

# Vers une Expérience Audiovisuelle Interactive sur le Web : Un Modèle à Base de Composants et un Langage de Spécification Déclaratif pour l'Hypervidéo

SADALLAH Madjid<sup>1,3</sup> – MAREDJ Azze-Eddine<sup>1,4</sup> – AUBERT Olivier<sup>2,5</sup> – PRIÉ Yannick<sup>2,6</sup>

<sup>1</sup>CERIST, 3 rue des frères Aissou, Ben-Aknoun, Alger, Algérie

<sup>2</sup>Université de Lyon, CNRS Université Lyon 1, LIRIS, UMR5205, F-69622, France

<sup>3</sup>[msadallah@cerist.dz](mailto:msadallah@cerist.dz), <sup>4</sup>[amaredj@cerist.dz](mailto:amaredj@cerist.dz), <sup>5</sup>[olivier.aubert@liris.cnrs.fr](mailto:olivier.aubert@liris.cnrs.fr), <sup>6</sup>[yannick.prie@liris.cnrs.fr](mailto:yannick.prie@liris.cnrs.fr)

---

**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 formes de diffusion de documents audiovisuels à structures complexes. Son objectif est de permettre des expériences utilisateurs inouïes avec des paradigmes de présentation nouveaux et des alternatives d'interactions encore plus élaborées. Généralement considérées via des modèles hypermédiés, les concepts et les représentations associés aux hypervidéos restent encore trop généraux pour couvrir toutes leurs caractéristiques, les empêchant d'être mieux étudiées et plus répandues. Le besoin d'un fondement théorique spécifique aux hypervidéos a motivé la proposition du modèle CHM - Component-based Hypervideo Model -, élaboré suivant les principes de la modélisation à base de composants et la description à base d'annotations. Cette nouvelle base théorique sert comme fondement pour une implémentation des hypervidéos ayant pour cible la vidéo sur le Web, présentée comme preuve des principes avancés et ayant en perspective de proposer une meilleure prise en charge et une exploitation optimale de la vidéo sur le 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 considered 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 using a component-based modeling and an 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.

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

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

---

## 1. Introduction

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 *Youtube*, *Vimeo*, *Dailymotion* 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 numérique, dont le contenu ne cesse d'augmenter en quantité mais aussi en qualité, soit l'une des sources d'information les plus sollicitées. La vidéo interactive est une forme récente et très attractive de la vidéo numérique qui présente des caractéristiques *informationnelles* et *interactionnelles* nouvelles. De nouvelles formes d'expérience utilisateur émergent des possibilités d'influencer l'expérience audiovisuelle à priori *filmique*, s'étendant au delà de celles répandues actuellement vers des utilisations plus avancées comme l'apprentissage et la publicité interactive.

L'intégration de la vidéo interactive dans des espaces hypermédiés hétérogènes tels que le Web soulève de nouveaux besoins issus des spécificités des vidéos interactives quant à la

manière de créer, représenter, échanger et présenter l'information contenue. En effet, les approches usuelles pour l'exploitation et l'interaction avec la vidéo dans les hypermédias ne sont pas encore à même de capitaliser le déluge de la matière audiovisuelle brute constamment produite [1]. De ce fait, les expériences acquises séparément avec les hypermédias et l'audiovisuel doivent être combinées à même de mieux prendre en charge intégrer la vidéo interactive dans les systèmes hypermédias. Cette combinaison est le but principal de la technologie *hypervidéo* que nous abordons dans cet article. 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.

Bien que certains concepts propres aux hypervidéos soient discutés depuis la fin des années 1990s [2], leur application effective n'est pas encore largement répandue. 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 prévenu les hypervidéos d'être examiné et utilisé à une plus grande échelle. En effet, 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 ce constat, nous proposons d'aborder la définition formelle de l'hypervidéo via la proposition d'un modèle hypervidéo, confortée par une implémentation sur l'environnement hypermédia le plus répandu : le Web.

La suite de cet article commence par la revue de l'état de l'art relatif aux hypervidéos de points de vue : concepts, modèles, systèmes et déploiement sur le Web. La discussion de l'existant permet la proposition, dans la section 3, du modèle CHM pour la représentation formelle des hypervidéos. Ce modèle servira pour le développement d'un framework open-source appelé WebCHM, présenté à la section 4. La dernière section conclut l'article.

## 2. Les hypervidéos

### 2.1 Définition et caractéristiques

Le terme hypervidéo a été initialement introduit avec le système *HyperCafe* [2] pour dénoter une nouvelle façon de structurer et de présenter dynamiquement une combinaison entre des flux vidéo et d'autres données. Aujourd'hui, l'hypervidéo désigne un type particulier de documents hypermédias qui combine la richesse de l'audiovisuel avec les capacités du multimédia et l'interactivité de l'hypermédia. Dans de cet article, le concept d'hypervidéo est défini comme tout document hypermédia centré sur un contenu audiovisuel interactif. Ainsi, ayant la vidéo comme centre de gravité sémantique, spatial et temporel, notre définition de l'hypervidéo introduit 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 hypermédias.

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 [3]. Par conséquent, l'hypervidéo peut être représentée par un graphe dont certains de ses nœuds sont des fragments audiovisuels [4].

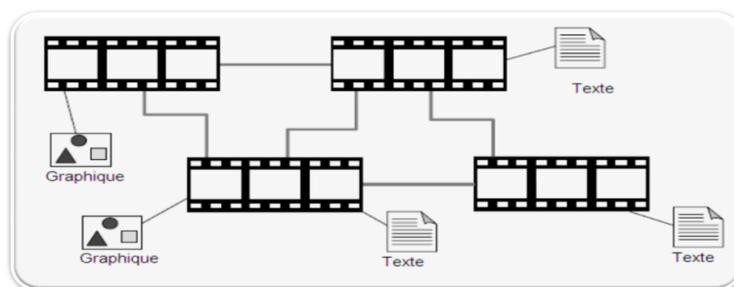


Figure 1. Structure de l'hypervidéo

La structure hypervidéo est constituée de deux composantes complémentaires : le flux audiovisuel continu et les données associées (voir figure 1). Ces dernières enrichissent le flux et permettent d'interagir avec lui et de le contrôler [4], 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 [5]. 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 [4].

## 2.2 Modélisation des hypervidéos

Un modèle de document fournit une description de l'organisation des données contenues dans un document [6] 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 [7]. Un tel modèle spécifiquement dédié aux hypervidéos n'existe pas. De ce fait, les hypervidéos sont souvent décrites par des modèles hypermédia très généraux tels que ceux de *Dexter* [8] et d'*Amsterdam* [9]. Ces modèles peuvent effectivement théoriser d'une manière très générale les hypervidéos considérées comme documents hypermédia. Développés pour couvrir tous les concepts pertinents des systèmes de l'hypertexte et de l'hypermédia, il n'y a pratiquement aucune méthodologie de modélisation qui répond entièrement à leurs spécifications [10].

## 2.3 Environnements d'édition et de présentation hypervidéos

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» [12]. Néanmoins, la technologie de l'hypermédia à base de la vidéo a mis du temps pour porter ces idées à pleine réalisation [9, 10]. 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 HyperCafé [2]. Plusieurs systèmes hypervidéos ont été développés depuis, on peut citer : Hyper-Film [13], Hyper-Hitchcock [14], HyperSoap [15], VisualSHOCK MOVIE [16], Hvet [17], Advene [18] et SIVA Suite [19].

## 2.4 Hypervidéos sur le Web

### 2.4.1 Support multimédia sur le Web

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, comme Adobe Flash et Apple Quicktime, une plus grande intégration du multimédia est possible. Cette technologie a néanmoins plusieurs insuffisances et failles [20]. Par exemple, les plugins renvoient en général un contenu opaque incrusté sur la page avec des possibilités d'interaction très limitées. Leur format 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, leur utilisation dans une page ajoute une charge supplémentaire à la CPU, ralentissant les navigateurs et accentuant les vulnérabilités sécuritaires. C'est dans ce contexte que le nouveau standard HTML5 (HyperText Markup Language Version 5) [21] introduit de nouvelles balises et des spécifications DOM pour intégrer un contenu multimédia en natif, sans faire recours à des applications tierces. Pour cela, il propose que les pages Web soient traitées comme des conteneurs atemporels d'éléments continus <video> et <audio>.

#### 2.4.2 Problématique du temps sur le Web

La définition d'un comportement temporel aux documents structurés permet de leur associer une sémantique temporelle [22]. Or, la dimension temporelle est absente du Web exprimé par le standard HTML. Par conséquent, le contenu multimédia présente une opportunité et un défi à la communauté Web : des objets massifs de contenu opaque contraint 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.

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 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.

Le langage SMIL [23, 24] a été dès sa première version proposé comme standard d'expression multimédia sur le Web. XHTML+SMIL [25], 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.) et 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 [26]. 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.

Le standard *SMIL Timesheets 1.0* [27] est introduit par la spécification SMIL 3.0 pour la définition de *feuilles de temps* (Timesheets), par analogie aux feuilles de style CSS. Cette spécification définit un langage permettant de rendre disponible les éléments et attributs des modules *Temps et Synchronisation* de SMIL à une plus grande panoplie de langages à base de XML, y compris les langages absolument atemporels comme HTML.

#### 2.4.3 Quelques implémentations des hypervidéos sur le Web

Des sites tels que Youtube et DailyMotion ont gagné une audience incontestable grâce à leurs efforts pour vulgariser la publication de l'information audiovisuelle en montrant comment la vidéo peut effectivement servir pour le partage de la connaissance sur Internet [28]. Avec ces services, il est par exemple possible de poster un commentaire sur un contenu vidéo, ajouter une évaluation et partager un certain contenu par l'intégration d'un code HTML qui inclut le flux audiovisuel. Des formes primitives de l'hypervidéo sur le Web ont été développées dans plusieurs projets tels que *Adivi* [29]. Le projet *Popcorn.js* [30] s'intéresse à l'ajout de

métadonnées à la vidéo HTML5, permettant 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.

## 2.5 Discussion

L'approche de systèmes hypervidéos actuels est axée sur le développement d'outils pour des besoins spécifiques mettant en jeu des documents audiovisuels munis des capacités d'interaction. La plupart des représentations définies par ces systèmes s'appuient sur des aspects pratiques, exprimées en termes de besoins 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 des implémentations. De tels environnements sont développés sans discuter des bases théoriques sur lesquels les documents sont produits et d'un cadre formel propre. De ce fait, ils présentent une grande hétérogénéité dans la façon de définir les hypervidéos, résultant en un manque de conformité des réponses qu'ils réservent aux besoins et aux exigences des hypervidéos. Or, ce volet préalable de modélisation formelle doit être considéré dans le but d'avoir des références théoriques claires, solides et bien définies.

Un certain nombre de ces systèmes décrivent les hypervidéos via des modèles hypermédias généraux, principalement Dexter et Amsterdam. Or, avec ces modèles, certaines particularités clés des hypervidéos ne peuvent clairement prises en charge. Par exemple, 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) [11], empêchant l'ordonnancement des objets qui les composent, caractéristique prédominante dans les hypervidéo. D'autre part, le comportement dynamique des documents est spécifié uniquement par des liens [11] sans considérer d'autres mécanismes de changement dynamique tels que les événements. A ce problème d'applicabilité s'ajoute le 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.

En plus de la problématique de définition formelle, l'hypervidéo n'est pas encore largement expérimentée sur le Web à cause principalement du caractère atemporel de ce dernier qui entrave le développement à grande échelle d'applications multimédia en. Les résultats des efforts de standardisation dans ce domaine n'ont pas eu un réel succès jusqu'à présent puisqu'aucun outil de production multimédia en ligne diffusé à grande échelle n'utilise leurs propositions. Avec les techniques actuelles, l'hypervidéo - à l'image des autres applications multimédia - ne peut donc être pleinement et correctement exprimée sur le Web qui, comme souligné dans [31], 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* ». Par conséquent, la vidéo sur le Web n'est pas facilement *recherchable*, navigable, recomposable et mise en cache. Beaucoup d'autres fonctionnalités restent aussi à développer [32]. Par exemple, afin de permettre la navigation intra-objet et l'adressage de portions spatio-temporelles tels qu'expérimentés aujourd'hui avec les hypervidéos, des mécanismes de fragmentation de la vidéo doivent être mis en œuvre.

### 3. Modèle CHM pour les documents hypervidéos

Afin de répondre au besoin de modélisation des hypervidéos 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, nous proposons le modèle CHM – *Component-based Hypervideo Model* – des documents hypervidéos [33] [34].

#### 3.1. Description générale

L'hypervidéo est définie dans CHM comme étant une spécialisation de l'hypermédia centrée sur un ou plusieurs documents audiovisuels (généralement une vidéo ou un corpus de vidéos) annotés. La proposition du modèle s'articule sur un ensemble de principes :

- L'ajout d'une structure d'annotations à la vidéo pour pouvoir la délinéariser et de lui définir des comportements interactifs. D'autre part, différents rendus multimédias peuvent lui être associés comportant divers artefacts de visualisations comme les sous-titres, les illustrations et les enrichissements. De ce point de vue, l'hypervidéo résulte de l'adoption des méthodes de l'hypermédia à la vidéo numérique annotée [35].
- Un contenu multimédia associé à la vidéo grâce aux annotations permet d'enrichir et d'explique la partie audiovisuelle principale autour de laquelle la présentation est organisée en espace et en temps.
- La modélisation logique du document se base sur le concept de composant. Une telle approche est utilisée dans plusieurs modèles tels que ceux de Dexter et d'Amsterdam.
- Les paradigmes de l'hypermédia sont adaptés et parfois redéfinis, permettant des définitions formelles concises des paradigmes rencontrés dans les hypervidéos.
- Etant forcément multimédia, l'hypervidéo est décrite suivant les différentes dimensions du multimédia. La gestion dynamique est réalisée via une approche événementielle.

#### 3.2. Modèle d'Annotation

Le modèle d'annotation dans CHM tient ses origines du modèle *Cinelab* [36]. L'annotation dans CHM est définie comme étant une donnée associée à un segment logique de la vidéo appelé *fragment* [37] 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. 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*.

Une hypervidéo CHM est le résultat de la présentation d'un contenu audiovisuel auquel 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.

#### 3.3. Modèle logique à base de composants

Le modèle logique dans CHM introduit un formalisme à base de composants définissant le document suivant plusieurs niveaux hiérarchiques, comme illustré sur la figure 2. Pour 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 des composants.

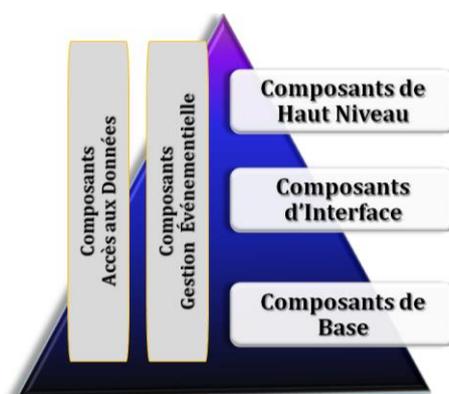


Figure 2. Modèle logique hiérarchique de CHM

### 3.3.1. Composants de base

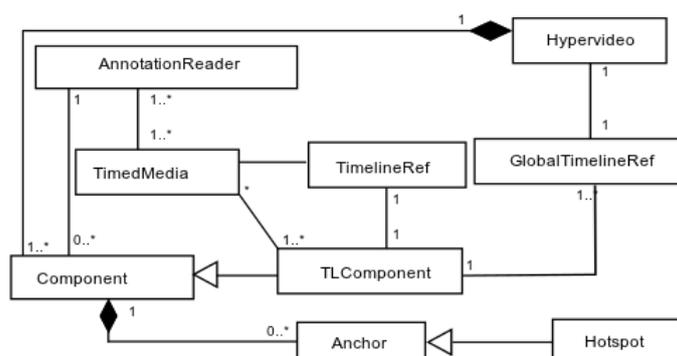


Figure 3. Les composants de base de CHM

Une hypervidéo dont l'élément racine est *Hypervideo* (figure 3) est formée d'un ensemble de composants temporalisés et non temporalisés, présentés lors de son exécution sur une interface physique de rendu. Elle fait référence à au moins un lecteur d'annotations, exprimé par l'élément *AnnotationReader* et qui permet d'accéder à une structure d'annotations et de renvoyer les données en réponse à une requête spécifique.

*Composants de rendu atomiques*: représenté par 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. Les composants avec des manifestations visuelles explicites et dont le message principal est véhiculé par son apparence sur l'interface utilisateur sont représentés via les éléments *VisualComponent*.

*Composants Temporalisés et de Synchronisation*: 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 temporalisé (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*) pour la synchronisation des objets de rendu reliés à l'élément *TimedMedia* en cours de présentation. Plusieurs lecteurs, et du fait, plusieurs TLRs peuvent coexister, définissant des sous-documents hypervidéos différents.

Les composants reliés à des TLR sont des éléments spécialisés appelés *TLComponent* contraints temporellement. Ceux indépendants du temps peuvent être utilisés comme conteneurs ou pour des besoins d'illustration statique. 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*.

### 3.3.2. Composants d'interface simple

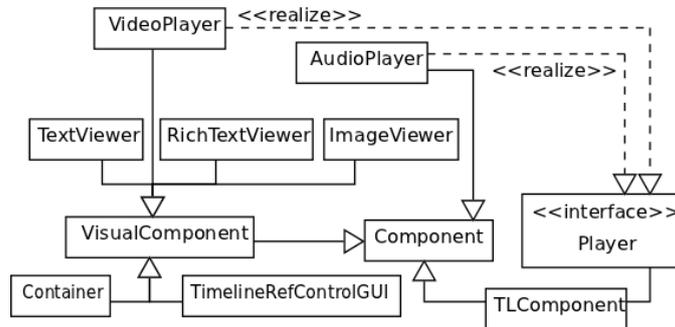


Figure 4. Composants d'interface simple de CHM

Les composants d'interface (ou composants simples - *plain components*-), présentés sur la figure 4, offrent un cadre abstrait 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.

*Conteneurs statiques*: Un conteneur statique *Container* est un composant offrant une interface graphique pour simplifier le regroupement spatial et/ou sémantique de plusieurs composants et l'unification des traitements sur les objets contenus.

*Interfaces de rendu des flux audiovisuels*: Les éléments *Player* sont des composants génériques définis dans le but de présenter un flux de données continu. Pour représenter un média, un fragment ou un montage, le lecteur (*player*) fait le rendu du média avec un ensemble de contrôles. Plus spécialisés, les éléments *AudioPlayer* et *VideoPlayer* définissent des interfaces adaptées aux médias audibles uniquement et audiovisuels respectivement.

*Interfaces de rendu des médias statiques*: Un lecteur de texte est représenté par l'élément *TextViewer*. La visionneuse représentée par l'élément *ImageViewer* est spécialisée dans le rendu des graphiques. Un contenu textuel plus complexe peut être présenté par le composant *RichTextViewer* qui prend en charge un certain nombre de types de document orienté XML tels que les documents HTML, RSS, XML+CSS, XSL, etc.

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

Les composants de haut niveau reprennent les artefacts les plus communs dans les hypervidéos. L'ensemble de ces composants est présenté à la figure 5. 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.

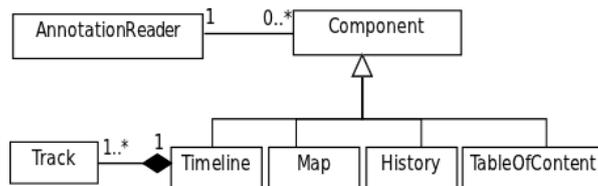


Figure 5. Composants CHM de haut niveau

*Historique*: Un historique de navigation permet de repartir vers un point du parcours déjà visité pour que l'utilisateur retrouve la configuration exacte de la présentation à l'instant correspondant. Lors d'une restitution, 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.

*Indexes de Navigation* : 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. Le plan ou cartographie d'un document hypervidéo, exprimé par un composant *Map*, est une représentation succincte ou un aperçu général du contenu permettant un branchement vers un point particulier de la progression narrative du document.

*Les lignes de Temps*: Une ligne de temps (composant *Timeline*) est une représentation graphique et interactive du temps de l'hypervidéo permettant de représenter graphiquement et d'organiser visuellement des caractéristiques particulières survenant au cours du temps. Elle place les éléments média, les événements importants ainsi que les liens sur un axe chronométré, sur différentes pistes ou *tracks* (composant *Track*) qui sont des représentations temporelles des périodes actives des annotations correspondant.

### 3.3.4. Composants d'accès aux données

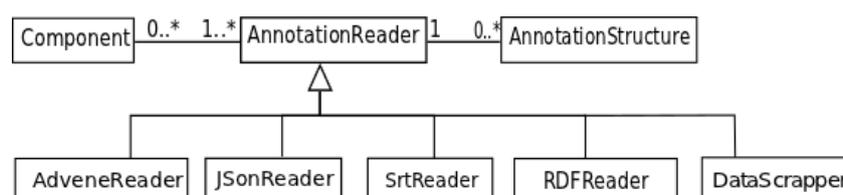


Figure 6. Composants CHM d'accès aux données

Au niveau composant, certaines données peuvent être internes à l'objet ou 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 6.

L'élément *AnnotationReader* représente le gestionnaire d'accès aux données le plus standard. Le modèle d'annotation que nous avons proposé peut être efficacement exprimé par le modèle d'Advene; 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 Advene. Le lecteur *JSonReader* se charge des structures d'annotation décrites sous le format JSON (JavaScript Object Notation). L'accès à certains formats de texte temporalisé peut utiliser le lecteur de données *TimedTextReader*. Le lecteur *RDFReader* peut être utilisé pour ajouter des capacités de requête sémantique exprimées en RDF alors que le lecteur *DataScrapper* permet de récupérer un contenu d'annotations à partir d'un sous-arbre DOM d'un document.

### 3.4. Modèle multimédia des hypervidéos CHM

#### 3.4.1. Dimension spatiale

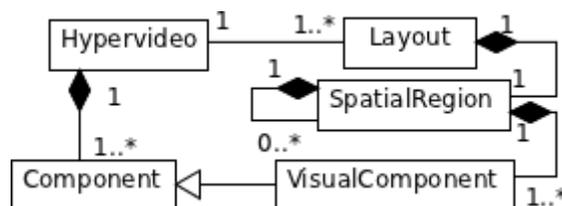


Figure 7. Modèle spatial des hypervidéos CHM

Le modèle spatial adopté dans CHM, présenté à la figure 7, représente chaque objet par son conteneur. Le conteneur parent de la présentation est défini par l'élément *Layout* qui maintient la configuration spatiale du document. Une *layout* est composée d'un ensemble de régions spatiales (*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 1 : Attributs spatiaux

Le modèle définit un ensemble d'attributs, présenté sur le tableau 1, permettant de spécifier les coordonnées et les dimensions des objets visuels.

#### 3.4.2. Dimension temporelle

*Paradigme TimeLine Reference (TLR)*: Une référence de temps, dénotée par TLR pour *TimeLine Reference*, est un référentiel temporel virtuel attaché à un composant audiovisuel principal afin d'ordonner 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.

La référence de temps présente une interface permettant de l'interroger et de la contrôler. Elle manipule trois attributs clés : *position* indique le niveau de progression de la présentation, *stat* indique si la présentation du document est temporellement en progression, mis à pause ou arrêté - et *duration*, attribut en lecture seule, renseigne sur la durée complète de l'hypervidéo.

*Attributs temporel et de synchronisation* : Les attributs présentés sur le tableau 2 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.

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 2 : Attributs temporels

La portée temporelle d'un média peut être 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 effective de lecture d'un composant équivaut à la durée intrinsèque de média contenu s'il est continu, multipliée par le nombre de boucles (*loopcount*), sans dépasser la durée d'affichage.

*Synchronisation multimédia* : La synchronisation des médias peut être fine (*hard*) ou pas (*soft*). La synchronisation *hard* force les média à maintenir un haut niveau de synchronisation avec la référence de temps. Une synchronisation *soft* permet au média de glisser temporellement. Ainsi, un composant peut être 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 sa durée de vie. L'attribut *CanSlideTLR* indique si le composant peut influencer la référence de temps.

### 3.4.3. Dimension hypermédia

Les fonctions de navigation sont spécifiées par un mécanisme de liens qui définit une connexion logique entre des composants de l'hyperespace. Les liens dans CHM sont *unidirectionnels*, *uni-sources* et *uni-destinations*, associés à des ancrs. Ils peuvent être de deux types: hypertextuels classiques et hypervidéos. Un lien hypervidéo a au moins une ancre définie sur une région spatio-temporelle de la vidéo appelée *hotspot*.

La cible d'un lien, et par extension, un lien peut être *interne* au document ou *externe* à ce dernier. Un lien interne a pour destination une ancre au sein du document, un instant de la référence de temps (dans ce cas la référence est la TLR) ou tout point de l'espace hypermédia. L'activation d'un tel lien provoque la mise à jour de la position de progression de la TLR. Un attribut de retour 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 replacer 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.

### 3.5. Modèle événementiel

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. La gestion événementielle est exprimée par les composants *Event* et *Action* (voir la figure 8). Les observateurs d'événement sont chargés de détecter les occurrences des événements et d'invoquer les procédures de traitement appropriées, exprimées en termes d'*actions*.

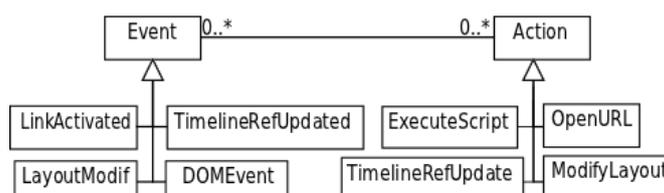


Figure 8. Les événements et les actions dans CHM

*Événements* : Les événements qui ne sont pas spécifiques aux hypervidéos sont modélisés par l'élément *DOMEvent* regroupent la plupart des événements génériques. Ceux affectant directement la progression temporelle de la ligne de temps sont représentés par les éléments *TimelineRefUpdated*. 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 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.

*Actions* : Un événement déclenche un ensemble de traitements. 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*. Les primitives permettant la navigation (accéder à une adresse URL), suite au déclenchement d'un certain événement peuvent être exprimées par le biais de l'élément *OpenURL*. L'utilisateur désirant définir ses propres actions ou étendre celles qui sont prévues dans le modèle peut utiliser l'élément *ExecuteScript* pour spécifier les actions désirées.

## 4. Framework WebCHM

Le framework WebCHM est développé afin d'illustrer une implémentation possible du modèle CHM pour le Web, ce dernier étant l'hypermédia le plus large, le mieux déployé et le plus utilisé.

### 4.1. 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 le succès et la vulgarisation de toute nouvelle technologie Web [14]. De ce fait, les choix technologiques retenus lors du développement de WebCHM sont opérés dans le but de permettre des mesures de qualité telles que l'accessibilité, la "recherchabilité", la structuration, la conformité aux standards et la réutilisabilité. Parmi ces choix :

- Un langage d'expression des hypervidéos par une syntaxe déclarative. Consistant en un ensemble d'éléments définis sur un espace de nom dédié<sup>1</sup>, il permet de décrire des documents hypervidéos riches et complexes.
- L'instrumentation du langage HTML : La spécification de la syntaxe CHM est implémentée comme une extension au-dessus du langage HTML.
- Des bibliothèques JavaScript pour interpréter la syntaxe CHM.

L'implémentation des concepts de CHM sera détaillée et illustrée progressivement à travers le scénario exemple ci-dessous.

<sup>1</sup> L'espace de nom CHM porte l'adresse `xmlns:chm="http://www.advene.org/ns/chm"`

*Scénario* : En 2009, au cours de la conférence *Technology Entertainment Design (TED)*, Tim Berners-Lee promut le projet *Linked data* décrivant une méthode pour exposer, partager et connecter les données via des URIs sur le Web. Nous voulons présenter une vue de la conférence d'une manière interactive, en se basant sur les annotations et en utilisant les technologies hypervidéos. Le scénario peut être résumé comme suit :

- 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 ;
- Pour chaque nouveau concept, sa définition est récupérée à partir de Wikipédia puis affichée sur un espace dédié ;
- A 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 ligne de temps place tous les événements suivant l'axe temporel.

## 4.2. Annotations

L'objectif du WebCHM n'étant pas de développer des outils d'annotation, les formats utilisés peuvent être obtenus en utilisant des outils dédiés tels que le prototype Advene et son module d'annotation et de génération de vues et d'exports. Les étapes du processus d'annotations peuvent être résumées en un ensemble de points :

1. analyse et annotation de la vidéo ;
2. export du contenu et des annotations vers les formats appropriés.

Pour notre implémentation, nous illustrons l'utilisation du format JSON (JavaScript Object Notation). La vidéo du scénario d'exemple a été analysée et annotée avec Advene et une partie du code de l'export de la structure d'annotation est représentée sur le listing suivant :

```

1 [...]
3 {
4   "id": "a1072",
5   "type": "Transcript",
6   "media": "TimBerners-Lee_2009.mp4",
7   "begin": 36223,
8   "end": 44701,
9   "content": "So going back to 1989,
10  I wrote a memo suggesting global hypertext system."
11 },
12 {
13   "id": "a1242",
14   "type": "Links",
15   "media": "TimBerners-Lee_2009.mp4",
16   "begin": 36223,
17   "end": 44701,
18   "content": "text=Tim Berners-Lee's
19 1989 memo\nurl=http://info.cern.ch/Proposal.html",
20   "parsed": {"_all":
21   "text=Tim Berners-Lee's 1989 memo\n
22   url=http://info.cern.ch/Proposal.html",
23   "text": "Tim Berners-Lee's 1989 memo",
24   "url": "http://info.cern.ch/Proposal.html"}
25 },
26
27 [.....]

```

## 4.3. Architecture du prototype WebCHM

Un document hypervidéo peut être rédigé par l'auteur avec la syntaxe CHM sous n'importe quel éditeur HTML. La génération de la présentation fait appel à un ensemble de bibliothèques quiinstancient les modules du moteur de rendu.

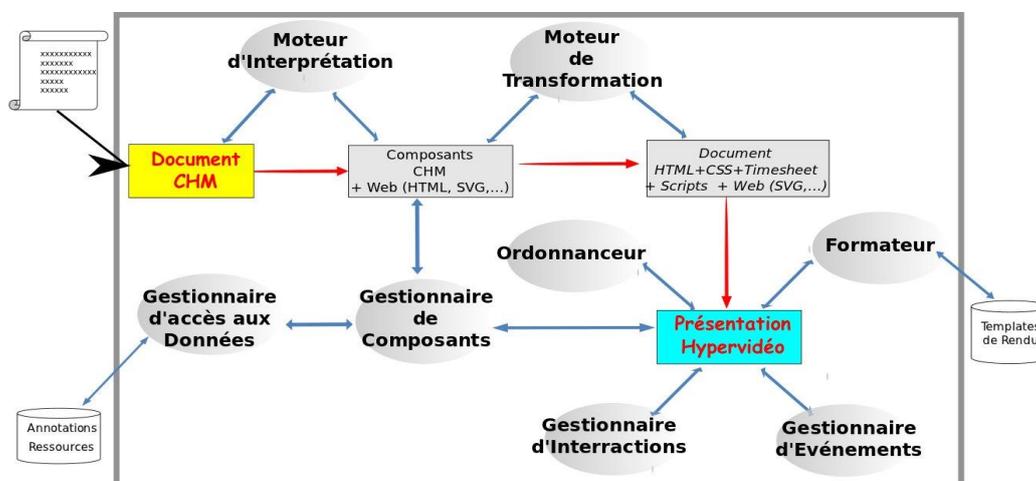


Figure 9. Processus général de création hypervidéo

La figure 9 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:
  - a. le contenu générique est exprimé en HTML simple auquel des classes, des scripts, du style, etc. peuvent être définis;
  - b. les éléments hypervidéos sont exprimés en syntaxe CHM appliquée à des éléments HTML;
  - c. le positionnement spatial et les dimensions de la layout, des régions spatiales et des éléments visuels sont décrits par des feuilles de style CSS;
3. Gestion de la présentation
  - a. la spécification temporelle et la synchronisation sont gérées par des scripts dédiés;
  - b. la gestion dynamique de la présentation est réalisée par l'ensemble des modules du moteur de rendu.

L'architecture générale du prototype *WebCHM* est à base de plusieurs modules, représentés sur la figure 10, mis en œuvre par des bibliothèques qui effectuent une analyse et une transformation de contenu pour générer dynamiquement un code HTML.

Figure 10. Architecture générale du prototype *WebCHM*

#### 4.4. Edition et spécification des 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.

*Éléments hypervidéos* : lors de l'édition d'un document, un élément HTML (typiquement une *DIV*) 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.

*Composants hypervidéos* : 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 *TLCComponent*).

*Lecteurs d'annotations* : les annotations sont stockées sous les formats appropriés et ont une structure générale qui inclut au moins les attributs: *id*, *type*, *begin*, *end*, *content* et *media*.

*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*. L'adressage de fragment est fait d'une manière augmentée suivant un sous-ensemble de la spécification *W3C Media Fragment* [38] pour pointer des fragments de média. 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";
```

#### 4.5. Interprétation, transformation et génération de code

##### 4.5.1. 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.

##### 4.5.2. Moteur de transformation :

Le moteur de transformation génère le code HTML des composants identifiés par le moteur d'interprétation. Pour chaque composant, ce processus peut être très complexe pouvant nécessiter plusieurs itérations. Par exemple, une table de matière doit d'abord être transformée en du code HTML contenant d'autres composants CHM tels que des composants *TextReaders* et *ImageViewers*. L'exécution de telles transformations peut générer des milliers de lignes, dépendant de la taille de données renvoyées par le lecteur d'annotations.

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.

##### 4.5.3. Mécanisme d'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. 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.

#### 4.5.5. Gestionnaire de composants

Les propriétés des composants CHM ainsi que leurs contenus sont définis par le *Gestionnaire de Composants* 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>`, `<IMG>`, 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>
```

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 moteur de transformation interroge le lecteur de données en lui fournissant un ensemble de paramètres. Cet appel est bloquant au niveau de la transformation. Une fois la réponse reçue, le transformateur effectue les opérations permettant de construire la table de matière à partir de ses composants atomiques. Un niveau donné de la table a la structure suivante:

```
tag="<tr><td><a href='${media}#t=${begin},${end}' target='timedmedia'>${content}</a></td></tr>";
```

## 4.6. Formatage spatial, temporel et synchronisation

### 4.6.1 Formatage spatial et mise en style

Nous utilisons 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.

#### 4.6.2. Gestion du temps et de la synchronisation

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* : à toute déclaration d'un *TimedMedia*, une TLR est définie. 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 l'ensemble des attributs temporels du modèle.

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 modifie les spécifications des composants *TLComponent* permettant de les synchroniser.

#### Implémentation de la dimension temporelle par les Timesheets

La dimension temporelle étant absente du Web, nous faisons appel à la spécification SMIL Timesheets pour réaliser le modèle temporel défini par CHM. En se basant sur la logique montrée dans le projet *Timesheet Scheduler* [39] 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.

#### 4.7. Comportement dynamique des hypervidéos

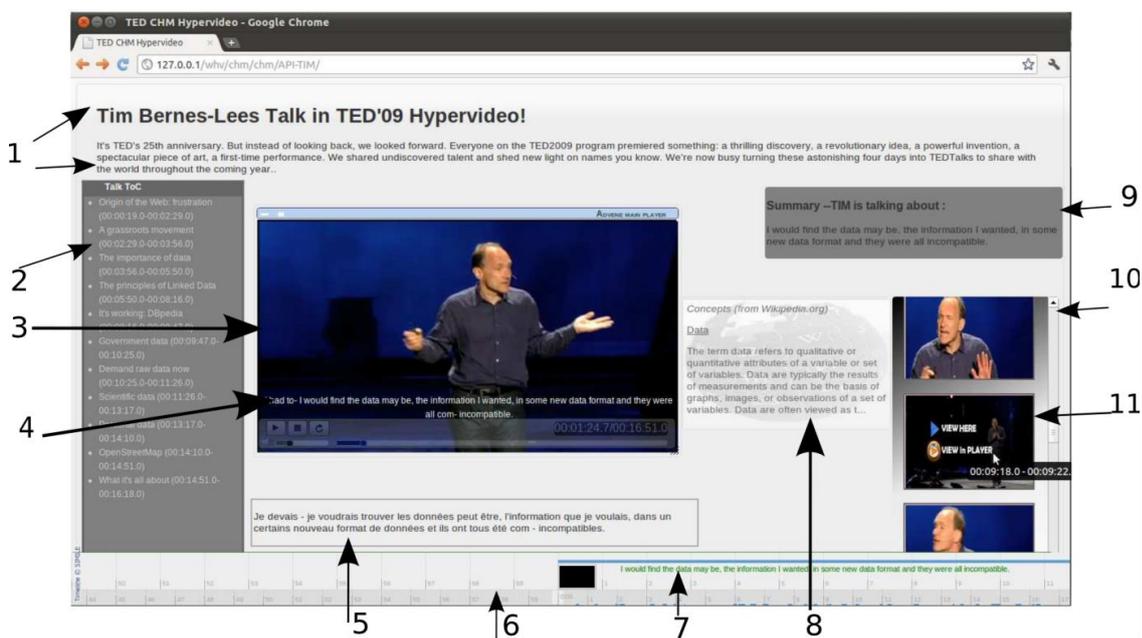
Le comportement dynamique de la présentation hypervidéo est induit principalement par sa progression dans le temps ainsi que par les événements de l'utilisateur lors des actions d'interaction et de navigation. Le gestionnaire d'interactions permet de gérer les opérations de manipulation de document et 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 ancrs et des liens, tels que le retour après navigation.

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. A la déclaration d'un événement, le gestionnaire événementiel lui définit un écouteur (*listener*) qui se charge de surveiller ses occurrences et d'invoquer la routine d'actions appropriée. L'implémentation de ce module est réalisée par l'utilisation du mécanisme événementiel du langage HTML via la manipulation par Javascript des événements de l'arbre DOM des composants.

### 4.8. Interface de présentation

Alors que *WebCHM* libère le concepteur d'hypervidéo de la rigidité de certains systèmes multimédia, nous avons fourni avec les bibliothèques un ensemble de choix par défaut lui permettant la production rapide d'hypervidéos. 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. Pour le rendu audiovisuel, *WebCHM* est mené d'un lecteur avancé qui exploite les nouveaux éléments HTML5 avec un ensemble de fonctionnalités permettant de proposer un lecteur hypervidéo avec plus de capacités que ceux existant, tout en permettant le passage à une implémentation Flash si les éléments HTML5 ne sont pas pris en charge par le navigateur.

La figure 11 montre le prototype *WebCHM* en action. Plusieurs composants sont illustrés sur ces deux figures, décrits sur le tableau 3.



(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 11. *WebCHM* en action

Elément	Composant CHM
01	Contenu HTML standard
02	Table de matière ( <i>TableOfContent</i> ) avec des 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.
03	Lecteur vidéo ( <i>VideoPlayer</i> ) principal, comportant une vidéo annotée (un composant <i>TimedMedia</i> )
04	Sous-titrage accompagnant la vidéo présentée ( <i>TextViewer</i> avec référence de temps - donc, un <i>TLCComponent</i> ) et synchronisé avec le flux.
05	Traduction en français ( <i>TextViewer</i> temporalisé) du discours affiché comme sous-titre synchronisé.
06	Ligne de temps ( <i>TimelineRefGUI</i> ) de la vidéo et donc de toute la présentation..
07	Pistes ( <i>Tracks</i> ) affichées tout au long de la ligne de temps et contenant un descriptif et une image - si disponible - du contenu d'annotation associé.
08	Données récupérées à partir de Wikipedia.org et présentées sous formes textuelles riches ( <i>RichTextViewer</i> temporalisé) synchronisée.
09	Résumé textuel enrichi et synchronisé ( <i>RichTextViewer</i> temporalisé) de la partie en cours.
10	Ensemble d'images clés (appelées vignettes) de début des parties du discours ( <i>Map</i> ).
11	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 ou de (pré-)visualiser uniquement le fragment correspondant sur une version simplifiée du lecteur vidéo et qui est présente au même endroit que la vignette (la remplaçant).
12	Région sensitive ( <i>hotspot</i> ) permettant d'aller vers le lien énoncé par l'orateur.
13	Version simplifiée du lecteur vidéo ( <i>player vignette</i> ).

Tableau 3 : éléments hypervidéos du scénario

#### 4.9 Discussion

Le framework WebCHM a été développé comme validation des principes énoncés par le modèle CHM. Suivant une approche déclarative, WebCHM permet de greffer les concepts spécifiques aux hypervidéos comme attributs au langage HTML pour 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. WebCHM consiste donc à prendre en charge les comportements induits par la déclaration d'un élément augmenté par un comportement hypervidéo et gérés par un ensemble de bibliothèques JavaScript génériques, réutilisables, extensibles tout en étant transparentes pour l'utilisateur (auteur et lecteur) et conformes aux normes et standards du Web. D'autre part, 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.

La rédaction d'un document hypervidéo est aussi facile que la création de n'importe quelle page Web. La syntaxe déclarative permet de décrire ce qui est attendu plutôt que de spécifier comment obtenir le comportement voulu. D'autre part, le rendu du contenu audiovisuel promeut l'utilisation de HTML5 comme base plutôt qu'adopter les solutions classiques comme tous les systèmes, à base de plugins.

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 métadonnées et les annotations permettent d'entrevoir d'autres manières de structurer et de présenter les contenus sur le Web.

## 5. Conclusion

Dans cet article, une théorisation des documents hypervidéos a été décrite via un modèle à base d'annotations et représentant 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 les relations 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. La nature basée sur les annotations du modèle CHM ajoute une nouvelle couche d'abstraction au contenu du document, le séparant de sa forme et son contexte de présentation.

La validation du modèle proposé est réalisée dans un environnement réel, établi, assez répandu et largement déployé : le Web. Les choix technologiques que nous avons opérés ont été motivés par le besoin d'être conformes au Web actuel et nous nous sommes focalisés sur une implémentation mettant en jeu les technologies Web les plus répandues et supportées : HTML, CSS et JavaScript. Écrire un hypervidéo complexe devient aussi simple que l'édition toute édition Web. 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. De nouvelles expériences utilisateur de la vidéo interactive sur le Web sont rendues possibles grâce à cette démarche consistant à greffer les capacités et les nouvelles idées véhiculées par les hypervidéos à un spectre de types de documents plus large.

## Références

- [1] Goldman, D. R. (2007). A Framework for Video and Annotation, Visualization and Interaction (Doctoral dissertation). University of Washington, Graduate School.
- [2] Sawhney, N. N., Balcom, D., & Smith, I. E. (1996). HyperCafe: Narrative and Aesthetic Properties of Hypervideo. In UK Conference on Hypertext (pp. 1–10). Bethesda, Maryland, United States.
- [3] Chen, H. Y. (1997). A hypervideo system generator. *Software Practice and Experience*, 27(11).
- [4] Aubert, O., & Prié, Y. (2004). From video information retrieval to hypervidéo management. In Corimedia, the international workshop on multidisciplinary image, video, and audio retrieval and mining, Sherbrooke, Canada.
- [5] Aubert, O., & Prié, Y. (2004). Documents audiovisuels instrumentés. temporalités et détemporalisations dans les hypervidéos. *Document Numérique*, 8(4), 143–168.
- [6] Whitehead, J. (2002). Uniform comparison of data models using containment modeling. In Proceedings of the thirteenth ACM conference on Hypertext and hypermedia, ACM Hypertext '02 (pp. 182–191). New York, NY, USA: ACM.
- [7] Richard, B. (2010). Lecture active de documents audiovisuels: de la gestion de connaissances personnelles au soutien de l'attention du lecteur actif (Doctoral dissertation). INSA de Lyon.
- [8] Halasz, F., & Schwartz, M. (1994). The dexter hypertext reference model. *Communications of the ACM*, 37(2), 30–39.
- [9] Hardman, L., Bulterman, D. C. A., & van Rossum, G. (1994). The Amsterdam hypermedia model: adding time and context to the dexter model. *Communications of the ACM*, 37(2), 50–62.

- [10] Zoller, P. (2001). HMT - Modeling Interactive and Adaptive Database-driven Hypermedia Applications (Doctoral dissertation). Technische Universität München.
- [11] van Ossenbruggen, J. R. (2001). Processing Structured Hypermedia - A Matter Of Style (Doctoral dissertation). Vrije Universiteit Amsterdam.
- [12] Nelson, T. H. (1987). *Computer Lib/Dream Machines*. Microsoft Press.
- [13] Pollone, M., Rusconi, M., & Tua, R. (2002). From hyper-film to hyper-web: The challenging continuation of a European project. In *Electronic Imaging & the Visual Arts Conference*, Florence, Italia.
- [14] Shipman, F., Girgensohn, A., & Wilcox, L. (2008). Authoring, viewing, and generating hypervideo: An overview of hyper-hitchcock. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 5(2), 1–19.
- [15] Bove, M., Dakss, J., Agamanolis, S., & Chalom, E. (2000). Hyperlinked television research at the MIT media laboratory. *IBM System Journal*, 39.
- [16] Mitsubishi America Inc. (2000). Authoring with hyperlinked video, whitepaper by Mitsubishi electric America Inc.
- [17] Tiellet, C. A. B., Pereira, A. G., Reategui, E. B., Lima, J. V., & Chambel, T. (2010). Hvet: a hypervideo environment to support veterinary surgery learning. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia, HT '10* (pp. 313–314). New York, NY, USA: ACM.
- [18] Aubert, O., & Prié, Y. (2004). Documents audiovisuels instrumentés. Temporalités et détemporalisations dans les hypervidéos. *Document Numérique*, 8(4), 143–168.
- [19] Meixner, B., Siegel, B., Hölbling, G., Lehner, F., & Kosch, H. (2010). SIVA suite: authoring system and player for interactive non-linear videos. In *Proceedings of the international conference on Multimedia, MM '10* (pp. 1563–1566). New York, USA.
- [20] Jansen, J., & Bulterman, D. C. (2009). Smil state: an architecture and implementation for adaptive time-based web applications. *Multimedia Tools and Applications*, 43, 203–224.
- [21] Hyatt, I. H. (s.d.). HTML5: A vocabulary and associated APIs for HTML and XHTML. <http://www.w3.org/TR/html5/>. Consulté le 08 avril 2012.
- [22] Jansen, J., Cesar, P., & Bulterman, D. C. A. (2010). A model for editing operations on active temporal multimedia documents. In *Proceedings of the 10th ACM symposium on Document engineering, DocEng '10* (pp. 87–96). New York, NY, USA: ACM.
- [23] Bulterman, D. C. A. (2008). Synchronized multimedia integration language (SMIL 3.0). W3C recommendation, W3C. <http://www.w3.org/TR/SMIL3/>.
- [24] Bulterman, D. C. A., & Rutledge, L. W. (2008). *SMIL 3.0: Flexible Multimedia for Web, Mobile Devices and Daisy Talking Books*. Springer Publishing Company, Incorporated.
- [25] Schmitz, P., Patterson, A., & Newman, D. (2002). XHTML+SMIL Profile. W3C note, W3C. <http://www.w3.org/TR/XHTMLplusSMIL/>.
- [26] Schmitz, P., Yu, J., & Santangeli, P. (1998). Timed interactive multimedia extensions for HTML. W3C note, W3C. <http://www.w3.org/TR/NOTE-HTMLplusTIME>.
- [27] Vuorimaa, P., Bulterman, D., & Cesar, P. (s.d.). SMIL Timesheets 1.0. W3C. <http://www.w3.org/TR/timesheets>. Consulté le 08 avril 2012.
- [28] Guimarães, R. L., Cesar, P., & Bulterman, D. C. A. (2010). 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: ACM.
- [29] InnoTeamS GmbH (2010). ADIVI - Hypervideo für Lernen und Support. <http://www.adivi.net/>. Consulté le 08 avril 2012.
- [30] Popcorn.js, The HTML5 Media Framework. <http://www.popcornjs.org/>. Consulté le 08 avril 2012.
- [31] Pfeiffer, S., & Parker, C. (2009). Accessibility for the html5 <video> element. In *W4A '09: Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility (W4A)* (pp. 98–100). New York, NY, USA: ACM.
- [32] Cesar, P., Bulterman, D. C. A., Jansen, J., Geerts, D., Knoche, H., & Seager, W. (2009). Fragment, tag, enrich, and send: Enhancing social sharing of video. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 5(3), 1–27.

- [33] Sadallah, M., Aubert, O., & Prié, Y. (2011). Component-based Hypervideo Model: high-level operational specification of hypervideos. In Proceedings of ACM Document Engineering DocEng2011 (pp. 53-56). September 2011, Mountain View, California.
- [34] Sadallah, M., Aubert, O., & Prié, Y. (2014). CHM: an annotation-and component-based hypervideo model for the Web. *Multimedia tools and applications*, 70, 869-903.
- [35] Finke, M. (2004). Mobile interactive video. Rapport technique, MUMMY Project.
- [36] Cinelab. <http://liris.cnrs.fr/advenc/cinelab.html> Consulté le 08 avril 2012.
- [37] Aubert, O., & Prié, Y. (2005). Advenc: active reading through hypervideo. In Proceedings of the sixteenth ACM conference on Hypertext and hypermedia, ACM Hypertext '05 (pp. 235–244). New York, NY, USA: ACM.
- [38] Troncy, R., & Mannens, E. (Eds.). Media Fragments URI 1.0. <http://www.w3.org/TR/media-frag>. Consulté le 08 avril 2012.
- [39] Cazenave, F., Quint, V., & Roisin, C. (2011). Timesheets.js: When SMIL meets HTML5 and CSS3. In Proceedings of the 11th ACM symposium on Document engineering, DocEng'11 (pp. 43-2). ACM, New York, NY, USA.