative au contenu audiovisuel principal le situant sur un point courant de sa ligne du temps. En d’autres termes, la cadence de présentation hypervidéo est principalement dictée par le temps et les évolutions de ce média principal qui assujettit l’évolution de tout objet média qui lui est associé.

Définition : Une référence de temps, dénotée par TLR pour *TimeLine Reference*, est un référentiel temporel virtuel attaché à la composant audiovisuelle annotée, pouvant être issu d’un montage à partir d’un corpus annoté, afin d’ordonnancer tous les événements temporels pouvant exister dans un document hypervidéo et inférer à ses composants des capacités temporelles. Les composants temporalisés du document sont activés et désactivés par référence à cette TLR. Plusieurs références de temps et au moins une peuvent exister dans un même document.

Propriétés de la TLR : Les caractéristiques des références de temps peuvent être résumées en un ensemble de points :

- Les valeurs d'horloge de la référence de temps reflètent l'évolution du flux principal.
- Par défaut, une synchronisation fine est assurée entre la référence et chaque objet lui reliant. Toute opération d'avance ou recul dans la référence a un impact sur le temps des composants temporalisés.
- Un composant peut changer de référence temporelle à tout moment.
- Une référence de temps peut avoir une autre pour référence de niveau supérieur.
- La terminaison d'une référence de temps implique la terminaison de tous les composants qui lui sont reliés.
- La terminaison de toutes les références implique la fin de la présentation.
- Les composants non temporels et ceux temporalisés par rapport à toute la présentation sont associés à une référence de temps particulière et unique dans le document appelée l'horloge global ou référence de temps global. Cette dernière est utilisée comme référence de toutes les TLRs définies sur des TimedMedia.

Attributs de la TLR : La référence de temps présente une interface permettant de l'interroger et de la contrôler. Elle manipule trois attributs clés : '*position*' et '*state*' et '*duration*'. *Position* renvoie le niveau de progression de la présentation hypervidéo, *State* indique si la présentation du document est temporellement en progression, mise à pause ou arrêtée. La durée, contenue dans l'attribut *duration* est en lecture seule et renseigne sur la durée complète de l'hypervidéo. Cette durée peut être infinie si la terminaison de la présentation est tributaire d'un événement extérieur (action utilisateur).

Notion de Contexte de présentation : La notion de référence de temps est utilisée pour définir le contexte d'une présentation. Un contexte est observé à des instants précis de la présentation correspondant à l'occurrence d'événements particuliers (événements utilisateur, les dynamiques du temps, navigation, interaction, etc.).

Le contexte d'une présentation hypervidéo est défini à tout instant par l'ensemble des informations qui concernent les composants hypervidéos et les données qui sont définies par et pour ces composants. Ainsi, il regroupe l'état et la position de la référence du temps, la configuration spatiale du document, le style appliqué, tous les attributs courants de tous les composants. Un contexte permet de prendre une photo instantanée (*snapshot*) de l'hypervidéo et de la sauvegarder. Il est observé à l'occurrence d'un événement particulier pouvant correspondre aux actions de navigation (hyperliens), aux actions de contrôle de type lecture/mise à pause/arrêt de la présentation et de ses composants ainsi qu'aux

différentes opérations de l'ordonnanceur.

3.5.2 Attributs temporels et de synchronisation

Les attributs présentés sur le tableau 3.8 permettent de préciser le comportement des composants temporalisés par rapport à la référence de temps ; si la référence n'est pas définie, la ligne de temps globale est considérée. La portée temporelle d'un média peut être

Attribut	Signification
<i>begin</i>	instant de début de la présentation du composant.
<i>end</i>	instant de la fin de la présentation du composant.
<i>duration</i>	durée de la présentation du composant.
<i>loopcount</i>	nombre de boucles de lecture pour un média avec temps intrinsèque.
<i>FineSync</i>	type de synchronisation avec la TLR (<i>soft</i> ou <i>hard</i>).
<i>CanSlideTLR</i>	cet attribut permet ou interdit à un élément d'influencer la TLR.

Tableau 3.8 – Attributs temporels

définie à travers un estampillage absolu, obtenu directement de la structure d'annotations, spécifié par l'utilisateur ou résultant d'événements qui permettent à l'utilisateur de définir des relations entre les médias.

La durée de présentation est une information redondante pouvant être inférée des instants de début et de la fin s'ils sont définis. Par défaut, elle a la valeur de la durée intrinsèque du média continu ou une durée infinie pour un contenu média statique. Le durée effective de lecture qui équivaut à la durée intrinsèque de média contenu s'il est continu, multipliée par le nombre de boucles (*loopcount*) ne peut dépasser la durée d'affichage.

3.5.3 Synchronisation multimédia

La synchronisation des médias peut être fine (*hard*) ou pas fine (*soft*). La synchronisation *hard* force les média à maintenir un haut niveau de synchronisation avec la référence de temps. Dans ce cas, l'état de lecture du composant est toujours égal à celui de la TLR et à chaque position temporelle de la référence de temps TLR, une seule valeur est associée au temps de présentation du composant. Par exemple, si l'élément est une vidéo et qu'à un moment donné, l'état de la TLR est à pause, la vidéo aussi se met à pause sans que son état ne soit modifié (le temps s'est arrêté). Une synchronisation *soft* permet au média de glisser temporellement. Un composant peut être par exemple mis à pause par l'utilisateur alors que le reste de la présentation est en progression. Le seul événement pouvant survenir est sa terminaison, à l'arrivée à échéance de sa durée de vie. L'attribut *CanSlideTLR* indique si le composant peut influencer la référence de temps.

Dans l'exemple précédent, si la vidéo progresse dans le temps, la TLR et le tout le reste va la suivre. La synchronisation, quand elle est *soft*, n'est maintenue que pour des instants précis correspondant au début et à la fin de la lecture du média. Durant l'intervalle de lecture d'un média, la synchronisation n'est pas maintenue, inférant une grande flexibilité de la présentation. Dans tous les cas et abstraction faite du type de synchronisation, l'arrêt ou la mise à pause de la référence de temps introduit automatiquement l'arrêt ou la mise à pause de tous les composants liés.

3.5.4 Gestion de la dé-synchronisation

La dé-synchronisation des médias d'un document est un scénario très fréquent dans les présentations multimédia et peut être due entre autres à l'accès distant à une ressource où à un réseau instable. Avec CHM, la gestion de telles situations tient compte de l'élément qui est dé-synchronisé.

Vidéo principale dé-synchronisée : Avec l'importance du composant audiovisuel principal, toute modification de son état et de sa position est reflétée sur le reste des composants du document. Ainsi, la dé-synchronisation de cet élément entraîne la mise à pause de toute la présentation jusqu'à ce que le média principal soit resynchronisé.

Média secondaire dé-synchronisé : Lorsqu'un média secondaire du document est dé-synchronisé, la présentation du document continue. À chaque mise à jour du temps de la progression de la présentation, la référence tente de resynchroniser tous les objets qui ne sont plus à jour temporellement.

3.6 Dimension hypermédia et interactionnelle

3.6.1 Interactions dans les hypervidéos

La dynamité que confère la temporalité des éléments contenus aux documents multimédia implique une grande complexité de perception. De nouveaux besoins, comme un retour sur un média particulier, un détail passé ou un saut vers un instant donné, doivent être considérés pour maîtriser cette complexité et améliorer la lecture du document [54]. La prise en charge de l'interaction utilisateur permet de libérer l'utilisateur de la lecture rigide tout en assurant au concepteur l'édition de documents riches et complexes.

Le modèle CHM définit les interactions de l'utilisateur suivant deux types : les *fonctions de manipulations* et les *fonctions de navigation*. Les fonctions de manipulation comprennent l'utilisation des contrôles offerts pour les composants de rendu tels que la lecture et l'arrêt. Ce type d'interaction est géré soit par les méthodes définies par l'implémentation des composants (par exemple, les composants *player* sont munis de contrôles standards), soit par le mécanisme de gestion événementielle que nous présentons à la section 3.7. Les interactions de navigation sont spécifiées par un mécanisme de liens qui définit une connexion logique entre des composants de l'espace hypermédia.

3.6.2 Notion de lien

Alors que des modèles tels que Amsterdam définissent le lien comme composants, les propriétés du modèle CHM nous orientent à adopter la définition la plus généralement utilisée, associant les liens - *uni-directionnels*, *uni-soucres*, *uni-destinations* - à des ancrages. Ce choix a plusieurs motivations. La plus importante est relative au fait qu'un lien interne à une hypervidéo provoque toujours un décalage temporel de la présentation et que chaque ancre destination d'un lien correspond à un instant du composant cible associé à un fragment de donnée, étant contenu d'une annotation. Par conséquent, le lien n'est qu'un mécanisme permettant d'atteindre cet instant où le composant cible sera présenté. D'autre part, le lien ne peut avoir plusieurs destinations car la présentation ne peut faire un décalage temporel que vers un seul instant. Notons néanmoins qu'un concepteur peut vouloir définir à un lien une ancre hypervidéo en plus il requiert l'affichage d'une URL dans une nouvelle fenêtre séparée. Grâce au mécanisme de gestion événementielle, ceci est possible : comme nous le présenterons ultérieurement, il suffira d'ajouter à l'événement d'activation du lien une action qui permet d'afficher une autre destination sur une nouvelle fenêtre.

Comportement dynamique des liens : L'activation de liens dans une présentation hypervidéo peut servir pour spécifier des comportements divers, grâce à la gestion événementielle du document. Par exemple, une ancre peut être utilisée pour d'autres besoins de visualisation comme l'affichage d'une fenêtre pop-up au passage de la souris.

Types de liens : Définis sur les ancrages de composants, les liens peuvent être de deux types :

- *Liens hypertextuels* : Un tel lien est de type hyperlien classique qui permet d'adresser une ancre simple définie en espace. C'est le type de lien que l'on peut retrouver dans la plupart des systèmes hypermédias/hypertextuels.

- *Liens hypervidéos* : Un tel lien définit un lien reliant deux ancrages dont au moins une est une *hotspot* - portions spatio-temporelles de composants - définie sur un flux audiovisuel. Autrement dit, les liens hypervidéos sont définis en temps et en espace.

Cibles de Liens : La cible d'un lien, et par extension, un lien peut être *interne* ou *externe*. Un lien interne a pour destination une ancre au sein du document, un instant de la référence de temps ou tout point de l'espace hypermédia. L'activation d'un tel lien provoque la mise à jour de la position temporelle de la TLR. Un attribut de retour (*backbehav*) spécifie le comportement de la présentation à la fin de la lecture de la cible : mise à pause, retour au point de départ, arrêt de la lecture ou poursuite de la lecture.

Un lien externe a pour cible une ancre externe au document, exprimée par son URI. L'ancre de destination peut être présentée sur une nouvelle fenêtre ou remplacer le contenu de la présentation en cours. A l'activation d'un tel lien et quand la destination est jouée sur une nouvelle fenêtre, la présentation peut être mise à pause, continuer ou s'arrêter, selon la valeur de l'attribut *leavebehav*.

3.6.3 Ancres hypervidéos

L'ancre dans les hypermédias est une zone au sein d'un nœud pouvant être la source ou la destination d'un ou de plusieurs liens. Cette ancre peut faire référence au média dans son ensemble ou certaines parties du média.

Dans le cas de l'hypervidéo, la notion d'ancre classique n'est pas suffisante ; elle ne peut pas prendre en considération la continuité temporelle du contenu. Par exemple, un lien peut faire référence à un certain objet de la vidéo qui se déplace avec la progression de la lecture du flux (l'ancre peut disparaître et réapparaître à des instants différents, son emplacement peut changer avec le temps. Une telle ancre ne peut pas être caractérisée par une zone du nœud mais un ensemble de zones conjointement avec des instants précis pour chaque zone. Ceci nous amène à définir la notion de zone sensitive ou hotspot.

Une région sensitive (*hotspot, hotobj*) désigne l'espace au sein du nœud vidéo pouvant servir d'ancres de liens. Une région sensitive est caractérisée par sa forme géométrique, les intervalles de temps de son apparition et de son emplacement à chaque apparition successive (exemple d'objet en mouvement). Les liens sont donc définis sur ces régions, accompagnés par des informations spatiales et temporelles régissant leur apparition et leur activation ainsi que des mécanismes de matérialisation permettant à l'utilisateur d'en avoir conscience et d'interagir avec.

Une ancre peut être définie sur des objets non visuels, par exemple sur de l'audio. Une telle ancre ne peut servir d'ancre de départ sans passer par une matérialisation, lui permettant d'être représentées visuellement. Une ancre non représentée visuellement n'aura que des attributs temporels. Un résumé des attributs des ancras et hotspots est présenté sur le tableau 3.9.

Pour uniformiser le traitement des ancras hypertextuelles classiques, les ancras hypermédiás qui permettent d'adresser une partie d'un objet statique (partie d'une image dans HTML par exemple) et les ancras présentées comme régions sensitives ambulantes, nous définissons une structure générale de l'ancre ayant au moins les attributs suivants :

- identificateur de l'ancre ; il sera nécessaire pour désigner l'ancre comme cible, pour définir des traitements particuliers et pour avoir la description du contexte ;
- style d'affichage ;
- liste ordonnée de tuples d'une structure de données (que l'on désigne par *hotInstant*) ayant la forme suivante :
 - coordonnées spatiales de la zone par rapport au composant (*top*, *left*, *width*, *height* et éventuellement *right*, *bottom*) .
 - spécification temporelle par rapport au début de la présentation du composant (*begin*, *duration* et éventuellement *end*) .

Valeur Élément	Attributs	Définition
anchor	id	Identificateur unique de l'ancre.
	href	la cible du lien. Cet attribut peut avoir pour valeur un instant, un identificateur ou une URL externe. Par défaut, sa valeur est "#".
	target	fenêtre d'affichage de la cible du lien, à l'activation. Cet attribut a les mêmes caractéristiques que l'attribut <i>target</i> de l'ancre classique <a>. Ainsi, les valeurs possibles sont _self, _blank...
	leavebehav	comportement de la présentation à l'activation du lien quand ce lien est externe. Les valeurs possibles sont <i>stop</i> , <i>pause</i> , <i>continue</i> avec <i>continue</i> comme valeur par défaut.
	backbehav	comportement de la présentation à la fin de la présentation de la destination quand ce lien est interne. Les valeurs possibles sont <i>back</i> , <i>pause</i> , <i>continue</i> , <i>stop</i> avec <i>continue</i> comme valeur par défaut.
hotspot	[Anchor Attributes]	attributs définis pour toute ancre.
	style	style d'affichage de la zone sensible.
	tableau de <i>hotInstant</i>	liste ordonnée de tuple ayant les attributs temporels et spatiaux standards, relativement au composant visuel contenant la hotspot.

Tableau 3.9 – Ancreas et hotspots CHM

Les deux entités *ancre* et *hotspot* sont définis séparément bien que la première puisse

être exprimée par la deuxième étant donné qu'elle est un cas particulier de cette dernière.

3.7 Modèle à base d'événements

Le comportement dynamique d'une hypervidéo CHM est principalement géré par un mécanisme événementiel. Un événement peut être formellement défini comme étant une réalité physique observée paramétrée en espace et en temps [88]. L'événement est un concept classique utilisé dans divers modèles multimédia. Les formats de présentation tels que SMIL utilisent ce concept pour spécifier le comportement des présentations.

Les événements CHM correspondent à des incidents instantanés ou des changements de statut durant la présentation d'un document hypervidéo et dénotent des changements pouvant avoir de l'incidence sur le déroulement de la présentation. La spécification événementielle est exprimée par les composants **Event** et **Action**. Les observateurs d'événements (*events listeners*) sont chargés de détecter les occurrences des événements et d'invoquer les procédures de traitement appropriées, exprimées en termes d'*actions*. Une action peut être une instruction atomique qui agit sur le document ou un ensemble d'opérations qui peuvent déclencher d'autres événements et provoquer de nouvelles actions.

Valeur Élément	Attributs	Définition
event	id	identificateur de l'événement.
	src	identificateur du composant sur lequel l'observateur d'événements est instancié
	eventtype	type de l'événement.
	condition	conditions supplémentaires à vérifier.
	do	identificateurs des actions à exécuter.
action	id	identificateur de la routine de traitement
	eventids	identificateurs des événements.
	typeaction	type de l'action.
	param	instructions d'exécution.

Tableau 3.10 – Événements et actions de l'hypervidéo

3.7.1 Événements

Le modèle événementiel permet d'associer à tout événement un observateur d'événement qui déclenche les traitements associés. En plus des événements standards, l'auteur peut en définir d'autres en spécifiant un déclencheur et les traitements appropriés. Les événements multimédia/hypermédiyas standards sont considérés par le modèle et peuvent donc être utilisés directement dans les blocs de données :

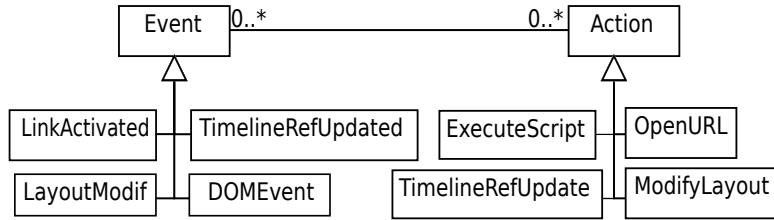


Figure 3.7 – Les événements et les actions dans une hypervidéo CHM

- *interactions utilisateur* : Ces événements sont générés suite aux opérations que l'utilisateur effectue sur le contexte du document, réalisées par le biais d'outils et de pointeurs tels que la souris et le clavier. Les interactions de type lire, mettre à pause, continuer, ajuster le volume et suivre un lien sont des exemples de cette classe d'événements.
- *événements intra-objet* : Ces événements sont reliés aux fonctionnalités internes des objets du document. Ils peuvent correspondre à un changement d'état de l'objet, par exemple la terminaison de la lecture ou la mise à pause d'un média.
- *événements inter-objets* : ces événements relient plusieurs objets du document et peuvent être utiles dans un modèle relationnels (spatial et temporel). Ainsi, la fin de la lecture d'un objet peut déclencher le début de la lecture d'un autre objet.
- *événements applicatifs* : Ces événements sont reliés à tout le document et peuvent correspondre par exemple à un changement dans l'état du document (chargement, lecture, terminaison) ou une de ses propriétés (temps local, taille d'affichage,...).
- *événements définis par l'utilisateur* : Ces événements sont définis par l'auteur du document et sont reliés à son état, à son contenu ou à ceux de ses composants (objets) à l'exécution.

La plupart de ces événements ne sont pas spécifiques aux hypervidéos et sont représentés dans CHM par les éléments **DOMEvent**. A ces derniers s'ajoute la modification d'un attribut quelconque du composant (spatial, temporel, contenu, changement de ligne de temps, etc) qui sont regroupés dans des classes plus spécifiques tels que ceux reliés à la référence de temps et aux liens hypervidéos. Ces événements et les actions possibles sont représentés sur la figure 3.7. D'autres événements propres aux hypervidéos CHM sont introduits par le modèle.

Les événements affectant directement la progression temporelle de la ligne de temps sont représentés par les éléments **TimelineRefUpdated**. Un changement de l'état de la TLR implique sa mise en lecture ou en pause ainsi que les événements de mise à jour du temps courant, communément appelés les *timeupdates* et qui correspondent à une nouvelle valeur du temps courant.

Les événements qui affectent la disposition spatiale (*layout*) correspondent à des changements dans la configuration de l'espace de rendu du document durant sa présentation et sont représentés par l'élément **LayoutModified**. Les exemples de ces événements incluent l'affichage ou la fermeture d'un composant visuel, le changement dans la position d'un composant ou d'une région spatiale.

Les événements de la navigation de l'utilisateur sont définis par les éléments **LinkActivated** et correspondent à une interaction ayant pour objectif d'aller vers un nœud interne ou externe.

3.7.2 Traitements et actions sur le document hypervidéo

Les événements que l'utilisateur définit doivent être pris en charge lors de leur déclenchement par les traitements appropriés. Le même événement peut être déclenché à différents niveaux et les mêmes traitements seront déclenchés. Un événement comporte donc la liste des actions à exécuter. L'action est un ensemble de traitements sur le contenu et les propriétés du document. Une action peut générer encore plusieurs autres événements qui impliquent d'autres actions.

La figure 3.7 illustre quelques types d'actions possibles. Les traitements sur un composant peuvent affecter directement ou indirectement son contenu. Ainsi, pour mettre à jour un composant (modifier un attribut, son contenu, son état d'affichage ou de lecture, changement de ligne de temps, etc.), l'élément **DomAction** est une action spécialisée qui a en entrée le composant concerné et en paramètres les attributs à modifier.

Les actions de modifications de la ligne de temps sont spécifiées par l'élément **TimelineRefUpdate**. Une telle action a pour objectif d'assigner de nouvelles valeurs à l'état de la TLR ou à sa position courante.

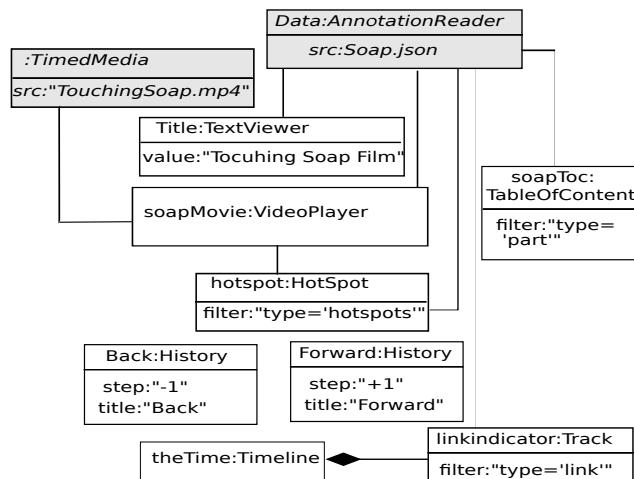
Les actions affectant la disposition spatiale du document sont définies par les éléments **ModifyLayout** qui ont en entrées les identificateurs de la *layout*, des régions spatiales ou des composants visuels à mettre à jour et en paramètres les nouvelles valeurs des attributs.

Les primitives permettant la navigation, c'est à dire d'accéder à une adresse URL suite au déclenchement d'un certain événement, peuvent être exprimées par le biais de l'élément **OpenURL** qui a en entrée l'adresse URL et en paramètres les attributs définis pour tout lien hypervidéo.

Enfin, l'utilisateur peut vouloir définir ses propres actions ou étendre celles qui sont prévues dans le modèle. Nous introduisons l'élément **ExecuteScript** qui permet de spécifier les actions en un langage qui peut être celui implémentant le modèle ou tout autre, que l'on appellera ici comme étant *langage de Script*.



(a) Exemple hypervidéo de Story of PI



(b) Modélisation de l'hypervidéo Story Of PI

 Figure 3.8 – Hypervidéo *Story Of PI*

Exemple de modélisation : Hypervidéo *Story of PI*

L'exemple de la figure 3.8(a), présenté dans [18], est une illustration d'un ensemble de concepts hypervidéos tels que les liens spatio-temporels et les tables de matière. Des boutons de navigation *back* et *forward* permettent d'avancer ou de reculer dans l'historique de la navigation. Les liens spatio-temporels permettent d'accéder directement à des parties de la vidéo.

La représentation formelle de cet exemple exprimée en format CHM est donnée à la figure 3.8(b). La structure d'annotations est capturée à partir du code HTML écrit par l'auteur par un composant *DataScraper*. La table de matière, représentée par le composant *soapToc* permet de parcourir le récit contenu dans la vidéo et d'accéder directement à des

parties spécifiques. Les ancrages et liens spatio-temporels sont représentés par des hotspots et des liens qu'ils contiennent. Les boutons de navigation sont des instances du composant *history* et sont utilisés pour restaurer les contextes enregistrés durant l'activité de lecture et de navigation. Une seule région spatiale, un seul *TimedMedia* et une seule référence de temps sont définis dans le document ; ils ne sont pas représentés sur la figure.

3.8 Discussion du modèle CHM

L'évaluation du modèle se situe par rapport à l'état de l'art proposé conjugué aux spécificités du modèle et du domaine cible. Elle est présentée suivant deux points de vue : les caractéristiques du modèle et son rapport aux modèles de références de l'hypermédia.

3.8.1 Caractéristiques du modèle

L'approche proposée pour représenter formellement les hypervidéos est différente de celle entreprise par systèmes existants. Plusieurs aspects que nous jugeons importants pour le domaine sont pris comme critères pour évaluer le modèle : (1) la conformité aux spécifications du multimédia et de l'hypermédia ; (2) la prise en charge des spécificités de l'audiovisuel ; (3) la puissance d'expressivité ; (4) l'intuitivité et la simplicité ; (5) la générativité ; (6) les spécificités du modèle d'annotation et (7) l'implémentabilité, définie comme étant les possibilités d'implémentation du modèle.

1. Expression des dimensions multimédia : Le modèle proposé met en exergue le fait qu'une hypervidéo soit un cas spécifique des documents multimédia/hypermédias et propose un ensemble de modèles spécifiques qui répondent aux exigences de ces domaines tout en se spécialisant aux hypervidéos. Pour cela, des modèles de spécifications logique, spatiale, temporelle et hypermédia sont définis dans CHM.

2. Prise en charge de l'audiovisuel interactif : Les documents audiovisuels numériques sont augmentés grâce au modèle CHM par de l'annotation et des mécanismes de visualisation permettant d'être proprement intégrées dans des espaces d'information hétérogènes. Ainsi, les hypervidéos exprimées par CHM permettent d'introduire de nouvelles utilisations pour la vidéo interactive qui est au centre d'intérêt du modèle.

3. Puissance d'expressivité : Un autre objectif du modèle est de pouvoir décrire tout ce que l'on peut expérimenter actuellement comme hypervidéo. Le concept d'expressivité renvoie aux possibilités du modèle à représenter n'importe quel document hypervidéo.

Nous croyons que la philosophie ayant présidé à la définition du modèle est assez forte pour offrir un cadre théorique assez représentatif des hypervidéos permettant d'en faire une bonne abstraction logique et - à défaut d'un modèle hypervidéo de référence -, supposée pouvoir exprimer formellement tout type d'hypervidéo. Quand un document a un nouveau composant haut niveau (par exemple, un ruban, un menu, etc.), le modélisation peut se faire via des composants de base.

4. Intuitivité : Les concepts du modèle sont assez intuitifs et sont expérimentés par l'utilisateur le moins averti. Ainsi, les composants d'interface tels que les lecteurs, les composants de haut niveau du genre table de matière, les mécanismes de liens et la gestion événementielle ont été expérimentés dans différents modèles et systèmes et sont assez communs et intuitifs dans la pratique documentaire quotidienne.

5. Généricité : La modélisation à base d'annotation et de composants d'un document singularité bien spécifique au modèle CHM. Ce modèle ne considère pas le contenu mais plutôt les abstractions de plus haut niveau. Une généricité plus accrue est ainsi conférée à ce modèle.

6. Spécificités du modèle d'annotation : Basée sur le modèle Cinelab, la définition de la structure d'annotations dans CHM est très générale et englobe toute association de données au composant audiovisuel. Ainsi, le modèle est indépendant des considérations logicielles, bien qu'un schéma minimal soit requis. Ce schéma est toutefois intuitif, compte tenu de la définition du concept d'annotation et sa mise en œuvre peut être envisagé par exemple en exploitant les possibilités de *MPEG-7*.

7. Implémentabilité : Le modèle a été introduit d'une manière le rendant indépendant de toute plateforme ou environnement. Cela ne l'empêche pas d'être effectivement implémenté et mis à épreuve par des utilisations réelles, comme sera vu au prochain chapitre.

3.8.2 Relation de CHM aux modèles DHM/AHM

Le modèle que nous venons de présenter a pour but de représenter d'une manière formelle l'essentiel des concepts que peuvent exprimer les hypervidéos. A notre connaissance, CHM est parmi les rares modèles existant élaborés dans cet objectif, ce qui rend difficile d'établir un référentiel théorique pouvant servir pour le confronter. Néanmoins, les modèles de Dexter (DHM) et d'Amsterdam (AHM) constituent encore aujourd'hui

une référence importante pour tout formalisme hypermédia, couvrant d'une manière assez générale tous les concepts relatifs au domaine, dans de larges perspectives, et faisant abstraction des particularités des domaines applicatifs et des implémentations logicielles. CHM, par contre, a été proposé comme modélisation d'un type particulier des hypermédias. Nous croyons donc intéressant de comparer d'une manière générale leurs bases respectives et de relever quelques unes de leurs similitudes et différences.

Concept de composant : les concepts de composant, bien qu'ayant des définitions différentes, ont le même objectif : modéliser l'unité logique. Alors que cette unité est informationnelle dans le cas de AHM, elle est juste logique dans CHM. En effet, avec ce dernier, aucune donnée n'est contenue dans la déclaration d'un composant qui n'est pas un média mais un réceptacle de contenu. Au moment du rendu, un composant CHM contient toutes les informations que présente tout composant AHM : contenu, spécifications de présentation, attributs et ancrés. D'autre part, dans CHM et contrairement à AHM, la distinction n'est pas faite entre composants atomiques et ceux composites (issus d'une composition temporelle ou atemporelle) et en plus, tout composant peut être adressé dans sa granularité la fine n'étant pas considéré comme boîte noire indivisible.

Les liens : les liens dans CHM sont très différents de ceux de DHM/AHM qui définissent les liens comme composants et qui leur permettent d'être bi-directionnels et multi-destinataires. Comme la plupart des langages et systèmes hypermédias - y compris HTML et SMIL -, CHM ne définit pas une entité séparée pour un lien mais le considère attribut des ancrés.

Les aspects spatial et temporel : comme proposé dans AHM et SMIL, CHM sépare explicitement la spécification spatiale de celle temporelle. Alors que AHM définit la notion de *Channel* pour abstraire les styles de présentation, les composants CHM haut niveau comme les *players* et *viewers* au même temps que les layouts et les régions spatiales incluent la notion de *Channel* en définissant des attributs de haut niveau comme le niveau du son, les fontes par défaut, les dimensions, etc. Avec CHM, alors que certains attributs sont propres au composant, d'autres peuvent être intrinsèques au médias présentés.

Gestion dynamique : A la différence des modèles DHM et AHM, CHM ne dépend pas uniquement d'un modèle à base de liens pour définir et représenter un changement dynamique lors de la présentation du document, par exemple la navigation à travers des fragments indépendants du récit hypervidéo. Il s'articule aussi sur un mécanisme de

Couche de Dexter Model	Mécanismes et Concepts CHM
Runtime Layer	Génération et gestion du document
<i>Presentation Specification</i>	Attributs, Contenu d'annotations, Layout et Régions spatiales
Storage Layer	Composants bas et haut niveau et les structures d'annotation
<i>Anchoring</i>	Ancres, hotspots, contrôles des <i>lecteurs</i> et <i>visionneuse</i> , Interfaces des lecteurs d'annotations
Within-component Layer	Contenu d'annotation + vidéo

Tableau 3.11 – CHM exprimé par le modèle de Dexter

gestion axé sur la définition des événements et des traitements.

Modèle en couche de DHM : Pour situer les éléments introduits dans CHM par rapport au modèle de couche de DHM, le tableau 3.11 résume le *mappage* entre concepts CHM et couche DHM.

3.9 Conclusion

La proposition du modèle CHM est basée sur un ensemble de principes qui ont pour objectif de lui assurer une grande applicabilité en lui permettant d'être assez générique car présentant des caractéristiques intéressantes tout en respectant les préceptes des modèles simples et dédiés. Abordant les hypervidéos d'un point de vue nouveau, celui de la théorisation, sa validité ne peut être vérifiée que sur des domaines d'application bien définis. Pour cela, nous introduirons dans le prochain chapitre une implémentation qui aura pour cible l'hypermédia le plus largement répandu, le Web.

Chapitre 4

Hypervidéos CHM sur le Web : Architecture et Mise en Œuvre

LE MODÈLE CHM à composants et à base d'annotations nous a permis d'introduire un cadre théorique pour la conceptualisation des hypervidéos. Cette théorisation est d'un haut niveau d'abstraction ouvrant la voie à diverses implémentations, sur divers environnements et ciblant différentes plate-formes. Étant incontestablement le système hypermédia le plus large, le mieux déployé et le plus utilisé, le Web est l'environnement idéal pour mieux illustrer les implémentations possibles du modèle.

4.1 Motivations pour l'hypervidéo sur le Web

Dans la foulée de la révolution audiovisuelle actuelle, le phénomène de partage et de l'enrichissement de la vidéo sur le Web incarné par les sites Youtube⁴¹, Vimeo⁴², Dailymotion⁴³ et autres, ainsi que l'émergence des vidéoblogs et le marché de la vidéo à la demande (VOD) ont fait qu'aujourd'hui la vidéo en ligne, dont le contenu ne cesse d'augmenter en quantité mais aussi en qualité⁴⁴, soit l'une des sources d'information les plus sollicitées.

Les approches usuelles pour l'exploitation et l'interaction avec la vidéo numérique ne sont pas encore à même de capitaliser le déluge de la matière audiovisuelle brute constamment produite [31]. Les expériences acquises séparément avec les hypermédias et l'audio-

41. <http://www.youtube.com/>

42. <http://www.vimeo.com/>

43. <http://www.dailymotion.com/>

44. Le *Wall Street Journal* révèle en aout 2006 que le site Youtube tout seul héberge pas moins de 6.1 millions de vidéo avec 65000 nouvelles vidéos déposées dans ses serveurs quotidiennement demandant plus de 45 tera bytes d'espace de stockage et générant des recettes entre 5 à 6 milliards de dollars.

visuel peuvent être combinées dans une sorte de feuille de route permettant l'émergence d'une toute nouvelle culture pour la production et l'exploitation de la vidéo sur Internet. Bien que beaucoup de systèmes hypervidéos fassent appel à des navigateurs Web pour leurs rendus, ils restent confinés dans des langages impératifs souvent propriétaires et qui ne font que diminuer la synergie pouvant résulter de la combinaison de l'hypervidéo avec les technologies du Web. *WebCHM*⁴⁵, la mise en œuvre du modèle CHM, nous est l'occasion de discuter de la prise en charge effective de la vidéo sur le Web.

4.2 Choix technologiques

La conformité et l'intégration avec la pile existante des technologies Web les plus largement supportées par les navigateurs, et la possibilité de les réutiliser au besoin, sont des éléments fondamentaux pour l'utilisation, le succès et la vulgarisation de toute nouvelle technologie Web [29]. De ce fait, les choix technologiques retenus lors de la conception de *WebCHM* sont opérés de telle manière à garantir aux documents résultant d'être *recherchables*, accessibles, structurés, réutilisables et conformes aux standards Web.

a) Langage d'expression des hypervidéos

Pour exploiter l'omniprésence du Web et de l'évolution constante des navigateurs, nous nous intéressons naturellement au standard HTML et les langages accompagnant (JavaScript, CSS, etc.) pour l'implémentation du modèle. Notre implémentation se base sur plusieurs principes et paradigmes, que nous développons dans ce qui suit.

Syntaxe déclarative : Les standards W3C préconisent les approches déclaratives pour la manipulation des documents complexes [45]. C'est dans cet esprit qu'ont été conçus des langages de transformation de document XML, de définition de style (*CSS*, *XSL*), d'animation et graphiques 2D (*SVG*) et de synchronisation (*SMIL*).

La spécification de langages de *markups* descriptif est un cas particulier des approches de développement des *DSLs* (*Domain Specific Languages*) à base de syntaxes déclaratives. Ces langages permettent à l'auteur de créer une description de haut niveau du comportement désiré au lieu de *coder* ce comportement, ce dernier aspect étant considéré du ressort de la plate-forme hôte. Cette approche offre une meilleure lisibilité du document, une plus grande facilité d'utilisation et de réutilisation, un meilleure maintenance et une accessibilité accrue [42].

45. Une description de CHM avec un exemple d'analyse filmique réalisé avec *WebCHM* sont disponibles à l'adresse : <http://www.advene.org/chm/>

Nous proposons de mettre en œuvre le modèle CHM sous forme d'un langage exprimé via une syntaxe déclarative pour la création d'hypervidéos pour le Web. Consistant en un ensemble d'éléments définis sur un espace de nom dédié⁴⁶, il permet, une fois ses éléments combinés à ceux du HTML, de décrire des documents hypervidéos riches et complexes.

Instrumentation du langage HTML : La spécification de la syntaxe CHM est implémentée comme une extension au-dessus du langage HTML. L'approche consistant à ajouter un nouveau comportement aux langages XML statiques a déjà fait l'objet de divers travaux antérieurs [42]. D'autre part, bien que le langage HTML ne soit pas principalement conçu pour les présentations multimédia, des solutions l'étendant peuvent être envisagées pour une prise en charge, même partielle, de l'aspect multimédia.

Le standard HTML ne prenant pas en charge les éléments d'un autre espace de nom⁴⁷, l'implémentation des éléments CHM comme ensemble d'éléments (*balises*) exige que les documents soient écrits en format XHTML - qui se base sur la syntaxe définie par XML, plus récente, mais plus exigeante que celle définie par SGML sur laquelle repose HTML -. Or, XHTML est très peu répandu pour les documents sur le Web et son support par les navigateurs est encore mitigé. Pour optimiser l'utilisabilité de la syntaxe, nous proposons d'ajouter aux éléments HTML de nouveaux attributs propres à l'*espace de nom (namespace) CHM*. Un moteur de transformation permettra alors de réécrire et générer automatiquement un code compatible avec la spécification HTML et compréhensible par les navigateurs.

b) Librairies pour interpréter la syntaxe CHM

Motivés par des soucis de fiabilité, de sécurité, de performances et de rapidité, les créateurs de navigateurs n'implémentent que peu du nombre toujours croissant de nouveaux standards et langages pour le Web [95]. Ceci est la raison principale qui a motivé les développeurs à adopter les *plugins* tels que Flash, malgré toutes leurs vulnérabilités [42], comme technologie de recharge pour le multimédia sur le Web .

Une des solutions les plus préconisées pour rehausser les navigateurs tout en assurant un meilleur support par les navigateurs et moins de vulnérabilités - donc, sans recourir aux plugins -, consiste à développer des librairies (généralement JavaScript et/ou Ajax) qui sont exécutées uniquement au niveau client. Ces librairies permettent d'écrire des portions de code au navigateur qui se charge de les analyser, les *parser* puis de les interpréter, sans

46. L'espace de nom *CHM Hypervideo* porte l'adresse `xmlns:chm="http://www.advene.org/ns/chm"`

47. La spécification (<http://www.w3.org/TR/html5/syntax.html#the-doctype>) stipule que : "The HTML syntax does not support namespace declarations, even in foreign elements".

recourir à d'outils supplémentaires ou de logiciels tiers. L'inconvénient majeur de cette technique provient du fait que certaines fonctionnalités sont absentes du code exécutable au côté-client. Par exemple, l'accès aux fichiers et ressources locaux est restreint par les politiques de sécurité et de confidentialités des navigateurs.

Le prototype *WebCHM*, disponible en accès libre⁴⁸, est constitué d'un ensemble de librairies JavaScript développées avec le framework JQuery⁴⁹. Ce dernier est une bibliothèque qui porte sur l'interaction entre JavaScript comprenant AJAX et HTML, et qui a pour but de simplifier des commandes communes du langage (modifier le DOM, réagir aux événements, faire des requêtes HTTP, etc.). La principale difficulté avec JavaScript est que différentes implémentations du langage sont utilisées par les navigateurs. Un des objectifs des bibliothèques JavaScript est de s'affranchir de ces différences et incompatibilités.

4.3 Scénario : Intervention de Tim Bernes Lee lors de la conférence TED'09

L'implémentation des concepts de CHM sera détaillée progressivement tout en illustrant les étapes de sa mise en œuvre à travers un exemple de scénario.

Scénario : En 2009, au cours de la conférence *Technology Entertainment Design* (TED⁵⁰) - une importante rencontre qui se tient chaque année en Californie - , Tim Berners-Lee, le co-inventeur avec Robert Cailliau du *World Wide Web* et président du *World Wide Web Consortium*(W3C), promeut le projet *Linked data*⁵¹ qui a pour objectif de décrire une méthode pour exposer, partager et connecter les données via des URIs sur le Web. Le site Web du projet Advene présente une analyse de cette intervention. Il est lui-même généré par Advene⁵² pour illustrer comment certaines idées émises par Tim Berners-Lee lors de cette conférence peuvent être mises en œuvre⁵³.

Description de la présentation hypervidéo : Nous voulons présenter une vue du talk de la conférence d'une manière augmentée et interactive, en se basant sur les annotations et en se référant aux technologies hypervidéos. Le scénario que nous voulons réaliser

48. <http://www.advene.org/chm/api>

49. www.jquery.com, sous licences : Licence MIT, GNU GP

50. <http://www.ted.com>

51. http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html

52. http://www.advene.org/examples/tbl_linked_data/

53. Le recueil Advene de la conférence et d'autres exemples peuvent être récupérés à l'adresse : <http://www.advene.org/examples.html>

4.3. Scénario : Intervention de Tim Berners Lee lors de la conférence TED'09

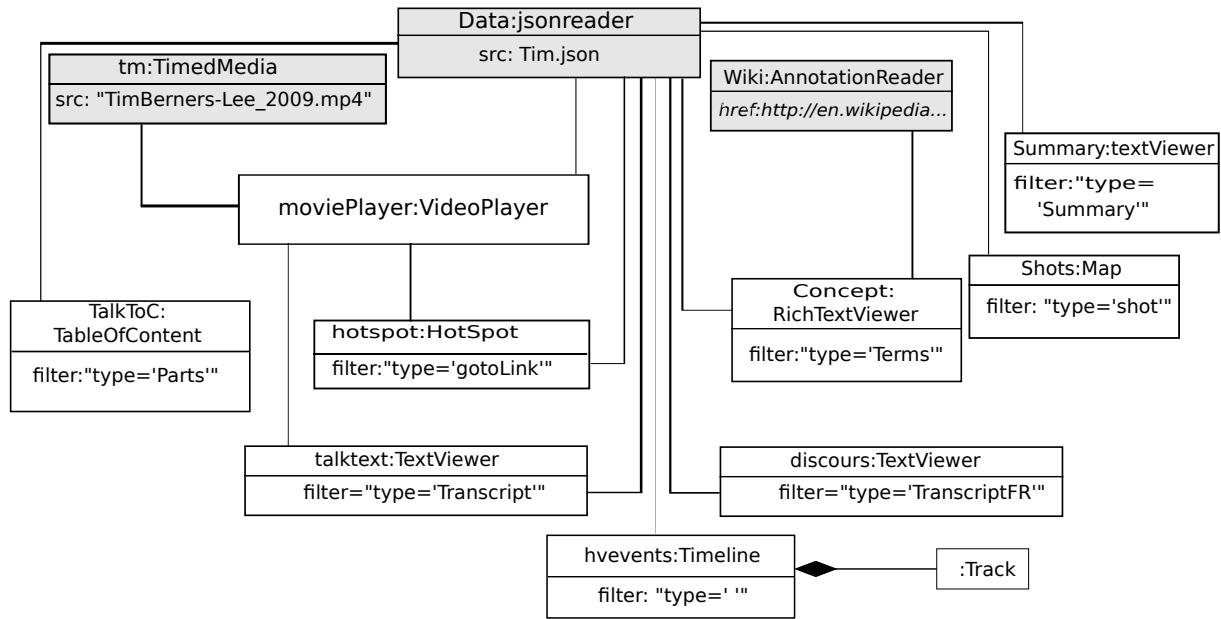


Figure 4.1 – Représentation formelle du scénario

peut être résumé en un ensemble de points :

- la vidéo est l'élément central de la présentation et un utilisateur (lecteur) peut se contenter de juste suivre la conférence de bout en bout, sans aucune interaction ;
 - une table de matière des parties du film est présentée en permanence à gauche de la page avec comme contenu les parties du *talk*. L'activation d'une entrée donne accès directement au fragment correspondant.
 - La transcription du discours de Tim Berners-Lee est affichée comme sous-titrage, au coin inférieur du lecteur vidéo ;
 - la traduction française de la transcription est affichée au dessous de la vidéo ;
 - une cartographie des scènes de la vidéo en forme d'images est affichée en permanence du côté droit de la page. L'activation d'une entrée de la cartographie donne accès soit à un aperçu vidéo du fragment de destination, soit la lecture de la destination ;
 - un résumé de la partie en cours est disponible à côté du lecteur vidéo ;
 - à chaque fois qu'un nouveau concept est énoncé, sa définition est récupérée à partir de *Wikipédia* puis affichée sur un espace dédié ;
 - à chaque fois que l'orateur donne une adresse URL, l'utilisateur peut cliquer sur la figure de l'orateur pour aller vers le lien annoncé ;
 - une timeline place tous les événements suivant l'axe temporel.

Alors que cet exemple peut être enrichi et rendu plus complexe par le biais d'autres composants nécessitant une gestion plus avancée, il est à noter que son édition par les outils disponibles est déjà une tâche très difficile.



Figure 4.2 – Processus général de création hypervidéo

À fur et à mesure que les concepts et les méthodes mis en œuvre sont énoncés, nous progressons dans le traitement de cette vidéo dans le but d'illustrer comment une hypervidéo peut être exprimée facilement et gérée par le prototype *WebCHM*.

Nous nous basons sur le scénario et sa représentation formelle par le biais du modèle CHM présentée à la figure 4.1 dont les composants et relations seront détaillés progressivement dans les prochains paragraphes.

4.4 Architecture du prototype WebCHM

Un document hypervidéo est édité par l'auteur avec la syntaxe CHM sous n'importe quel éditeur HTML. La génération de la présentation fait appel à un ensemble de librairies qui instancient les modules du moteur de rendu. La figure 4.2 résume l'ensemble des étapes de conception d'hypervidéos, distinguant trois phases principales :

1. annotation d'un corpus de documents audiovisuels, manuellement ou par un outil dédié. Nous illustrons l'utilisation du prototype Advene pour l'exemple introduit et qui nous permet d'enregistrer et d'exporter la structure d'annotation sous les formats appropriés ;
2. composition d'un document CHM à base du corpus annoté :
 - le contenu générique est exprimé en HTML simple auquel des classes, des scripts, du style, etc. peuvent être définis ;
 - les éléments hypervidéos sont exprimés en syntaxe CHM appliquée à des éléments HTML ;
3. génération, à la présentation, d'une hypervidéo à partir du document CHM :

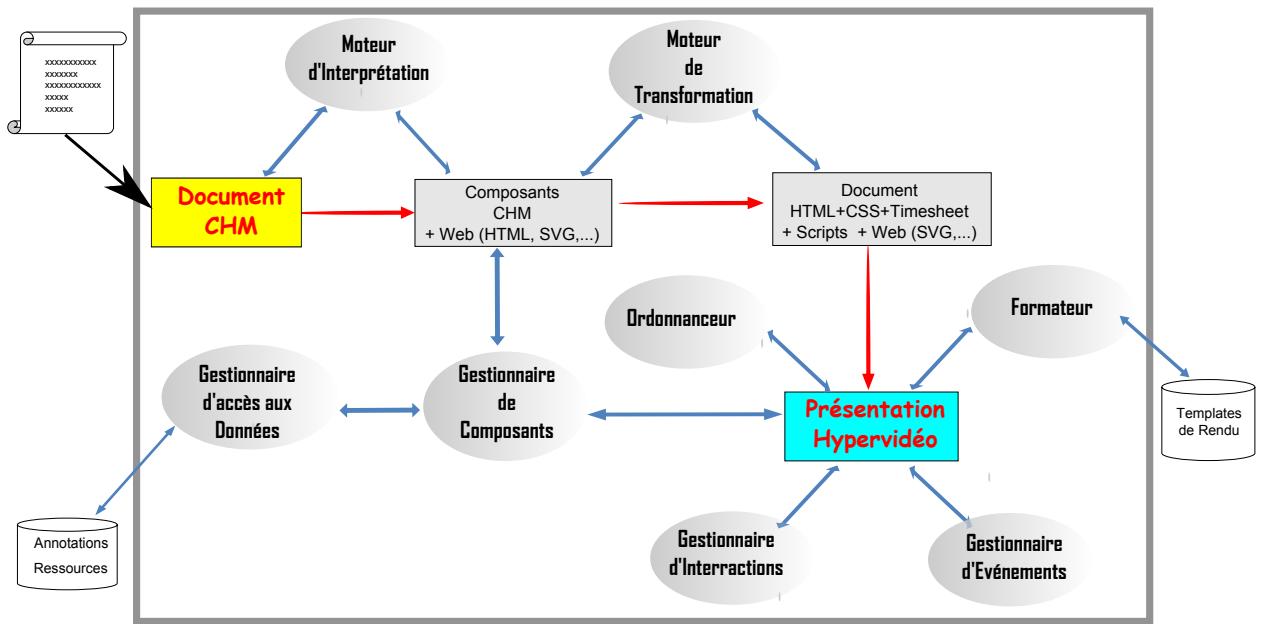


Figure 4.3 – Architecture générale du prototype WebCHM

- le positionnement spatial et les dimensions de la layout, des régions spatiales et des éléments visuels sont décrits par des feuilles de style CSS ;
- la spécification temporelle et la synchronisation sont décrites et gérées par des scripts dédiés (mettant en œuvre *SMIL TimeSheets* que nous introduirons ultérieurement) ;
- la gestion dynamique de la présentation est réalisée par l'ensemble des modules du moteur de rendu.

L'architecture générale du prototype *WebCHM* est composé de plusieurs modules, mis en œuvre par des librairies qui effectuent une analyse et une transformation de contenu pour générer dynamiquement un code HTML. Les principaux modules le composant sont représentés à la figure 4.3 :

- *Moteur d'Interprétation* (*CHMParse*) : sa tâche est d'analyser le document et de recenser les composants CHM ;
- *Moteur de Transformation* (*CHMTransformer*) : sa tâche est de transformer les composants hypervidéos en éléments HTML standards ;
- *Gestionnaire de composants* (*CHMComponentsManager*) : sa tâche est de gérer les éléments HTML générés comme composants atomiques ;
- *Gestionnaires d'accès aux données* (*DataAccessManager*) : leur tâche est de gérer les requêtes d'accès aux données (annotations et ressources) ;
- *Formateur* (*CHMFormatter*) : sa tâche est de gérer les spécifications spatiales - et



Figure 4.4 – Annotation de la vidéo par Advene

éventuellement les modèles de rendu spécifiés - du document ;

- *Ordonnanceur (CHMSchedule)* : sa tâche est de gérer les spécifications temporelles et de synchronisation du document ;
- *Gestionnaire d’Interactions (CHMInteractionManager)* : sa tâche est de gérer les interactions de l’utilisateur avec le contenu du document ;
- *Gestionnaire d’Événements (CHMEventManager)* : sa tâche est de gérer les spécifications événementielles du document ;

4.5 Annotation comme pré-traitement

L’objectif du *WebCHM* n’étant pas de développer des outils d’annotation, les formats utilisés peuvent être obtenus en utilisant des outils dédiés. Nous illustrons l’utilisation du prototype *Advene* pour réaliser ce pré-traitement en analysant les étapes d’annotation de l’intervention de Tim Berners-Lee lors de TED’09.

Préparation des données et du contenu du scénario : la préparation des données consiste en l’annotation de la vidéo et au collecte des contenus et ressources qui seront utilisés lors de la construction de l’hypervidéo.

Étape 1- analyse et annotation de la vidéo de la conférence : La vidéo de la conférence est analysée et annotée avec le prototype Advene. La figure 4.4 montre la prise d'écran du prototype durant cette activité.

La démarche qui a permis l'annotation de la vidéo est décrite en détail sur le site du projet Advene où toutes les annotations avec leurs types et leurs vues générées peuvent être explorés en détail. Cette démarche peut être résumée comme suit :

L'analyse de la vidéo commence par la détection de plans réalisée par l'outil de détection d'Advene (**Player > Detect shots**) permettant d'établir une première structuration de la vidéo. La définition de ces plans établit un type d'annotations (**shot**). En définitive, 165 annotations sont définies pour ce type d'annotation.

La transcription de la parole n'est pas automatique mais utilise l'interface de prise de note (**Note-taking editor**). Avec la fonction d'export en annotation, un nouveau type, **Transcript**, est défini, regroupant 158 annotations. La transcription du discours est traduite vers la langue française, définissant un nouveau type d'annotation **TranscriptFR** regroupant autant d'annotations que pour le type *Transcript*.

Le discours peut être divisé en plusieurs parties, regroupées en un seul type d'annotation **Parts**. Les 12 annotations de ce type sont alignées temporellement avec celles de transcription correspondant.

À certains endroits du discours, l'orateur fait référence à des ressources externes. Pour capturer ces références, un type d'annotation, **Links**, est défini, regroupant 7 annotations. À chaque fois que l'orateur mentionne une référence intéressante, la transcription d'annotation correspondante est copiée comme une annotation *Link* (afin d'obtenir l'horodatage), puis son contenu est changé en un contenu structuré avec un texte (l'étiquette du lien) et une URL. Simultanément à la définition des annotations de type *Links*, des hotspots sont définies sur la figure de l'orateur permettant d'aller directement vers le lien énoncé, dans une fenêtre séparée. Ce type d'annotation est appelé **GotoLink** et contient donc 7 annotations.

Étape 2 : Export du contenu et des annotations L'une des fonctionnalités les plus pratiques du logiciel Advene réside dans la possibilité d'exporter la structure d'annotations vers divers formats : SRT, SMIL, SVG timeline, Spreadsheet, HTML+CSS timeline, MPEG7, IRI, RDF, JSON et d'autres formats XML.

Pour notre implémentation, nous illustrons l'utilisation du format JSon (JavaScript Object Notation) qui est un format de données textuel, générique, dérivé de la notation des objets du langage ECMAScript. Compréhensible par l'être humain, il est facilement

analysable (*parseable*) par la machine. Un document JSON ne comprend que des ensembles de paires (nom, valeur) et des listes ordonnées de valeurs. Ces éléments représentent trois types de données : des objets, des tableaux et des valeurs génériques (*tableau*, *objet*, *booléen*, *nombre*, *chaîne* ou *null*).

Une partie du code de l'export de la structure d'annotation réalisée avec Advene est représentée sur le listing 4.1.

```

1 [...]                               "begin": 0,
3 {                                     "end": 44701,
4   "id": "a1072",                      "content": "So going back to 1989, I wrote a memo
5   "type": "Transcript",                suggesting global hypertext system."
6   "media": "TimBerners-Lee_2009.mp4",  },
7   "begin": 36223,                      35 {
8   "end": 44701,                        "id": "a9",
9   "content": "So going back to 1989,      "type": "shots",
10  I wrote a memo suggesting global hypertext system."39  "media": "TimBerners-Lee_2009.mp4",
11 },                                     "begin": 37708,
12 {                                         "end": 46291,
13   "id": "a1242",                       "content": "num=8",
14   "type": "Links",                     "parsed": { "_all": "num=8", "num": "8" }
15   "media": "TimBerners-Lee_2009.mp4", 43 },
16   "begin": 36223,                      45 {
17   "end": 44701,                        "id": "a1073",
18   "content": "text=Tim Berners-Lee's      "type": "Transcript",
19 1989 memo\nurl=http://info.cern.ch/Proposal.html",  "media": "TimBerners-Lee_2009.mp4",
20   "parsed": {"_all": 47 "begin": 44701,
21   "text=Tim Berners-Lee's 1989 memo\n      "end": 47822,
22   url=http://info.cern.ch/Proposal.html", 51 "content": "Nobody really did anything with it very
23   "text": "Tim Berners-Lee's 1989 memo",      much."
24   "url": "http://info.cern.ch/Proposal.html"} 53
25 },                                     55 [.....]
26 {
27   "id": "a1341",                      55
28   "type": "Summary",                  55
29   "media": "TimBerners-Lee_2009.mp4", 55

```

Listing 4.1 – Code JJson (partie de Tim.json)

Les exports Web d'Advene permettent de récupérer les ressources utilisées lors de l'annotation de la vidéo, entre autres les images clés extraites. D'autres ressources peuvent aussi être incluses après une préparation préalable du contenu.

Avec la structure d'annotation, les ressources des exports et le fichier audiovisuel, tous les éléments constitutifs sont réunis pour l'édition hypervidéo en CHM.

4.6 Édition et spécification des composants

4.6.1 Déclaration de composants CHM

Le modèle de composants est mis en œuvre dans *WebCHM* par l'utilisation des éléments HTML standards auxquels des attributs du nom d'espace CHM sont ajoutés.

Élément hypervidéo : lors de l'édition d'un document, un élément HTML peut être déclaré comme conteneur de composants CHM en lui ajoutant l'attribut *chm:component*. La définition de cet attribut permet de traiter l'élément comme composant hypervidéo dont le type est déterminé selon la valeur de l'attribut. L'utilisation de conteneur de type DIV est préconisé.

Composant hypervidéo : tous les composants hypervidéos héritent de l'élément **component** et sont déclarés en donnant leurs noms comme valeurs de l'attribut *chm:component*. Par défaut, les composants sont atemporels. La définition de l'attribut *chm:timelineref* permet de lier le temps du composant à celui d'une référence de temps (et d'en faire un **TLComponent**) et par conséquent, la valeur *tlcomponent* pour l'attribut *chm:component* n'est pas utile (ni utilisable).

Lecteurs d'annotations : les annotations sont stockées sous les formats appropriées et ont une structure générale qui inclut au moins les attributs : *id*, *type*, *begin*, *end*, *content* et *media*. Les lecteurs d'annotations utilisés doivent comprendre la structure de stockage des annotations pour pouvoir en extraire les informations et utiliser une syntaxe appropriée pour les filtres. Plusieurs lecteurs d'annotations peuvent être combinés pour le même composant.

Liens hypervidéos Les liens hypervidéos sont des liens avec au moins une ancre définie sur une région spatio-temporelle d'un média continu. Quand un lien part d'une région spatio-temporelle d'un média, son ancre de départ est appelé *hotspot*. Quand un lien part vers une région spatio-temporelle d'un média, son ancre de départ est soit une hotspot définie par son identificateur, soit l'identificateur ou l'URI du composant de rendu média.

L'adressage de fragment est fait d'une manière augmentée suivant un sous-ensemble de la spécification *W3C Media Fragment*⁵⁴ pour pointer des fragments de média. La

54. <http://www.w3.org/2008/WebVideo/Fragments/>

spécification étant à stade de travail (*Working Draft*)⁵⁵, la spécification n'est pas encore adoptée.

Cette syntaxe permet de définir un fragment temporel délimité par un début et une fin avec la fin pouvant ne pas être spécifiée. Il est possible de spécifier par exemple :

```
chm : href = "hotspotID";
chm : href = "componentID#t=.120";
chm : href = "componentID#t=.120,1.230";
chm : href = "http://www.exemple.com/video.mp4\#t=.120,1.230";
```

La spécification prévoit de pouvoir définir les dimensions *spatiale*, *temporelle* et de *piste* aux fragments. Nous retenons de la spécification la dimension temporelle, les autres étant considérées comme contenu d'annotation.

Événements et actions : les événements peuvent être utilisés d'une manière *programmatique* grâce aux mécanismes de programmation par scripts. La mise en œuvre du modèle événementiel CHM permet l'utilisation des événements et actions d'une façon déclarative. Un événement est déclaré par une valeur `event` attachée à l'attribut `chm:component`. Il

Action	Paramètres	Exemple
domaction	Attributs et fonctions de l'interface DOM standard (attributs et fonctions).	<code>chm:param= "element('id').attr('chm:src', ...); element.show() "</code>
openurl	Ouvre une adresse exprimée par son URL.	<code>chm:param= "href=http://www.advene.org/chm/&&target = '_self' && behav='pause'"</code>
modifyleayout	Nouveaux attributs de la région ou de l'élément visuel à modifier.	<code>chm:param = "region ='id'&&top=20px&&width = 50px; component ='idc'&&chm:region = 'idnewregion'"</code>
tlrupdate	modifie l'état de la TLR status ou son instant courant	<code>chm:param = "position = '1:33:12.500' &&status = 'play'"</code>
executescript	nom ou corps de la fonction à exécuter.	<code>chm:param = "executeTheScript(10,22);alert('this is a classic javascript alert message');"</code>

Tableau 4.1 – Exemples d'événements *WebCHM*

peut comprendre une liste d'actions identifiées par des identificateurs (IDs) ; cette liste n'est pas exhaustive car il est possible, à la déclaration d'une action, de l'attacher à un événement qui lui, n'en fait pas référence. Les actions sont explicitement déclarées en assignant la valeur `action` à un composant CHM. Des exemples simples sont fournis sur le tableau 4.1.

55. Un second appel par le *W3C Media Fragments URI 1.0 Working Draft* est émis en mars 2011, voir sur <http://www.w3.org/TR/2011/WD-media-frags-20110317/>

4.6.2 Composants CHM du Scénario

Nous illustrons l'utilisation de la syntaxe en développant le code permettant la déclaration du contenu hypervidéo de l'exemple du scénario introduit.

a) Définition de la layout et du style de l'hypervidéo de TED'09 : la disposition générale de la présentation est structurée en une *layout* avec cinq régions spatiales, déclarées dans le listing 4.2.

1. haut (*top*) : la région du haut est occupée par une description du contexte de l'hypervidéo ;
2. milieu (*center*) : au milieu de la page est affichée la vidéo principale et du fait, la transcription d'origine sur le lecteur et la traduction française au dessous du lecteur ;
3. bas (*bottom*) : dans la région inférieure on trouve la ligne de temps et les différentes pistes ;
4. gauche (*left*) : la table de matière occupe la région gauche de la présentation.
5. droite (*right*) : la région de droite est occupée par un ensemble de composants explicatifs : le résumé de la partie du discours courante, les épisode représentant les plans de la vidéo et enfin, une région d'affichage du contenu Wikipédia sur les notions nouvelles.

```
<!-- layout -->
<div chm:component="layout" id="sp_lay"></div>
<!-- Régions spatiales -->
<div chm:component="region" id="center" chm:layout="sp_lay"></div>
<div chm:component="region" id="top" chm:layout="sp_lay"></div>
<div chm:component="region" id="bottom" chm:layout="sp_lay"></div>
<div chm:component="region" id="left" chm:layout="sp_lay"></div>
<div chm:component="region" id="right" chm:layout="sp_lay"></div>
```

Listing 4.2 – Déclaration de la *layout* et des régions spatiales

Le style appliqué à ces régions spatiales est décrit en langage CSS, dans une feuille de style accompagnant le document et ce, dans le souci de réutilisabilité⁵⁶.

b) Déclaration du média principal et de son lecteur : la déclaration du *Timed-Media* ainsi que le lecteur permettant de le jouer est illustrée dans le listing 4.3.

56. N'ayant rien de particulier à *WebCHM*, un exemple de ces feuilles CSS peut être exploré en ligne.

```
<!-- TimedMedia -->
<div chm:component="timedmedia" id="tm"
chm:src="TimBerners-Lee_2009.mp4"></div>

<!-- Lecteur vidéo -->
<div chm:component="videoplayer" id="movieplayer"
chm:src="tm" chm:region="center"></div>
```

Listing 4.3 – Déclaration du *TimedMedia* et de son VideoPlayer

L’élément *TimedMedia* est dans cet exemple la vidéo intégrale. Un fragment ou un ensemble de fragments de plusieurs régions peuvent aussi être utilisés pour réaliser un montage. Dans ce cas, le *TimedMedia* lui-même fait appel aux lecteurs d’annotations. Par exemple, si l’on voulait utiliser les fragments délimités par le type d’annotation “*shot*”, et supposant que l’on déclare un lecteur d’annotations avec comme identificateur *data*, la syntaxe du *TimedMedia* serait la suivante (listing 4.4) :

```
<!-- TimedMedia -->
<div chm:component="timedmedia" id="tm"
chm:src="data" chm:filter="type=='shot'"></div>
```

Listing 4.4 – Déclaration d’un *TimedMedia* comme montage

c) Lecteurs d’annotations : deux sources de données sont introduites par l’exemple : la structure d’annotation sauvegardée sous format JSON (‘*data*’) et l’encyclopédie en ligne Wikipédia (‘*wiki*’). Notons que les ressources média nécessaires sont récupérées à partir d’un export Web ainsi que la vidéo de la conférence.

```
<!-- The JSON format Annotation Reader -->
<div chm:component="jsonreader" id="data"
chm:src="Tim.json"></div>

<!-- The Annotation Reader from wikipedia.org-->
<div chm:component="annotationreader" id="wiki"
chm:src="http://en.wikipedia.org/w/api.php?action=parse&props=text&page=""></div>
```

Listing 4.5 – Déclaration des lecteurs d’annotations

d) Transcription : la transcription du discours est affichée sur le composant de rendu du *TimedMedia* d’une manière synchronisée. Pour celà, un style bien particulier doit être défini mettant en relation le rendu de la transcription avec celui de la vidéo. Le listing 4.6 contient la déclaration du composant de rendu textuel de la transcription.

```
<div id='talktext' chm:component='textviewer' chm:timelineref='tm'
chm:src='data' chm:filter="type=='Transcript'" chm:region="center"/>
</div>
```

Listing 4.6 – Déclaration du conteneur de transcription

e) Traduction : la traduction du discours est affichée par un composant textuel synchronisé avec la vidéo. Sa déclaration est présentée sur le listing 4.7.

```
<div id='discours' chm:component='textviewer' chm:timelineref='tm'
chm:src='data' chm:filter="type=='TranscriptFR'" chm:region="center"/>
</div>
```

Listing 4.7 – Déclaration du conteneur de la transcription traduite

f) Table de matière : la table de matière est un composant ToC non temporalisé. Le code permettant de la générer est le suivant (Listing 4.8) :

```
<!-- Table of Content -->
<div id="test" chm:component='toc' chm:src='data'
chm:filter="type=='Parts'" chm:region="left"></div>
```

Listing 4.8 – Déclaration de la table de matière

g) Map des shots : la cartographie des shots (image et vignettes vidéo) est un composant map non temporalisé représentant l'ensemble des scènes détectées par le prototype Advene. Le code permettant de la générer est présenté dans le listing 4.9

```
<!-- Map -->
<!-- Table of Content -->
<div id="shots" chm:component='map' chm:src='data'
chm:filter="type=='shot'" chm:region="right"></div>
```

Listing 4.9 – Déclaration d'une Map

h) Résumé : le résumé du discours est représenté par un composant temporalisé qui se met à jour d'une manière synchronisée avec la lecture de la vidéo. Le listing 4.10 résume la déclaration de cet élément.

```
<div id="summary" chm:component='textviewer' chm:timelineref='tm'
chm:src='data' chm:filter="type=='Summary'" chm:region="right"></div>
```

Listing 4.10 – Déclaration du conteneur du résumé du discours

i) Définition Wikipédia : quand l'orateur cite un concept-clé, sa description est obtenue à partir de Wikipédia. Le composant temporalisé *richtextviewer* utilise donc les deux lecteurs. Ce composant illustre bien l'utilisation simultanée de plusieurs lecteurs d'annotations.

```
<div id="concepts" chm:component='richtextviewer' chm:timelineref='tm'
chm:src='wiki' chm:target='data' chm:filter="type=='Terms'"
chm:param="wiki.page={name}"" chm:region="right">
    <a style="font-style: italic"> Concepts (from Wikipedia.org) </a>
</div>
```

Listing 4.11 – Déclaration d'un conteneur de description *Wikipédia*

j) Hotspots : les liens spatio-temporels sont définis sur la figure de l'orateur et permettent d'aller vers le lien énoncé. Leur déclaration est donnée dans le listing 4.12.

```
<div chm:component='hotspot' chm:src='data' chm:filter="type=='gotoLink'"
chm:src='data' chm:target='_blank' chm:leavebehav='continue',
chm:region="center"></div>
```

Listing 4.12 – Déclarations des hotspots

k) Ligne de temps : la ligne de temps et les pistes accompagnant, déclarées comme montré sur le listing 4.13 permettent de représenter graphiquement tous les événements de la vidéo.

```
<div id='hvevents' chm:component='timeline',
chm:src='data' chm:filter='', chm:region="bottom"> </div>
```

Listing 4.13 – Déclaration d'une ligne de temps

l) Événements : Pour illustrer la déclaration d'un événement et d'une action, définissons deux événements :

1. Nous supposons, pour ne pas gêner le lecteur, n'afficher la hotspot que si le lecteur le désire, par un survol de son emplacement durant l'intervalle de son activation. L'événement *ev1* est déclenché au passage de la souris sur lecteur vidéo principal (*hover*). Son action *act1* est de vérifier si une hotspot est active et de l'afficher.
2. *ev2* est un événement attaché à la TLR qui permet de relancer la présentation à la fin de la lecture (*fin* correspondant au fait que la position actuelle est égale à la durée, et non pas que l'état de la présentation est à *stop*) (action *act2*).

La déclaration des événements et des actions est présentée sur le listing 4.14.

```

        <! ---- EVENTS ---- >
<div id='ev1' chm:component='event' chm:src='movieplayer'
chm:eventtype='domevent.hover' ></div>

<div id='ev2' chm:component='event' chm:src='tm'
chm:eventtype='timelinerefupdated' chm:condition="tm.position==tm.duration"
></div>

        <! ---- ACTIONS ---- >
<div id='act1' chm:component='action' chm:eventids='ev1'
chm:typeaction='domaction' chm:param='$(".hotspot").css("visibility",
"visible")'></div>

<div id='act2' chm:component='action' chm:eventids='ev2'
chm:typeaction='timelinerefupdate'
chm:param='$(this timelineref).position=0;
$(this timelineref).status="play"); '></div>

```

Listing 4.14 – Déclaration des événements et des actions associées

4.7 Interprétation, transformation et génération de code

4.7.1 Moteur d’interprétation

Ce module définit un ensemble de règles d’inférence permettant de faire la cartographie des composants CHM utilisables dans le document. Un "*sac de composants*" est donc déduit et distingué du reste des éléments non CHM. Ces composants vont être affectés des références internes permettant de les gérer tout au long de la présentation. Chaque fois qu’un composant est référencé, son identificateur est sauvegardé pour le gestionnaire de composants. L’algorithme simplifié de ce moteur est le suivant :

```

function CHMParse(document)
{
    foreach ELEM from document
        if (ELEM.namespace=='CHM')
        {
            var Component;
            Component=InitializeComponent(ELEM);

```

```

    CHMComponentsManager.push(Component);
}
}
}
```

Listing 4.15 – Code simplifié du moteur d’interprétation

4.7.2 Moteur de transformation

Le moteur de transformation génère le code HTML des composants identifiés par le moteur d’interprétation. Pour un seul composant, le code généré peut être très long et assez compliqué d’autant plus que plusieurs transformations peuvent être nécessaires pour la génération du code final. Par exemple, une table de matière doit d’abord être transformée en du code HTML combiné avec des composants CHM de plus bas niveau (lecteurs de texte, liens,...). L’algorithme simplifié de ce moteur est le suivant :

```

foreach ELEM from document
    if (ELEM.namespace=='CHM')
    {
        var Component = CHMComponentsManager.pop();
        Transformer.Apply( Component ); // Appliquer la transformation adéquate.
        CHMParse( Component ); // Nouvelle passe du parser
    }
}
```

Listing 4.16 – Code simplifié du moteur de transformation

Des modèles de transformation permettent la génération de code d’un manière homogène. Par exemple, pour générer le composant de rendu textuel, le code exécuté par le moteur, écrit en langage de template de JQuery, a la forme générale suivante :

```

// Récupération du filtre d’interrogation du lecteur d’annotation
filter=$(this).attr('chm:filter');

// Déclaration du conteneur HTML qui va recevoir le nouveau code du composant
container=$( '<div data-timecontainer="excl" data-timeaction="display"
data-mediasync="video"/>' );

// Déclaration de la forme finale de chaque donnée renvoyée par le lecteur
d’annotation
tag="{{if "+filter+"}}<p data-begin=${begin} data-end=${end} style='display:
none;'>${content}</p>{{/if}}";
```

```

$.template( "TextReaderTemplate", tag );

// Envoi de la requête avec les instructions de mise en forme puis récupération
// de la réponse
transformationResult= $.tmpl("TextReaderTemplate", data);

// Transformation appliquée. Ajoute du nouveau contenu au conteneur.
$(container).append(transformationResult);

```

Listing 4.17 – Code générant un composant de rendu textuel

Ce code peut être lu comme suit :

- le filtre d’interrogation du lecteur d’annotations est récupéré ;
- un conteneur HTML est déclaré avec les attributs temporels nécessaires ;
- une template de rendu est déclaré. Il stipule que pour toutes les données en entrées, celles qui satisfont le filtre sont formatées dans des portions de paragraphes contenant le contenu d’annotation. Ces textes ne sont pas affichés et leurs attributs temporels sont récupérés des attributs des annotations ;
- la template est exécutée sur les données d’annotations (le lecteur d’annotation a l’identificateur *data*) ;
- le résultat (l’ensemble des paragraphes) est ajouté au conteneur.

L’exécution d’une telle transformation peut générer des milliers de lignes, dépendant de la taille de données renvoyées par le lecteur d’annotations. Pour chaque donnée respectant le filtre, un nouvel élément (paragraphe) est créé.

Les composants audiovisuels et plus spécifiquement les **TimedMedia** sont transformés en utilisant principalement les nouveaux éléments multimédia introduits par HTML5 (**<audio>** et **<video>**). Pour les navigateurs moins récents, ne supportant pas HTML5 ou le format audiovisuel utilisé, le système utilise une deuxième implémentation à base d’un plugin Flash présentant la même interface que la version HTML5 native (voir § 4.10.2).

4.7.3 Accès aux données

L’accès aux données est réalisé par l’ensemble de lecteurs d’annotation instanciés lors de l’analyse du document qui se posent comme interfaces avec les composants en traduisant en contenu formaté le résultat des requêtes de lecture. En résumé, la lecture de données suit l’ensemble des étapes présentées dans le listing suivantes :

```

foreach component
{

```

```

Dataaccess = CHMComponentsManager.getDataAccess ( component );
contentSpec = dataaccess.query( DataAccess ,
component.getFilter ( ) ) ;
DataAccessManager.Content (component , contentSpec );
}

```

Listing 4.18 – Mécanisme de lecture de données

L'algorithme permettant de déterminer le contenu d'un composant passe d'abord par l'interrogation de son lecteur de données pour avoir le contenu d'annotation. Ce contenu peut avoir besoin de formatage et d'accès aux ressources. Le gestionnaire de composant détermine donc, à partir de la spécification, le contenu exact à assigner au composant.

4.7.4 Gestionnaire de composants

Les propriétés des composants CHM ainsi que leurs contenus sont définis par le *Gestionnaire de Composants (CHMComponentsManager)* qui, pour chaque composant, lit sa spécification, détermine les valeurs de ses attributs ainsi son contenu et ce, grâce aux réponses reçues suites à des requêtes vers les composants d'accès aux données. Par la suite, il garde trace de son évolution pour pouvoir le manipuler en tant qu'entité entière et non pas en tant qu'ensemble d'éléments HTML indépendants. Par exemple, un composant *map* peut être transformé en des dizaines d'éléments HTML <DIV>, <A>, , etc. La manipulation de la table doit rester consistants : un événement de *click* sur une entrée correspond à un événement de la table de matière et non pas sur une portion de texte ou une image.

D'une manière asynchrone, à chaque fois qu'un composant est défini, le code HTML correspondant est injecté dans le document final qui se génère dynamique par un mécanisme de réécritures et de transformations. Les éléments conteneurs sont traduits en blocs permettant de contenir le code des composants correspondant. La génération des composants est réalisée suivant un ensemble de modèles *templates* qui permettent une édition rapide de gros blocs de code. En effet, l'ossature générale d'un composant est d'abord définie et tous les items renvoyés par le lecteur d'annotations seront transformés selon cette *template*.

Exemple : Génération d'une table de matière

Pour montrer comment générer une table de matière avec deux niveaux, prenons l'exemple d'analyse filmique définissant un ensemble de types d'annotation relatifs à des émotions exprimées par les acteurs : joie - type *Joy* - et peur -type *Fear*-). Le code permettant de déclarer une table de matière est

```
<div id="test" chm:component='toc' chm:src='data'
chm:filter="level=1&&['Joy', 'Fear'];level=2&&type==['Joy', 'Fear'];"></div>
```

Listing 4.19 – Déclaration d'une table de matières

La source de données est le lecteur d'annotations `data`. Le premier niveau de la table est constituée des deux chaînes de caractère *Joy* et *Fear*. Le deuxième niveau représente tous les fragments de type *Joy* et *Fear* ; les références vers ces fragments seront placées dans le niveau approprié. Le moteur de transformation interroge le lecteur d'annotations ; le code exécuté par le lecteur d'annotations pour récupérer les données à partir du fichier JSON est le suivant :

```
$.getJSON(src, function(data) {
    Data= data.annotations;}
```

Listing 4.20 – Code exemple exécuté par un lecteur d'annotations

Le moteur de transformation interroge le lecteur de données en lui fournissant un ensemble de paramètres. Ainsi, pour récupérer les fragments, le code simplifié utilisé par le moteur de transformation est le suivant :

```
AnnotationReader.getData("type=='[Joy', 'Fear']");

```

Cet appel est bloquant au niveau de la transformation du composant. A la réponse, le formateur effectue un ensemble d'opérations permettant de construire la table de matière à partir de ses composants atomiques. En général, un niveau donné de la table a la structure suivante (listing 4.21) :

```
tag=<tr><td><a href='${media}#${begin}, ${end}' target='timedmedia'>${content}</a></td></tr>";
```

Listing 4.21 – Niveau d'une table de matière

Ce code peut être traduit en HTML par une ancre ayant pour destination un fragment et dont l'activation affecte le conteneur du média principal (via son lecteur vidéo) et par conséquent la ligne de temps.

4.8 Formatage spatial, temporel et synchronisation

4.8.1 Formatage spatial et mise en style

Nous adoptons les feuilles de style CSS pour la spécification spatiale des documents hypervidéos. La *layout*, les *régions spatiales* et les différents *composants visuels* sont décrits

par des classes et des attributs CSS qui contrôlent leurs dimensions et emplacements ainsi que leurs manifestations visuelles (couleurs, fonts, etc.). Avec les feuilles de style, des modèles de rendu (*templates*) peuvent être préparées et appliquées au besoin, selon les préférences de l'auteur et les contraintes de l'environnement.

Le formatage du document et la définition du style de présentation sont gérés par un module appelé *formateur*. Ce dernier calcule et ajuste les valeurs contenues dans les spécifications CSS permettant le placement des composants et la détermination du style d'affichage. Le rôle de ce module est principalement au démarrage de l'application et la mise à jour du document ; par la suite, il passe la main au module de rendu du navigateur pour exécuter et maintenir l'aspect visuel du document. Durant la présentation, il permet de mettre à jour la configuration des *layouts* et régions spatiales.

4.8.2 Gestion du temps et de la synchronisation

4.8.2.1 Référence de temps

La gestion du temps des composants est réalisée par des ordonnanceurs définis sur les composants *TimedMedia* et mettant en œuvre la notion de référence de temps (*TimelineRef*).

Une référence temporelle n'est pas déclarée explicitement, elle est plutôt virtuelle et liée à un et un seul composant *TimedMedia* - autrement dit, en déclarant un *TimedMedia*, une TLR est automatiquement définie -. Par défaut, une horloge globale est utilisée dans le document. Comme une référence peut dépendre d'une autre, nous appelons la référence globale *horloge du document*.

Pour qu'un composant soit lié à une référence de temps, il suffit de déclarer dans la valeur de l'attribut *chm:timelineref* l'identificateur du *TimedMedia* qui a permis de définir cette référence.

La définition des options de synchronisation au sein du composant est réalisée par un ensemble d'attributs : *chm:begin*, *chm:end*, *chm:duration*, *chm:loopcount*, *chm:canslideTLR* et *chm:finesync*.

4.8.2.2 Ordonnanceur

La coordination des différentes activités lors de la présentation d'une hypervidéo est assurée par le module "Ordonnanceur" qui gère la progression temporelle de la présentation en se référant à une TLR pour maintenir le temps associé aux médias. A chaque mise à jour du temps ou de l'état de la référence, il émet des événements et modifient les spécifications des composants *TLComponent* permettant de les synchroniser. Le principe

et les mécanismes suivant lesquels fonctionnent les ordonnanceurs hypervidéos peuvent être résumés comme suivant :

- le moteur d'exécution parse le document hypervidéo et construit la cartographie de ses composants TLComponent puis initialise un *Gestionnaire de Composants* pour chaque sous-document hypervidéo, identifié par son média principal *TimedMedia* ;
- pour chaque hypervidéo, un ordonnanceur est instancié avec la structure d'exécution correspondant à l'ensemble de composants ;
- la structure hypervidéo est analysée par l'ordonnanceur qui règle son horloge local à celui de la référence.
- au démarrage de la présentation, l'ordonnanceur démarre l'affichage et la présentation de chaque composant ;
- à chaque mise à jour du temps et/ou de l'état du *TimedMedia*, l'ordonnanceur règle l'état et le temps de chaque composant, sauf les composants qui ont l'attribut *chm:finesync* mis à faux ;
- l'ordonnanceur surveille aussi la progression de tous ses composants ayant l'attribut *chm:canslideTLR* mis à vrai. Ces composants permettent d'agir sur l'état et la progression de la référence de temps ;
- lors d'une désynchronisation du *TimedMedia*, tous les composants - et donc toute la présentation - sont mis à pause jusqu'à la resynchronisation du média ;
- lors d'une désynchronisation d'un autre composant, la resynchronisation est faite naturellement car à chaque mise à jour de la référence du temps, l'ordonnanceur essaie de régler l'état et le temps du composant. La désynchronisation d'un tel composant n'affecte pas le reste de la présentation ;
- à la fin de présentation du *TimedMedia*, tous les composants désynchronisés sont arrêtés. Si aucun composant n'est en cours de présentation , la présentation se termine.

4.8.2.3 Implémentation de la dimension temporelle par les Timesheets

La dimension temporelle étant absente du Web, nous faisons appel à la spécification SMIL Timesheets - très prometteuse et dont la syntaxe est bien maîtrisée car parfaitement définie par le W3C - pour réaliser le modèle temporel défini par CHM. Plusieurs implémentations des SMIL Timesheets sont actuellement en cours d'expérimentation. En se basant sur la logique montrée dans le projet Timesheet Scheduler [13] qui offre une bibliothèque JavaScript implémentant SMIL Timing et SMIL Timesheets avec la particularité d'être indépendante du lecteur audiovisuel utilisé, HTML5 ou Flash, nous mettons en œuvre le modèle temporel de CHM, contournant les difficultés de synchronisation des

médias du document sur le Web.

La mise en œuvre du concept de référence temporelle et des ordonnanceurs est facilitée par l'utilisation des Timesheet de SMIL 3. Ces *feuilles temporelles* ont les caractéristiques suivantes :

Un conteneur temporel SMIL (**par** ou **seq**) englobe un *TimedMedia* avec tous les composants qui lui sont reliés. De cette manière la synchronisation est facilitée.

L'attribut *syncMaster* de SMIL permet qu'un élément soit défini comme source de synchronisation des éléments définis dans ce conteneur. Automatiquement, le composant de rendu du *TimedMedia* est assigné cet attribut avec comme valeur *true*.

L'attribut *timeAction* de SMIL est utilisé pour spécifier le comportement que l'ordonnanceur effectuera quand l'instant de début d'un composant est atteint, correspondant à son activation. Par défaut, l'élément sera affiché dans la zone qui lui est prévu.

Les valeurs possibles pour cet attribut sont les suivants :

Valeur <i>timeAction</i>	Actions de l'ordonnanceur
"none"	L'élément cible est toujours visible. Spécifier cette valeur permet de faire sortir l'élément du contrôle de la référence de temps.
"display"	A l'activation de l'élément, sa propriété CSS 'display' est mise à 'block' sinon à 'none'.
"visibility"	A l'activation, la propriété CSS 'visibility' est à 'visible' sinon à 'hidden'.
"style"	A l'activation, l'ordonnanceur applique les spécifications de l'attribut <i>style</i> , et en dehors il les supprime.
"class : className"	L'ordonnanceur ajoute/supprime la classe <i>className</i> à la liste des classes de l'élément cible.

Tableau 4.2 – Valeurs possibles de l'attribut *timeAction*

Pour illustrer ces propos, prenons un exemple simple :

- Un *TimedMedia* défini sur une vidéo ;
- Deux paragraphes qui apparaissent respectivement aux instant 00s jusqu'à 40s et de 05.03 jusqu'à 15.03.

La traduction de cet exemple en HTML5 avec Timesheets serait :

```
<div smil:timeContainer = "excl" smil:timeAction = "display">
  <video smil:syncMaster = "true" src = "vid.webm"> </video>
  <p smil:begin = "0:00.00" smil:end = "0:40.00"> Paragraphe textuel </p>
  <p smil:begin = "0:05.03" smil:end = "0:15.03"> Deuxième Paragraphe textuel
</p>
</div>
```

Listing 4.22 – Exemple d'utilisation de l'attribut *timeAction* de SMIL

Temporalisation des composants Pour comprendre le mécanisme de temporalisation et d'ordonnancement exécuté par le moteur de rendu et son moteur temporel, nous allons analyser comment un composant standard peut être temporalisé. Prenons l'exemple du composant permettant le rendu des transcriptions du discours de l'orateur. Ce composant, comme vu, est un `textviewer` temporalisé (on lui a défini l'attribut `chm :timelineref`).

La transformation de contenu de ce composant, telle que vue dans 4.7.2, résulte en le code présenté dans le listing 4.23.

```
<div data-timecontainer="excl"
      data-timeaction="display" data-mediasync="video" style="display: block; ">

  [...]
  <p data-begin="29116" data-end="36223" style="display: none; ">Now, twenty years
  on, TED, I want to ask your help in a new reframement.</p>

  <p data-begin="36223" data-end="44701" style="display: none; ">So going back to
  1989, I wrote a memo suggesting global hypertext system.</p>

  <p data-begin="44701" data-end="47822" style="display: none; ">Nobody really did
  anything with it very much.</p>

  <p data-begin="47822" data-end="59507" style="display: none; ">But eighteen
  months later - this is, you know, this is how innovation happens, eighteen
  months later my boss said I could do it on-on a side, as a sort of a play(?)  

  project, ehr, (??) on the computer we'd got.</p>

  <p data-begin="59507" data-end="62413" style="display: none; ">And so he gave me
  the time to code it up.</p>

  <p data-begin="62413" data-end="75334" style="display: none; ">So, I basically
  roughed out what HTML looks like, the hypertext protocol, HTTP, the idea if
  URLs, these names for things which sorted(?) HTTP.</p>
  [...]
</div>
```

Listing 4.23 – Exemple de transformation de contenu

La transformation de contenu de ce composant permet de spécifier que des passages textuels (`<p>...</p>`) sont affichés de manière exclusive (attribut `data-timecontainer="excl"`).

Un bloc textuel n'est visible que durant sa période d'activation (attributs `data-begin` et `data-end`) et l'affichage d'un bloc annule celui d'un autre déjà affiché.

4.9 Comportement dynamique des hypervidéos

4.9.1 Gestionnaire d'interactions

La dynamicité de la présentation hypervidéo est induite principalement par sa progression dans le temps ainsi que par les événements de l'utilisateur lors des actions d'interaction et de navigation.

Le gestionnaire d'interactions permet de gérer les opérations de manipulation de document et de navigation. Il surveille la dynamique et la progression du document et permet de réagir aux opérations de navigation standard en contrôlant les aspects induits par la définition CHM des ancrés et des liens, tels que le retour après navigation.

Contexte d'une hypervidéo : le contexte est un cliché de la présentation à l'occurrence d'un événement. L'état d'implémentation actuel de *WebCHM* ne prend pas encore en charge cette notion de contexte, un des développements futurs du prototype, pouvant être implémentée comme suit.

Un contexte *courant* est automatiquement sauvegardé à chaque occurrence d'un certain type d'événements . Ainsi, toutes les opérations de navigation nécessitent la mise à jour du contexte courant. Une *pile de contexte* peut être définie dans le but de sauvegarder plusieurs contextes et de les restituer selon la politique LIFO. Ce contexte peut être utilisé pour la restitution de l'état du document, par exemple suite à l'activation d'un lien qui spécifie un retour à l'état précédent à la fin de la lecture de la destination. Ainsi, si un lien L1 fait changer le contexte C1 vers C2 et que, dans le contexte C2, un lien fait évoluer le contexte vers C3 et que, pour chaque lien, un comportement de retour vers la source est défini, il est possible de retrouver le contexte C1 à partir de C3. Ce contexte est particulièrement utile pour la définition des composants `history`.

4.9.2 Gestionnaire événementiel

En plus du mécanisme de navigation, le gestionnaire événementiel est responsable du comportement dynamique des présentations. Les événements hypervidéos définis dans CHM sont implémentés selon une architecture globale permettant de définir un ou plusieurs écouteurs d'événements (*event-listeners*) sur les composants.

Gestionnaires d'événements : à la déclaration d'un événement, le gestionnaire événementiel lui définit un écouteur (*listener*) qui se charge de surveiller l'occurrence de l'événement et d'appeler la routine permettant de réaliser les actions appropriées.

Exemple : Le code suivant permet de définir un événement sur un click sur le composant. Le traitement est défini par la fonction invoquée.

```
<div chm:component="event" id="ev1" chm:src="component1"
chm:eventtype="domevent.click"
chm:do="treatment();"/>
```

Le gestionnaire d'événements instancie un nouveau *listener* de l'événement *click* sur le composant *component1*. En JQuery, un tel *listener* est créé par une simple fonction :

```
$( 'component1' ).addEventListener ( 'click' , treatment);
```

A chaque fois que l'événement *click* est détecté, la fonction *treatment()* est appelée.

Plusieurs types d'événements standards peuvent être utilisés, par exemple :

- les événements liés au mécanisme de présentation : *load* (chargement d'un élément du document), *reload* (rechargement), *focus* (l'élément gagne le focus) ;
- les événements liés aux actions temporelles : *finish* (l'élément se termine) ;
- les événements liés à la structure du document : *new* (la création de la structure de donnée associée à l'élément est terminée), *change* (un champ de la structure de l'élément a changé) ;
- les événements utilisateurs : *click* (clic sur un élément), *drag* (déplacement d'un élément), *keydown* (appui sur une touche).

4.10 Interface de présentation

4.10.1 Templates de rendu

Alors que *WebCHM* libère le concepteur d'hypervidéo de la rigidité de certains systèmes multimédia, nous avons fourni avec les librairies un ensemble de choix par défaut lui permettant la production rapide d'hypervidéos. Les choix retenus sont issus principalement des expériences dans le domaine de la vidéo interactive sur Internet. La mise en œuvre prévoit un ensemble de modèles (*templates*) CSS que l'auteur peut utiliser directement ou modifier. Ces modèles font parties du prototype et définissent l'allure générale et finale des composants. L'exemple du rendu du scénario, présenté à la figure 4.6 montre pour les composants présentés le style qui leur est appliqué.

The screenshot shows a web browser window with the URL liris.cnrs.fr/advene/examples/tbl_linked_data_html5/transcript.html. The page has a header with links to Home, Transcription, Summary, Making of, and RDF. Below the header is a section titled "Transcription of the talk". Under "Table of contents", there is a bulleted list of topics. To the right of the transcription is a video player window titled "Playing from 00:00:47.8" showing a man (Tim Berners-Lee) speaking on stage. The video player interface includes a progress bar at the bottom.

Figure 4.5 – Lecteur CHM dans le contexte d'un export Web de Advène

4.10.2 Lecteur vidéo sur Web *CHMVideoPlayer*

Dans le cadre de notre implémentation du modèle CHM, nous avons développé un lecteur vidéo sur Web, **CHMVideoPlayer**, qui exploite les nouveaux éléments HTML5 avec un ensemble de fonctionnalités permettant de proposer un lecteur hypervidéo avec plus de capacités que ceux existant. Ce lecteur est développé comme librairie JQuery utilisable dans le cadre du projet CHM, du prototype Advène mais aussi indépendamment⁵⁷.

Le prototype Advène peut exporter ses projets en différents formats. Il permet le développement d'hypervidéos à base d'annotations, en exportant ses recueils et ses vues en tant que sites Web pour être visualisés sur navigateur, en utilisant un lecteur vidéo de type *Google Video* ou *Youtube*. Dans le cadre de ce mémoire, durant un premier temps, nous avons implémenté un module *python* - le prototype Advène est développé en ce langage - d'export Web permettant de réaliser des formes basiques des hypervidéos en intégrant un lecteur vidéo HTML5/Flash dans ces exports Web avec la possibilité de définir des liens vers des fragments de données, remplaçant ainsi les lecteurs conventionnels. En plus des vidéos pouvant être rendues en HTML5 en natif, pour des formats non pris en charge ou pour des navigateurs sans support HTML5, ce lecteur *CHMVideoPlayer* utilise un plugin

57. *CHMVideoPlayer* est par exemple utilisé sur le site du projet *museographie* du Laboratoire interuniversitaire des Sciences de l'Education et de la Communication de l'Université de Strasbourg (voir http://www.museographie.fr/exemple/Tous_les_cours_d_action.html).

Flash. Le code du lecteur Flash est inspiré du projet *MediaElement.js*⁵⁸ que nous avons adapté pour présenter la même interface que celle HTML5. Ainsi, une haute abstraction logicielle est offerte. En effet, des mécanismes sont implémentés pour prendre en charge le lecteur vidéo CHM qu'il soit en HTML5 natif ou en Flash, pour un meilleur support des anciens navigateurs. Ce module est intégré au logiciel Advene depuis la *version 0.44*. La figure 4.5 montre l'utilisation du lecteur dans le contexte d'un export Web d'Advene, pouvant être expérimentés sur le site du projet Advene.

Le lecteur *CHMVideoPlayer* a plusieurs caractéristiques intéressantes :

- Flash n'est utilisé qu'au besoin et uniquement pour la lecture des médias continus : l'ensemble de la page Web reste en CHM (HTML/CSS/Timesheet) ;
- la couche d'abstraction logicielle permet pour une présentation hypervidéo écrite pour le HTML5 d'être présentée sur tous les navigateurs ;
- contrairement à ce que l'on trouve actuellement, l'implémentation du lecteur lui permet d'être flottant, n'occupant pas une région particulière de la fenêtre d'affichage, "redimensionnable" (*resizable*) et déplaçable à la souris (*draggable*) ;
- le lecteur peut n'être présenté que si un lien un fragment vidéo est détecté ;
- le lecteur permet l'affichage de graphiques, du texte et toute sorte d'*overlays* sur l'interface de la vidéo ;
- le lecteur peut réaliser un montage à la lecture, en spécifiant les portées temporelles des fragments à lire ;
- toutes les fonctions classiques d'un lecteur vidéo sont présentes, avec une interface simple et intuitive.

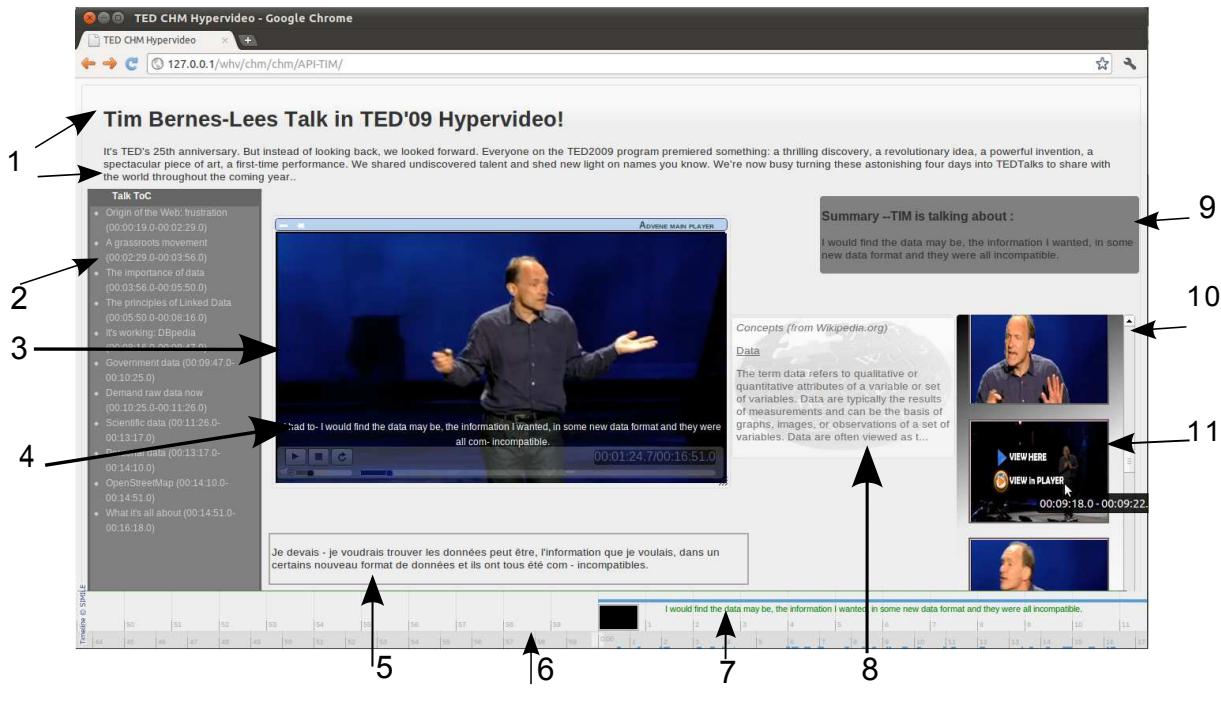
Bien que le lecteur vidéo CHM soit principalement développé pour la lecture des vidéos principales, une version du lecteur (*lecteur vignette*) est développée spécifiquement pour les besoins des clichés *thumbnail video* qui sont des vidéo en vignettes généralement de petites tailles qui ont moins d'options que le lecteur principal.

4.10.3 Exemples d'éléments de rendu hypervidéos

La figure 4.6 montre le prototype *WebCHM* en action : la figure 4.6(a) montre le rendu de l'hypervidéo du scénario que nous avons développé tout au long de ce chapitre, alors que la figure 4.6(b) illustre comment une *hotspot* est affichée au survol du pointeur de la souris, donc en exécutant l'action associée à l'événement du survol de la souris quand un lien spatio-temporel est disponible.

Plusieurs composants sont illustrés sur ces deux figures :

58. mediaelementjs.org



(a) Rendu de l'hypervidéo TED'09



(b) Exemple de rendu du lecteur. A gauche le lecteur montrant une hotspot ; à droite, lecteur vignette

Figure 4.6 – WebCHM en action

- L’élément (1) n’est pas un composant hypervidéo mais plutôt un contenu HTML standard.
- L’élément (2) représente la table de matière (composant **TableOfContent**) de l’hypervidéo contenant les parties du discours présentées sous forme textuelle estampillée par l’intervalle temporel de la partie. L’activation d’une entrée permet d’aller directement dans le segment pointé ; résultant en un décalage temporel de toute la présentation.
- L’élément (3) présente un lecteur vidéo (composant **VideoPlayer**) principal, comportant une vidéo annotée (un composant **TimedMedia**).
- L’élément (4) présente le sous-titrage accompagnant la vidéo présentée (composant **TextViewer** avec référence de temps - donc, un **TLComponent**) et synchronisé avec le flux.
- L’élément (5) présente la traduction en français (composant **TextViewer** temporaialisé) du discours prononcé par l’orateur et affiché comme sous-titre en anglais, d’une manière synchronisée.
- L’élément (6) représente la ligne de temps (composant **TimelineRefGUI**) de la vidéo et donc de toute la présentation étant donné que la vidéo est le seul composant inférant une référence temporelle. Cet élément permet de naviguer dans le temps.
- L’élément (7) représente des pistes (composants **Track**) affichées tout au long de la ligne de temps et contenant un descriptif et une image - si disponible - du contenu d’annotation associé. En activant (cliquant) sur une piste, l’instant correspondant de la présentation est déclenché.
- L’élément (8) représente des données récupérées à partir de l’encyclopédie en ligne *Wikipedia.org* et présentées sous formes textuelles riches (composant **RichTextViewer** temporaialisé) synchronisée.
- L’élément (9) présente un résumé textuel enrichi et synchronisé (composant **RichTextViewer** temporaialisé) de la partie en cours de présentation.
- L’élément (10) présente un ensemble d’images clés (appelées vignettes) de début des parties du discours (composant **map**).
- L’élément (11) présente une image-clé ou vignette permettant de visualiser la partie correspondant de deux manières : soit en déclenchant la lecture de la vidéo principale à l’instant correspondant - résultant en un décalage temporel de toute la présentation - ou de pré-visualiser, voire visualiser uniquement le fragment correspondant sur une version simplifiée du lecteur vidéo et qui est présente au même endroit que la vignette (la remplaçant). A la fin de la lecture du «player vignette», ce dernier se ferme automatiquement et l’image réapparaît.

- L’élément (12) présente une région sensitive (composant *hotspot*) permettant d’aller vers le lien énoncé par l’orateur.
- L’élément (13) montre la version simplifiée du lecteur vidéo (*player vignette*).

L’exemple du scénario a permis d’illustrer l’implémentation de plusieurs composants. D’autres formes plus complexes des hypervidéos peuvent être mises en œuvre. Ainsi, l’exemple de synchronisation entre plusieurs sous-documents hypervidéos permettrait de mieux appréhender les mécanismes de synchronisation et de visualisation offerts. D’autre part, les *overlays* présentées sont principalement textuelles, sauf le cas des *hotspots*. Un contenu plus riche peut être défini ; par exemple des figure, du contenu hétérogène voire des vidéos présentées par le *player vignette*. Bien qu’à première vue, de telles hypervidéos sont assez complexes, leur édition est assez simple, moyennant des feuilles de style.

4.11 Discussion

Une première version de *WebCHM* a été développée comme validation des principes énoncés. Encore partielle, son état présent a mis en action la plupart des idées énoncés dans CHM et d’autres idées telles que les concepts de contexte et son exploitation par le composant *historique* sont encore à intégrer.

Cette implémentation Web du modèle CHM suit une approche déclarative pour permettre de greffer les concepts spécifiques aux hypervidéos comme attributs au langage HTML. Ceci permet une création d’hypervidéos plus aisée comparée aux solutions courantes. Des hypervidéos complexes peuvent être écrites facilement, *stylées* avec CSS, animées par SVG et contrôlées par des scripts et ce, grâce à un ensemble de librairies JavaScript génériques, réutilisables, extensibles tout en étant transparentes pour l’utilisateur (auteur et lecteur) et conformes aux normes et standards du Web. La syntaxe déclarative permet de décrire ce qui est attendu plutôt que de spécifier comment obtenir le comportement voulu et la maintenance évolutive de l’application hypervidéo est grandement simplifiée puisqu’aucun code JavaScript spécifique n’est requis du côté de l’auteur pour une utilisation standard. Ainsi, la rédaction d’un document hypervidéo est aussi facile que la création de n’importe quelle page Web.

Bien que SMIL soit très attrayant pour le multimédia, il souffre de son manque d’appliquabilité voire du désintérêt que lui témoignent les constructeurs de navigateurs. Même les systèmes dédiés à SMIL ne l’ont pu l’implémenter intégralement. Les modules de temporalisation et de synchronisation sont les plus matures de ce langage et pour contourner les difficultés d’ajouter la notion du temps au Web actuel, nous avons voulu tirer profit des possibilités inégalées de SMIL et pour cela, nous nous sommes intéressés au TimeSheets

qui *offrent SMIL à HTML et XML* en général.

Du point de vue utilisation du multimédia sur le Web, la plupart des projets utilisant le nouveau standard Web comme *popcorn.js* se réfère exclusivement au temps induit par les éléments `<audio>` et `<video>` pour ordonner la présentation. Comme discuté, cela est principalement dû aux faiblesses induites par la spécification HTML5 qui ne définit pas de conteneur temporel. Dans *WebCHM*, la mise en œuvre de la temporalisation et la synchronisation définies par le modèle temporel de CHM permet la création sur le Web de document parfaitement multimédia.

Le projet *popcorn.js* qui introduit des idées innovantes est celui qui nous semble le plus proche du nôtre, du point de vue fonctionnalités. Néanmoins, comme la plupart des systèmes manipulant la vidéo interactive et les hypervidéos, le manque de formalisme le réduit à une environnement qui propose beaucoup plus des illustrations de capacités de HTML5 que de création d'hypervidéos à base théorique solide.

Grâce à l'utilisation des annotations, une séparation manifeste entre les données et leur présentation est rendue possible. Ceci a permis de développer des outils documentaires génériques basés sur un modèle explicite. Les méta-données et les annotations permettent d'entrevoir d'autres manières de structurer et de présenter les contenus sur le Web.

Enfin, nous croyons que notre utilisation de la norme HTML5 peut être une nouvelle brique illustrant les utilisations possibles offerts par les éléments multimédia de la nouvelle spécification HTML vers des utilisations innovantes de la vidéo sur le Web.

4.12 Conclusion

CHM a été mis en œuvre pour vérifier les assumptions du modèle. Le prototype *WebCHM* propose une syntaxe conforme aux exigences du Web pour augmenter ce dernier par des possibilités nouvelles de présentation de documents à base audiovisuelle. L'implémentation du modèle CHM permet de mieux expérimenter d'autres formes d'utilisation de documents audiovisuels en ligne et d'aborder les défis qui jalonnent une mise en œuvre effective et plus élaborée du multimédia.

En greffant les possibilités offertes par les hypervidéos au Web et en offrant aux utilisateurs le moyen de créer les hypervidéos et de les diffuser, le prototype confirme que d'autres utilisations du Web sont possibles. Ainsi, on peut conclure que, sans une théorisation préalable et une formalisation des concepts, les barrières techniques ne sont pas ce qui entrave le plus le développement de nouvelles idées et technologiques et d'autres orientations à celles déjà répandues.

Conclusion Générale

1 Synthèse et rappel des objectifs

La problématique abordée dans ce mémoire porte sur la proposition d'un formalisme permettant de décrire et de représenter les documents hypervidéos issus de l'intégration de la vidéo interactive aux hypermédias traditionnels. Alors que la vidéo interactive n'est plus un dispositif expérimental, il y a peu de travaux réalisés dans le but de formaliser les hypermédias à base audiovisuelle permettant l'émergence de l'hypervidéo comme canal de distribution de ce genre attractif et complexe de documents audiovisuels.

L'intérêt que portent différentes communautés aux systèmes d'information audiovisuels est sans appel. L'interactivité et l'usage intensif de l'annotation étant moyens d'appropriation du document, s'avèrent très prometteurs pour les documents audiovisuels. Or, l'intégration de ces documents dans des espaces informationnels hétérogènes crée de nouveaux besoins de conceptualisation voire de normalisation des concepts et des formalismes de représentations.

Aujourd'hui, il y a peu de barrières techniques entravant le développement d'hypervidéos à grande échelle et un nombre important d'expérimentations de la vidéo interactive a été rendues publiques (on peut citer naturellement *Youtube* et ses services - commentaires, annotations, interactivité). Des travaux ont ciblé directement ou indirectement le développement d'hypervidéos ; néanmoins, force est de constater que cela n'a pas été accompagné réellement et au préalable de fondements et de fondations théoriques clairs et solides.

Dans ce mémoire, la discussion des aspects et des domaines liés aux documents hypervidéo (vidéo interactive, hypermédia, multimédia et annotations), qui sont arrivés à maturité si pris séparément, démontre que les hypervidéos ont encore besoin d'un modèle *intégrateur* capable de représenter toutes leurs spécificités, leurs manifestations et les concepts reliés. Ceci constitue l'objet essentiel traité dans ce mémoire.

2 Bilan scientifique

2.1 Aspect *recherche*

Une modélisation des documents hypervidéos a été décrite dans ce mémoire. Le modèle proposé est à base d'annotations et représente les éléments logiques constitutifs d'un document par un ensemble de composants formés et reliés hiérarchiquement. Ces composants matérialisent les concepts de base des hypervidéos et l'ensemble des relations, non exhaustives, entre eux.

Pour cerner les différentes facettes de ce type de documents, nous avons cru utile de les représenter suivant les quatre dimensions du multimédia augmentées par une gestion dynamique à base d'événements et une structuration à base d'annotations.

Le modèle logique permet de décrire l'hypervidéo en tant qu'ensemble de composants conceptuels allant du composant de plus bas niveau à ceux assez complexes comme les tables de matière et les lignes de temps graphiques.

Le modèle spatial décrit le positionnement des différents éléments visuels d'un document. Pour cela, par souci de simplicité et de généréricité, nous avons considéré un modèle absolu. Notons que d'autres modèles peuvent se greffer sur ce dernier.

Le modèle temporel décrivant la temporalisation et la synchronisation des éléments d'un document est à base de *timeline* et introduit la notion de *TLR* pour assujettir tout le document à l'information audiovisuelle principale.

Le modèle hypermédia introduit une représentation plus complexe avec la notion d'ancres et de liens spatio-temporels et les fonctions de navigation avancées spécifiques aux hypervidéos.

La gestion dynamique du contenu est réalisée par un modèle événementiel permettant la spécification d'une manière très fine des comportements dynamiques caractérisant les hypervidéos.

L'approche à base d'annotation sur laquelle s'articule le modèle permet une réelle séparation du contenu de son rendu et des possibilités de visualisation et de génération de présentations et des représentations aussi riches que diverses. Bien qu'un modèle minimal soit exigé et que nous nous sommes focalisés durant la phase de mise en œuvre sur le prototyp Advene, la structure des annotations est totalement indépendante de toute plateforme et de tout système.

2.2 Aspect *développement*

Les modèles sont des représentations formelles qui servent principalement de repères (*guidelines*) pour le développement des systèmes et d'applications les mettant en œuvre dans un double but : la validation des théories et l'introduction de nouvelles pratiques technologiques. Pour éviter l'écueil des propositions restant à l'état théorique, nous nous sommes intéressés à valider le modèle proposé dans un environnement réel, établi, assez répandu et largement déployé : le Web. Cela nous a mis devant de nouveaux défis dûs principalement au fait que le Web actuel est un des environnements hypermédias qui répondent le moins aux exigences des modèles théoriques. La mise en œuvre de CHM a permis de revoir la théorie du Web et de discuter de quelques idées permettant l'émergence d'un Web qui considère le multimédia dans toutes ses spécificités.

La discussion des exigences d'un hypermédia centré sur la vidéo focalisé sur le Web comme environnement nous a conduit à la présomption que les hypervidéo en tant que technologie pourraient véhiculer des réponses théoriques et techniques pour ces besoins faisant en sorte que la vidéo devienne effectivement une entité de première classe sur le Web ("*video as a Web first class citizen*"). Les choix technologiques que nous avons opérés ont été motivés justement par le besoin d'être conformes au Web actuel et nous nous sommes focalisés sur une implémentation mettant en jeu la technologie la plus basique et la plus répandu : un mélange de HTML, CSS et JavaScript. Écrire une hypervidéo complexe devient aussi simple que l'édition d'une page Web conventionnelle.

Une syntaxe déclarative mettant en œuvre les concepts du modèle CHM a été proposée comme extension du langage HTML. Nous avons eu un intérêt particulier pour la nouvelle spécification de HTML et ses éléments multimédia que nous avons mis en œuvre, ce qui constituerait une brique de plus dans le processus d'adoption de ce nouveau format.

Bien qu'en cours d'état de développement et d'extension, le prototype WebCHM a été l'occasion d'expérimenter de nouvelles idées et d'ouvrir d'autres perspectives pour un Web totalement conforme aux exigences multimédia en général.

3 Perspectives

Au cours de ce mémoire, nous avons eu à débattre de plusieurs technologies et à explorer de nouvelles idées. Mené en parallèle avec la recherche théorique, et pour être en phase avec les évolutions actuelles, la réalisation de WebCHM a nécessité un suivi continu des nombreuses propositions et recommandations régissant l'évolution du Web. Cela a eu comme conséquence d'avoir des idées plus claires sur un ensemble de domaines.

Nos perspectives concernent en premier abord les domaines liés au multimédia et au Web, qui semble s'orienter de plus en plus à devenir le support de prédilection d'un large éventail d'applications multimédia ; en témoigne la spécification de la dernière version de son standard (HTML). Dans ce sens, nous croyons que le présent travail peut être amélioré et étendu suivant plusieurs axes, que nous pouvons regrouper en deux catégories principales : *recherche* et *développement*.

3.1 Aspect *Recherche*

Etude des modes de perception de l'information audiovisuelle : l'intérêt principal est de chercher à comprendre et à modéliser la synergie résultant de la combinaison de l'audiovisuel avec l'interactivité et de comprendre comment cela influe la réalité de l'information et quels sont les influences du point de vue théories cognitives et modes de perception de l'information et de partage de la connaissance. Cela permettrait de mieux appréhender les défis esthétiques et rhétoriques des hypervidéos.

Aspect collaboratif : la collaboration à base d'hypermédias est au cœur du domaine des *hypermédias collaboratifs* [80]. Dans la même logique, les hypervidéos *collaboratives* peuvent être définies avec pour objectif de faciliter, garantir et proposer aux utilisateurs (concepteur ou lecteurs) un environnement qui intègre des outils de collaboration et qui permettent entre autres d'annoter et de générer des présentations d'une manière collaborative : en local ou à distance, synchrone ou asynchrone, collective ou distribuée.

TV interactive (IDTV) : une des applications possibles des hypervidéos peut être la télévision numérique interactive (IDTV) qui met en exergue la possibilité de personnalisation voire d'enrichissement d'un contenu au moment de sa retransmission de telle manière à ce que les effets induits sont reflétés instantanément. Les exigences de tels systèmes hypervidéos doivent être étudiés et des modèles doivent être conçus pour permettre l'expérimentation de telles idées et la mise en œuvre de systèmes hypervidéos qui ne se basent pas sur un flux enregistré mais plutôt diffusé en temps réel.

Modèle multimédia et adaptation pour de larges perspectives et une plus grande flexibilité et puissance d'expressivité, les modèles multimédia de CHM peuvent être largement améliorées. Par exemple, les dimensions spatiale et temporelle peuvent être étendues par des modèles relationnels. D'autre part, bien que des solutions ad hoc peuvent être mises en œuvre pour adapter une hypervidéo CHM, l'extension de ses modèles mul-

timédia permettrait de définir des stratégies d'adaptation dynamique selon le contexte et l'environnement cible.

Multimédia et annotations sur le Web nous avons discuté de l'apport des annotations à une certaine classe de documents multimédia et nous avons expérimenté leur utilisations sur le Web. Nous pensons utile d'étendre cette étude pour modéliser à base d'annotations tout type de document multimédia et étudier les exigences et répercussions que le Web comme environnement pourra induire.

3.2 Aspect *Développement*

Développer le maximum de module de manière générique : actuellement, nous sommes en train de compléter le développement du prototype WebCHM. En envisageant une plus grande utilisation de ses librairies, nous comptons pouvoir développer tous les modules et de les offrir sous une forme générique prêtes à l'utilisation et réutilisation.

Spécification DOM : actuellement, la génération des hypervidéos est réalisée par un ensemble de modules qui effectuent une transformation de contenu. La prochaine étape de développement serait l'implémentation de la spécification des interfaces DOM *Document Object Model*. Ceci permettrait l'édition hypervidéo d'une manière beaucoup plus consistante en offrant une interface unifiée et indépendante de toute plateforme.

Éditeur CHM sous advene : le prototype Advene est muni d'un éditeur graphique HTML qui permet de composer des vues statiques des recueils manipulés. Il peut être utilisé comme éditeur textuel des hypervidéos CHM. Néanmoins, pour une édition rapide, nous projetons d'intégrer les composants CHM en natif dans l'éditeur pour pouvoir manipuler visuellement ces composants.

Éditeur graphique (GUI) : dans la même logique que le point précédent, l'idée d'un éditeur graphique des documents CHM est très intéressante. Elle permettra de manipuler tout type de composant CHM et d'étendre les spécifications pour les experts d'une manière rapide et conforme aux spécifications du modèle.

Bibliographie

- [1] James F. ALLEN : Maintaining knowledge about temporal intervals. *Commun. ACM*, 26:832–843, November 1983.
- [2] Olivier AUBERT et Yannick PRIÉ : Documents audiovisuels instrumentés. temporalités et détemporalisations dans les hypervidéos. *Document Numérique*, 8(4):143–168, 2004. Numéro spécial "Temps et documents".
- [3] Olivier AUBERT et Yannick PRIÉ : From video information retrieval to hypervideo management. In *Corimedia, the international workshop on multidisciplinary image, video, and audio retrieval and mining*, Sherbrooke, Canada, Oct 2004. 10 pp.
- [4] Olivier AUBERT et Yannick PRIÉ : Advene : active reading through hypervideo. In *Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, ACM Hypertext '05, pages 235–244, New York, NY, USA, 2005. ACM.
- [5] Bruno BACHIMONT : Indexation audiovisuelle : une problématique en pleine évolution. *L'OBJET*, 6(2), 2000.
- [6] Walter BENDER : Animation via video disk. Mémoire de D.E.A., Massachusetts Institute of Technology, USA, 1980.
- [7] Susanne BOLL, Wolfgang KLAS et Utz WESTERMANN : A comparison of multimedia document models concerning advanced requirements. Rapport technique, 1999.
- [8] M. BOVE, J. DAKSS, S. AGAMANOLIS et E. CHALOM : Hyperlinked television research at the mit media laboratory. *IBM System Journal*, 39, 2000.
- [9] Sandra BRINGAY, Catherine BARRY et Jean CHARLET : Les documents et les annotations du dossier patient hospitalier. *Revue i3*, 4(1), 07 2004.
- [10] Gwendal Auffret BRUNO BACHIMONT, Thomas Dechilly et Vincent BRUNIE : Représentation et structuration des connaissances pour les bibliothèques audiovisuelles. *Document numérique*, pages 194–214, 1999.
- [11] Dick BUTERMAN : Synchronized multimedia integration language (SMIL 3.0). W3C recommendation, W3C, 2008. <http://www.w3.org/TR/SMIL3/>.

-
- [12] Dick C.A. BULTERMAN et Lloyd W. RUTLEDGE : *SMIL 3.0 : Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books*. Springer Publishing Company, Incorporated, 2008.
 - [13] Fabien CAZENAVE, Vincent QUINT et Cécile ROISIN : Timesheets.js : When SMIL meets HTML5 and CSS3. *In to appear in Proceedings of DocEng2011*, 2011.
 - [14] Pablo CESAR, Dick C. A. BULTERMAN, Jack JANSEN, David GEERTS, Hendrik KNOCHE et William SEAGER : Fragment, tag, enrich, and send : Enhancing social sharing of video. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(3):1–27, 2009.
 - [15] T. CHAMBEL, N. CORREIA et N. GUIMARAES : Hypervideo on the web : models and techniques for video integration. *International journal of computers & applications*, 23(2):90–98, 2001.
 - [16] Teresa CHAMBEL : Context perception in video-based hypermedia spaces. *Information Systems*, pages 85–94, 2002.
 - [17] Teresa CHAMBEL et Nuno GUIMARAES : Learning with video in hypermedia, 2001.
 - [18] Teresa CHAMBEL et Nuno GUIMARAES : Context perception in video-based hypermedia spaces. *In Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 85–94, College Park, Maryland, USA, 2002. ACM Press.
 - [19] Teresa CHAMBEL, Carmen ZAHN et Matthias FINKE : Hypervideo and cognition : Designing video-based hypermedia for individual learning and collaborative knowledge building. *In Eshaa Alkhalfa (Ed.), éditeur : Cognitively Informed Systems : Utilizing Practical Approaches to Enrich Information Presentation and Transfer*, pages 26–49. 2006.
 - [20] J. CONKLIN : Hypertext : an introduction and survey. *IEEE Computer Magazine*, 20(9):17–41, 1987.
 - [21] Romain DELTOUR et Cécile ROISIN : The limsee3 multimedia authoring model. *In Proceedings of the 2006 ACM symposium on Document engineering*, DocEng ’06, pages 173–175, New York, NY, USA, 2006. ACM.
 - [22] Max J. EGENHOFER : Reasoning about binary topological relations. *In Proceedings of the Second International Symposium on Advances in Spatial Databases*, SSD ’91, pages 143–160, London, UK, 1991. Springer-Verlag.
 - [23] Roberto FAGÁ, Vivian Genaro MOTTI, Renan Gonçalves CATTELAN, Cesar Augusto Camillo TEIXEIRA et Maria da GRAÇA CAMPOS PIMENTEL : A social approach to authoring media annotations. *In 10th ACM symposium on Document engineering*, DocEng ’10, pages 17–26, 2010.

-
- [24] M. FINKE : Mobile interactive video. Rapport technique, IST-2001-37365, MUMMY Project, 2004.
 - [25] Daniela FOGLI, Giuseppe FRESTA, Piero MUSSIO, Andrea MARCANTE et Marco PADULA : Annotation in cooperative work : from paper-based to the web one. In *IWAC*, pages 1–10, 2005.
 - [26] International Organization for STANDARDIZATION : Information technology – hypermedia/time-based structuring language (hytime). ISO Standard, August 1997.
 - [27] Wen GAO, Yonghong TIAN, Tiejun HUANG et Qiang YANG : Vlogging : A survey of videoblogging technology on the web. *ACM Comput. Surv.*, 42:15 :1–15 :57, June 2010.
 - [28] Jörg GEISSLER : Surfing the movie space : advanced navigation in movie-only hypermedia. In *Proceedings of the third ACM international conference on Multimedia, MULTIMEDIA '95*, pages 391–400, New York, NY, USA, 1995. ACM.
 - [29] J. P. T. M. GEURTS : *A Document Engineering Model And Processing Framework For Multimedia Documents*. Thèse de doctorat, Technische Universiteit Eindhoven,, 2010.
 - [30] Andreas GIRGENSOHN, Frank SHIPMAN et Lynn WILCOX : Hyper-hitchcock : authoring interactive videos and generating interactive summaries. In *MULTIMEDIA '03 : Proceedings of the eleventh ACM international conference on Multimedia*, pages 92–93, New York, NY, USA, 2003. ACM.
 - [31] Daniel R. GOLDMAN : *A Framework for Video And Annotation, Visualization and Interaction*. Thèse de doctorat, University of Washington, Graduate School, 2007.
 - [32] N. GUIMARES, T. CHAMBEL et J. BIDARRA : From cognitive maps to hypervideo : Supporting flexible and rich learner-centred environments. *Educational Technology Research and Development*, 2(2), 2000.
 - [33] Frank HALASZ et Mayer SCHWARTZ : The dexter hypertext reference model. *Commun. ACM*, 37(2):30–39, 1994.
 - [34] R. I. HAMMOUD : Introduction to interactive video. In *Interactive Video*, pages 3–24. Springer-Verlag, 2006.
 - [35] Frank Allan HANSEN et Bent Guldbjerg CHRISTENSEN : Xlink-bridging open hypermedia and the world wide web. Mémoire de D.E.A., University of Aarhus, Arhus, Denmark, January 2003.
 - [36] Lynda HARDMAN, Dick C. A. BULTERMAN et Guido van ROSSUM : The amsterdam hypermedia model : adding time and context to the dexter model. *Commun. ACM*, 37(2):50–62, 1994.

-
- [37] Jon HEGGLAND : Ontolog : Temporal annotation using ad hoc ontologies and application profiles. *In Proceedings of the 6th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL '02, pages 118–128, London, UK, UK, 2002. Springer-Verlag.
 - [38] Kyoji HIRATA, Yoshinori HARA, Hajime TAKANO et Shigehito KAWASAKI : Content-oriented integration in hypermedia systems. *In HYPERTEXT '96 : Proceedings of the the seventh ACM conference on Hypertext*, pages 11–21, New York, NY, USA, 1996. ACM.
 - [39] Peter HOFFMANN et Michael HERCZEG : Hypervideo vs. storytelling integrating narrative intelligence into hypervideo. *In TIDSE*, pages 37–48, 2006.
 - [40] Mitsubishi America INC. : Authoring with hyperlinked video, whitepaper by mitsubishi electric america inc., 2000.
 - [41] Ramesh JAIN et Arun HAMPAPUR : Metadata in video databases. *SIGMOD Rec.*, 23:27–33, December 1994.
 - [42] Jack JANSEN et Dick C. BULTERMAN : Smil state : an architecture and implementation for adaptive time-based web applications. *Multimedia Tools Appl.*, 43:203–224, July 2009.
 - [43] Jack JANSEN, Pablo CESAR et Dick C.A. BULTERMAN : A model for editing operations on active temporal multimedia documents. *In Proceedings of the 10th ACM symposium on Document engineering*, DocEng '10, pages 87–96, New York, NY, USA, 2010. ACM.
 - [44] Muriel JOURDAN, Nabil LAYAIDA et Loay SABRY-ISMAIL : Madeus : An authoring environment for interactive multimedia documents. *In Proceedings of the 1997 International Conference on Multimedia Computing and Systems*, pages 267–272, Washington, DC, USA, 1997. IEEE Computer Society.
 - [45] Peter KING, Patrick SCHMITZ et Simon THOMPSON : Behavioral reactivity and real time programming in xml : functional programming meets smil animation. *In Proceedings of the 2004 ACM symposium on Document engineering*, DocEng '04, pages 57–66, New York, NY, USA, 2004. ACM.
 - [46] Michael KIPP : Anvil - a generic annotation tool for multimodal dialogue, 2001.
 - [47] F. KOKKORAS, H. JIANG, I. VLAHAVAS, A. K. ELMAGARMID, E. N. HOUSTIS et W. G. AREF : Smart videotext : a video data model based on conceptual graphs. *Multimedia Syst.*, 8:328–338, July 2002.

-
- [48] Rodrigo LAIOLA GUIMARÃES, Pablo CESAR et Dick C.A. BULTERMAN : Creating and sharing personalized time-based annotations of videos on the web. *In Proceedings of the 10th ACM symposium on Document engineering*, DocEng '10, New York, NY, USA, 2010. ACM.
 - [49] George P. LANDOW : *The rhetoric of hypermedia : some rules for authors*, pages 81–103. MIT Press, Cambridge, MA, USA, 1991.
 - [50] P. LÉVY : *Les technologies de l'intelligence*. Sciences et société. La Découverte, 1990.
 - [51] Gunnar LIESTOL : Aesthetic and rhetorical aspects of linking video in hypermedia. *In ECHT '94 : Proceedings of the 1994 ACM European conference on Hypermedia technology*, pages 217–223, New York, NY, USA, 1994. ACM.
 - [52] Craig LOCATIS, James CHARUHAS et Richard BANVARD : Hypervideo. *Educational Technology Research and Development*, 38(2):41–49, 1990.
 - [53] Pollone M., Rusconi M. et Tua R. : From hyper-film to hyper-web : The challenging continuation of a european project. *In Electronic Imaging & the Visual Arts Conference*, Florence, Italia, 2002. ACM.
 - [54] Azze-Eddine MAREDJ : *Spécification et validation des relations temporelles et spatiales dans un document multimédia interactif*. Thèse de doctorat en informatique, Université des Sciences et de la Technologie Houari Boumediène, Faculté d'Electronique et d'Informatique, 2009.
 - [55] Bernhard MARSCHALL : Integration of digital video into distributed hypermedia systems, 1995.
 - [56] Alan MARDEN, Harriet NOCK, Adrian MACKENZIE, Adam LINDSAY, John COLEMAN et Greg KOCHANSKI : Ict tools for searching, annotation and analysis of audiovisual media 2 executive summary, 2006.
 - [57] Britta MEIXNER, Beate SIEGEL, Günther HÖLBLING, Franz LEHNER et Harald KOSCH : SIVA suite : authoring system and player for interactive non-linear videos. *In Proceedings of the international conference on Multimedia*, MM '10, pages 1563–1566, New York, NY, USA, 2010. ACM.
 - [58] Thomas MEYER-BOUDNIK et Wolfgang EFFELSBERG : Mheg explained. *IEEE MultiMedia*, 2:26–38, March 1995.
 - [59] Jan MIK, Bao LE et Duc JAN : An Export Architecture for a Multimedia Authoring Environment. *Proceedings of the 8th ACM symposium on Document engineering - DocEng '08*, pages 28–31, 2008.

-
- [60] Gregor MILLER, Sidney FELS, Abir Al HAJRI, Michael ILICH, Zoltan FOLEY-FISHER, Manuel FERNANDEZ et Daesik JANG : Mediadiver : Viewing and annotating multi-view video. In *Computer Human Interaction ; Interactivity Session*, May 2011.
 - [61] M.H. MORALES : Hypervideo as a tool for communicating mathematics. Mémoire de D.E.A., European Master in Multimedia and Audiovisual Business Administration, Haute Ecole ICHEC - ISC ST-Louis-ISFSC, 2001.
 - [62] Matthias MULLER-PROVE : *Vision and Reality of Hypertext and Graphical User Interfaces*. Thèse de doctorat, Fachbereich Informatik (CS Department), 2002.
 - [63] IEEE MULTIMEDIA : MPEG-7 : The generic multimedia content description standard, part 1. *IEEE MultiMedia*, 9(2):78–87, 2002.
 - [64] Hans Peter BrøNDMO et Glorianna DAVENPORT : Creating and viewing the elastic charles - a hypermedia journal. In *UK Hypertext*, pages 43–51, 1989.
 - [65] Theodore H. NELSON : *Computer Lib/Dream Machines*. Microsoft Press, Redmond, WA, USA, 1987.
 - [66] NIELSEN, JAKOB : *Hypertext and Hypermedia*. Academic Press, San Diego, CA, 1990.
 - [67] Dimitris PAPADIAS, Yannis THEODORIDIS et Emmanuel STEFANAKIS : Multi-dimensional range query processing with spatial relations, 1997.
 - [68] Roy PEA, Michael MILLS, Joseph ROSEN, Kenneth DAUBER, Wolfgang EFFELSBERG et Eric HOFFERT : The diver project : Interactive digital video repurposing. *IEEE MultiMedia*, 11:54–61, January 2004.
 - [69] Roger T. PÉDAUQUE : Document : forme, signe et médium, les re-formulations du numérique. Rapport technique, CNRS RTP-DOC, juillet 2003. Disponible à http://archivesic.ccsd.cnrs.fr/sic_00000511.html.
 - [70] Silvia PFEIFFER et Conrad PARKER : Accessibility for the html5 <video> element. In *W4A '09 : Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility (W4A)*, pages 98–100, New York, NY, USA, 2009. ACM.
 - [71] Silvia PFEIFFER, Conrad PARKER et Claudia SCHREMMER : Annodex : a simple architecture to enable hyperlinking, search & retrieval of time-continuous data on the web. In *MIR '03 : Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 87–93, New York, NY, USA, 2003. ACM.
 - [72] Yannick PRIÉ : *Représentation de documents audiovisuels en Strates-Interconnectées par les Annotations pour l'exploitation contextuelle*. Thèse de doctorat, Université de Lyon, 12 1999.

-
- [73] Yannick PRIÉ et Serge GARLATTI : Annotations et métadonnées dans le web sémantique, 2004. Revue I3 Information-Interaction - Intelligence, Numéro Hors-série Web sémantique, 2004, 24 pp.
 - [74] J. RHÉAUME : Hypermédias et stratégies pédagogiques. In *Actes des premières journées scientifiques Hypermédias et Apprentissages*, 1991.
 - [75] Bertrand RICHARD : Etat de l'art sur la lecture active de documents audiovisuels. Rapport technique RR-LIRIS-2007-017, LIRIS UMR 5205 CNRS, mai 2007.
 - [76] Bertrand RICHARD : *Lecture active de documents audiovisuels : de la gestion de connaissances personnelles au soutien de l'attention du lecteur actif*. Thèse de doctorat en informatique, INSA de Lyon, janvier 2010.
 - [77] Madjid SADALLAH, Olivier AUBERT et Yannick PRIÉ : Component-based hyper-video model : high-level operational specification of hypervideos. In *to appear in Proceedings of DocEng2011*, 2011.
 - [78] Madij SADALLAH, Olivier AUBERT et Yannick PRIÉ : Hypervideo and annotations on the web. In *to appear in Proceedings of Multimedia on the Web Workshop MMWeb2011*, 2011.
 - [79] Abdulmotaleb El SADDIK, Stephan FISCHER et Ralf STEINMETZ : Reusability and adaptability of interactive resources in web-based educational systems. *J. Educ. Resour. Comput.*, 1, March 2001.
 - [80] Imad SALEH, Abderrazak MKADMI et Everardo REYES : *L'hypermédia au service du travail collaboratif*, pages 61–92. Hermes & Lavoisier, Paris, 2005. nt2 hypermédia.
 - [81] Andrew SALWAY : *Video Annotation : the role of specialist text*. Thèse de doctorat, Dept. of Computing, University of Surrey, 1999.
 - [82] Nitin Nick SAWHNEY, David BALCOM et Ian E. SMITH : HyperCafe : Narrative and Aesthetic Properties of Hypervideo. In *UK Conference on Hypertext*, pages 1–10, Bethesda, Maryland, United States, 1996.
 - [83] Patrick SCHMITZ, Aaron PATTERSON et Debbie NEWMAN : XHTML+SMIL Profile. W3C note, W3C, 2002. <http://www.w3.org/TR/XHTMLplusSMIL/>.
 - [84] Patrick SCHMITZ, Jin YU et Peter SANTANGELI : Timed interactive multimedia extensions for HTML. W3C note, W3C, 1998. <http://www.w3.org/TR/NOTE-HTMLplusTIME>.
 - [85] Ronald SCHROETER : Vannotea - A Collaborative Video Indexing , Annotation and Discussion System For Broadband Networks. In *In Knowledge Markup and Semantic Annotation Workshop, K-CAP 2003*, 2003.

-
- [86] Kerry SHEPHARD : Questioning, promoting and evaluating the use of streaming video to support student learning. *British Journal of Educational Technology*, 34(3): 295–308, 2003.
 - [87] Frank SHIPMAN, Andreas GIRGENSOHN et Lynn WILCOX : Authoring, viewing, and generating hypervideo : An overview of hyper-hitchcock. *ACM Trans. Multimedia Comput. Commun. Appl.*, 5(2):1–19, 2008.
 - [88] Rahul SINGH, Zhao LI, Pilho KIM, Derik PACK et Ramesh JAIN : Event-based modeling and processing of digital media. In *Proceedings of the 1st international workshop on Computer vision meets databases*, CVDB ’04, pages 19–26, New York, NY, USA, 2004. ACM.
 - [89] Arnold W. M. SMEULDERS, Marcel WORRING, Simone SANTINI, Amarnath GUPTA et Ramesh JAIN : Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22:1349–1380, December 2000.
 - [90] Elmar STAHL, Carmen ZAHN et Matthias FINKE : How can we use hypervideo design projects to construct knowledge in university courses ? In *CSCL ’05 : Proceedings of th 2005 conference on Computer support for collaborative learning*, pages 641–646. International Society of the Learning Sciences, 2005.
 - [91] Michael K. STENZLER et Richard R. ECKERT : Interactive video. *SIGCHI Bull.*, 28:76–81, April 1996.
 - [92] M. TANI, K. YAMAASHI, K. TANIKOSHI, M. FUTAKAWA et S. TANIFUJI : Object-oriented video : interaction with real-world objects through live video. In *CHI ’92 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 593–598, New York, NY, USA, 1992. ACM.
 - [93] Claudio A.B. TIELLET, André Grahl PEREIRA, Eliseo Berni REATEGUI, José Valdeni LIMA et Teresa CHAMBEL : Hvet : a hypervideo environment to support veterinary surgery learning. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, HT ’10, pages 313–314, New York, USA, 2010. ACM.
 - [94] Raphaël TRONCY : *Formalisation des connaissances documentaires et des connaissances conceptuelles à l'aide d'ontologies : application à la description de documents audiovisuels*. Thèse de doctorat, Université Joseph-Fourier - Grenoble I, 03 2004.
 - [95] Ronen VAISENBERG, Arjun SATISH, Keith A. MOGENSEN, Ramesh JAIN et Sharad MEHROTRA : A new approach for adding browser functionality. In *Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, HT ’08, pages 247–248, New York, NY, USA, 2008. ACM.

-
- [96] J. R. van OSSENBRUGGEN : *Processing Structured Hypermedia - A Matter Of Style*. Thèse de doctorat, Vrije Universiteit Amsterdam, 2001.
 - [97] E. VENEAU : *Macro-segmentation multi-critère et classification de séquences par le contenu dynamique pour l'indexation vidéo*. Thèse de doctorat, Université de Rennes I, Mention Traitement du Signal, February 2002.
 - [98] John WATLINGTON : Video finger. Mémoire de D.E.A., Massachusetts Institute of Technology, Massachusetts, United States, 1989.
 - [99] J. WHITEHEAD : Uniform comparison of data models using containment modeling. In *Proceedings of the thirteenth ACM conference on Hypertext and hypermedia*, ACM Hypertext '02, pages 182–191, New York, NY, USA, June 2002. ACM.
 - [100] Yan XU : *Representation of Story Structures for Browsing Digital Video*. Thèse de doctorat, Dept. of Computing, University of Surrey, 2005.
 - [101] Hung YUNG CHEN : A hypervideo system generator. *Software Practice and Experience*, 27(11), 1997.
 - [102] C. ZAHN, S. SCHWAN et B. BARQUERO : Authoring hypervideos : Design for learning and learning by design", in : R : Bromme, & e. stahl, writing hypertext and learning : Conceptual and empirical approaches. *Oxford : Elsevier Science*, 2002.
 - [103] Gang ZHAI, Geoffrey C. FOX, Marlon PIERCE, Wenjun WU et Hasan BULUT : esports : Collaborative and synchronous video annotation system in grid computing environment. In *ISM '05 : Proceedings of the Seventh IEEE International Symposium on Multimedia*, pages 95–103, Washington, DC, USA, 2005. IEEE Computer Society.
 - [104] Peter ZOLLER : *HMT - Modeling Interactive and Adaptive Database-driven Hypermedia Applications*. Dissertation, Technische Universität München, München, 2001.

ملخص

مع بروز أشكال جديدة لعرض و تمثيل الوثائق السمعية البصرية و المُتمثلة في هيرفيديو - وهي تكنولوجيا جذابة ترکز على الوسائل الفائقة التي تتكون غالبيتها من فيديوهات- و التي أحدثت ممارسات جديدة بالنسبة لطريقة الولوج إلى الوثائق السمعية البصرية و تقاسم المعرف التي تحتوي عليها. يمكن هدف هذه التكنولوجيا الجديدة في إرساء ثقافة سمعية بصرية ترکز على قراءة وتجارب مستخدمين غير مسبوقة مع نماذج تمثيل جديدة و بدائل تفاعلية أكثر تطورا.

تعتبر المفاهيم و الممارسات المُتعارف عليها في التمثيل التقليدي للوسائل الفائقة و المُعتمدة لحد الآن عامة جداً، بحيث لا يمكنها تغطية كل ما يميز الفيديو الفائق و هذا ما يحول دون دراسته الجيدة وبالتالي انتشاره.

- *Comonent-based Hypervideo Model - CHM*

إن الحاجة لأساس نظري حفز علي إقتراح *Comonent-based Hypervideo Model - CHM* و هو نموذج مُعطيات يعترف بتفوق الهيرفيديو عن الوسائل الفائقة و متعددات الوسائل التقليدية اللذان يمتازان بالعمومية وقلة التعبير. فهو يتبع مبادئ النمذجة التي ترکز على المكونات و الوصف الذي يعتمد على الشروح . يُعد النموذج المقترن بمثابة قاعدة نظرية لتطبيق آلي يستهدف الفيديوهات على الويب، فهو يستخدم كإثبات للتطور الحاصل في هذا المجال كما يطمح لتقديم بعض الأجوبة من أجل التكفل و الاستغلال الأمثل للفيديو على الويب.

الكلمات المفتاحية: فيديو تفاعلي، هيرفيديو ، متعدد الوسائل، شروح، نموذج يرتكز على المكونات، الوقت و التزامن، مرجعية زمنية (إسناد زمني)، CHM، ويب.

Résumé

Avec l'émergence de nouvelles formes de présentation et de représentation des documents audiovisuels, l'hypervidéo - une technologie attractive axée sur l'hypermédia à forte composante vidéo - introduit de nouvelles pratiques d'accès et de partage de la connaissance des documents audiovisuels complexes. Son objectif est d'instaurer une culture audiovisuelle axée sur une lecture active et des expériences utilisateurs inouïes avec des paradigmes de présentation nouveaux et des alternatives d'interactions encore plus élaborées. Généralement définis par des représentations hypermédia conventionnelles, les concepts et pratiques utilisés jusque là sont trop généraux pour couvrir tout ce qui caractérise les hypervideos, ce qui les empêche d'être mieux étudiées et plus répandues. Le besoin d'un fondement théorique a motivé la proposition de CHM - *Component-based Hypervideo Model* -, un modèle de données qui considère les hypervideos au delà de ceux de l'hypermédia et du multimédia trop généraux et peu expressifs. Il suit les principes de la modélisation à base de composants et la description à base d'annotations. Le modèle proposé sert comme base théorique pour une implémentation qui cible la vidéo sur le Web, présentée comme preuve des principes avancés et ayant comme perspective d'avancer des éléments de réponse pour une meilleure prise en charge et une exploitation optimale de la vidéo sur le Web.

Mots-clés: Vidéo Interactive, Hypervidéo, Multimédia, Annotations, Modèle à base de composants, Temps et Synchronisation, Référence de temps, CHM, Web

Abstract

Defined as a specialization of hypermedia focusing on interactive video, hypervideo is an attractive technology that promises non-conventional and powerful features. It aims to bring more powerful capabilities for video-centric documents and to offer novel user experiences with more elaborated practices and advanced interaction alternatives.

Usually defined from a conventional hypermedia perspective, the underlying concepts and representations are too general to qualify hypervideos, preventing this kind of documents from being broadly investigated. The need for theoretical foundations for hypervideos motivates the present work. With the aim to provide a formal data model that considers hypervideo beyond the very general and conventional hypermedia ones, CHM - Component-based Hypervideo Model - is an attempt to theorize hypervideos by following the general principles of component-based modeling and annotation-driven document description.

The proposed model serves as a basis for a Web-oriented implementation used as a proof of principles and a means to provide a basis for more elaborated practices of online video implementation since the features of hypervideo are grafted to wider video-based Web documents in a simple yet powerful and Web standards-compliant manner.

Keywords: Interactive Video, Hypervideo, Multimedia, Annotation-driven model, Time and Synchronization, component-based modeling, Timeline Reference, CHM, Web