

# Instrumentation de documents audiovisuels : temporalisations et détemporalisation dans le projet Advene

Olivier Aubert

Pierre-Antoine Champin

Yannick Prié

LIRIS - FRE 2672 CNRS - Université Lyon 1  
prenom.nom@liris.univ-lyon1.fr

## 1 Introduction

La diffusion de l'audiovisuel (AV) numérique est en train de se développer, portée par les connexions internet haut-débit, la télévision numérique et le format DVD. Ceci est valable tant au niveau du grand public, essentiellement pour les loisirs (fictions, documentaires, etc.) que dans une utilisation plus professionnelle : vidéoconférence, surveillance, enseignement, recherche en Sciences de l'Homme et de la Société. Ces derniers domaines font notamment appel à la *lecture active* des documents audiovisuels, pratique dans laquelle il s'agit d'*annoter* les vidéos dans une perspective analytique : enregistrements sportifs, corpus linguistiques ou de situations d'interaction (dans une classe par exemple).

Le contexte de notre travail est celui de la lecture active de documents audiovisuels sur support DVD et du partage des analyses au sein de communautés : analyse filmique pour les cinéphiles ou les chercheurs, utilisation en classe pour enseignants et élèves (étude de vidéos, enseignement en classe de langue ou de littérature). Notre travail a donc pour objectif de prendre en compte des utilisations de vidéos qui soient différentes de la simple visualisation/consommation. Ces utilisations doivent se baser sur des interfaces de visualisation d'*hypervidéos*<sup>1</sup> dans lesquelles il s'agit d'interagir avec le flux audiovisuel de façon non triviale. Les utilisateurs doivent avoir la possibilité d'annoter les flux et de construire leurs propres hypervidéos.

Il est alors nécessaire de disposer de descriptions structurées de documents audiovisuels (mises en place au cours d'un processus d'annotation), et d'outils permettant leur édition, ainsi que leur lecture/visualisation. Pour cela, il faut construire des modèles documentaires permettant tout d'abord de localiser des fragments audiovisuels pertinents pour certaines analyses, ensuite de les qualifier (de leur attribuer une sémantique) et enfin de les mettre en œuvre dans des interfaces graphiques utilisant les structures documentaires en lien avec le flux. Le projet Advene (Annotations de DVds Echangées sur le NEt) propose un tel modèle, ainsi que des outils permettant la lecture active et le travail sur des documents audiovisuels au format DVD<sup>2</sup>.

Nous avons identifié plusieurs fonctionnalités que notre système doit offrir (et que notre modèle doit supporter) :

- *l'annotation de documents audiovisuels* comme édition d'annotations de type variés, liées au flux AV, formant une structure qui sera utilisée pour construire et interagir avec les hypervidéos. Cette structure obéira à un *schéma d'annotation* ;
- *la construction de vues* des documents audiovisuels et des annotations ;
- *la visualisation de documents hypervidéos* construits à partir du flux et des annotations en mobilisant des vues plus ou moins spécifiques ;
- *la gestion et le partage* des annotations, des schémas d'annotation, et des vues à l'aide de *recueils*.

Dans la suite de cet article, nous présenterons tout d'abord le modèle Advene, puis nous nous concentrerons sur les processus et les interfaces permettant l'annotation, la construction de vues et la visualisation des hy-

---

<sup>1</sup>Ce terme provient du projet Hypercafé [12], mais nous l'utiliserons dans un sens étendu : il dénotera aussi bien les vidéos avec des possibilités hypertextes que les documents hypertextes dans lesquels la vidéo joue un rôle non anecdotique.

<sup>2</sup>Nous pensons en effet que des usages nouveaux de l'audiovisuel ne peuvent réellement se développer que si une utilisation large d'outils se met en place. Pour cela, nous proposons d'une part de considérer des films DVD qui nous permettent de nous affranchir des problèmes de droits liés à la circulation des contenus audiovisuels, tout en utilisant des documents de haute qualité. A cet effet, seules les descriptions d'annotations et d'hypervidéos circuleront sur le réseau au sein des communautés. Nous visons d'autre part un format ouvert et des outils développés suivant les principes du logiciel libre.

pervidéos, considérées suivant l'angle des possibilités d'adaptation aux différentes temporalités des documents audiovisuels.

En effet, au moins trois temporalités peuvent être considérées.

La temporalité du flux tout d'abord : les documents audiovisuels numériques (non générés automatiquement) sont en effet construits à partir de la superposition de flux audio et vidéo synchronisés. Une succession suffisamment rapide des images du flux vidéo permet, grâce à la persistance rétinienne, de donner à l'individu soumis à ces images l'illusion de mouvement des formes qu'il observe. Si le flux vidéo correspond à ce que l'individu est habitué à percevoir et qu'il est synchronisé avec le flux audio, alors le phénomène dit d'*illusion de réalité* se produit. Il est donc impératif pour la réception pleine et entière d'un document audiovisuel que le « flux de conscience » de l'individu soit synchronisé avec le flux audiovisuel, c'est-à-dire que l'individu se « soumette » à la temporalité propre de ce dernier.

La temporalité du document ensuite : le document en tant qu'objet temporel possède une durée, un début, une fin. Il peut également être décrit par une structure documentaire qui correspond au genre du document, et qui prescrit l'enchaînement de séquences identifiables. Ainsi, un journal télévisé se déroule toujours de la même manière, une publicité se termine en général par le produit présenté, un film par un générique, etc.

La temporalité de la narration enfin (ou temporalité diégétique) : quand une histoire est racontée — que celle-ci ait réellement eu lieu ou soit imaginée — se construit le temps de l'histoire, qui doit être cohérent avec la temporalité de la vie réelle (on parlera d'heures, de dates, d'événements, etc).

Ces différentes temporalités peuvent être considérées à des degrés divers, et jouent diversement sur la manière dont il est possible (aussi bien physiquement que conceptuellement) d'annoter des documents audiovisuels, et de visualiser ces documents instrumentés par les structures d'annotations attachées. La deuxième partie de cet article proposera quelques réflexions à ce sujet, et présentera quelques-unes des possibilités offertes par les modèles et les outils du projet Advene.

## 2 Modèle Advene

L'objectif d'Advene est de fournir un cadre d'applications simple et extensible. Nous nous sommes plus particulièrement attachés à la structure *externe* des annotations, c'est à dire la manière dont elles sont contraintes, mises en relation les unes avec les autres, et présentées (ou rendues) dans des hypervidéos. En effet, les annotations ne sont pas simplement des données attachées à des fragments quelconques du flux audiovisuel. Elles sont structurées selon des *schémas d'annotation*, qui dépendent de la tâche sous-tendant la lecture active. Un *recueil* Advene permet de regrouper et de partager un ensemble cohérent d'annotations, les schémas qui les structurent, ainsi que des *vues* permettant le rendu de ces annotations.

En revanche, le modèle Advene pose aussi peu de contraintes que possible sur la structure *interne* des annotations, c'est-à-dire quel type de données est utilisé pour annoter le flux audiovisuel (texte simple ou mis en forme, XML, son...). Il s'appuie sur des types de données définis par ailleurs, et sur un mécanisme de plugiciel pour leur visualisation et leur édition.

### 2.1 Vue d'ensemble

La notion centrale du modèle Advene est celle de *recueil*. Un recueil est un ensemble pertinent (du point de vue de son auteur) d'éléments Advene, qui peut être échangé. La figure 1 représente, dans le formalisme UML, les différentes classes d'éléments qui constituent un recueil (classe Package), et leurs associations. Les données annotant le flux audiovisuel sont portées par les Annotations et les Relations entre elles. Ces dernières ne sont pas équivalentes, et sont regroupées en *types* (respectivement AnnotationType ou RelationType) décrits dans un Schéma du même recueil. Les requêtes (Query) permettent de sélectionner un sous-ensemble d'éléments du recueil. Les vues (View) permettent quant à elles de produire des hypervidéos en combinant des fragments du flux annoté et les éléments du recueil.

**Éléments définis ou importés** Il est important de remarquer que ce diagramme est simplifié, les relations entre certains éléments pouvant revêtir deux formes : la définition et l'import. Chaque élément constituant un recueil peut être *défini* par ce recueil (c'est à dire décrit explicitement), ou *importé* depuis un autre recueil (c'est-à-dire décrit par une simple *référence* à sa définition dans cet autre recueil). De la même façon, un schéma peut définir ou importer les types d'annotation et de relation d'un autre schéma. En fournissant un

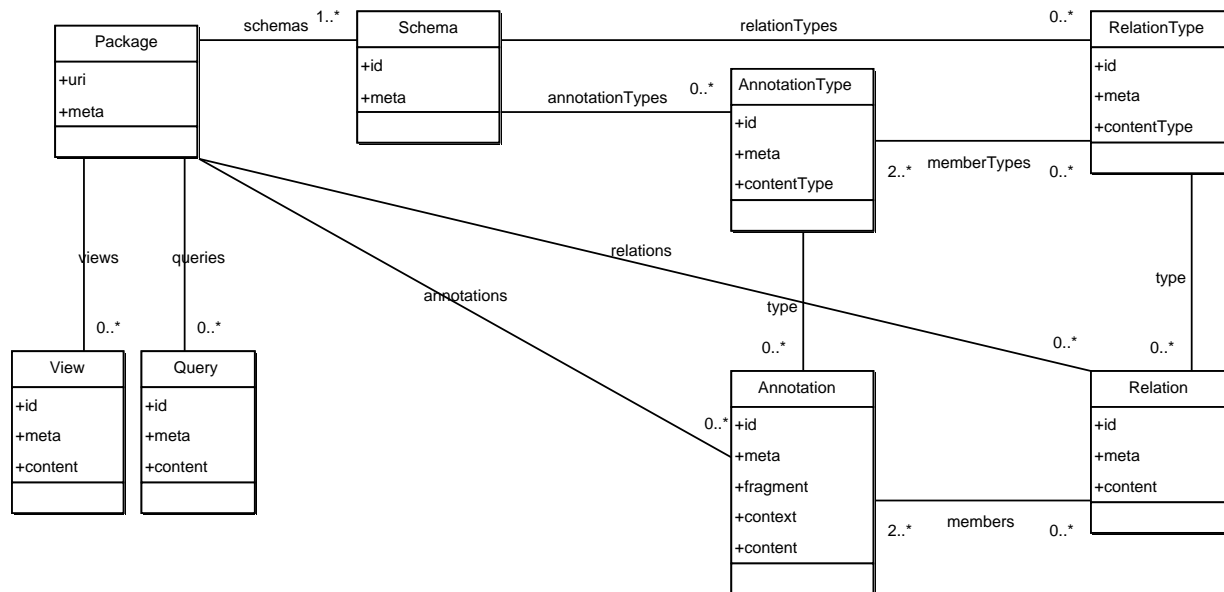


FIG. 1 – Diagramme UML du modèle Advene

accès uniforme aux éléments définis et importés, un recueil met à disposition toute l'information nécessaire<sup>3</sup> à la génération d'hypervidéos à partir des annotations.

## 2.2 Éléments du modèle

### 2.2.1 Points communs

Tous les éléments composant un recueil comportent un attribut `id` et un attribut `meta` (notons que le recueil lui-même comporte également l'attribut `meta`).

L'attribut `id` est un identificateur de l'élément, unique dans le recueil qui *définit* cet élément (par opposition aux recueils qui l'importent éventuellement).

L'attribut `meta` est une collection de méta-données. L'architecture Advene recommande fortement l'emploi (non exclusif) du schéma Dublin Core<sup>4</sup>. Par la suite, un certain nombre d'attributs du Dublin Core sera d'ailleurs recommandé pour chaque type d'élément.

### 2.2.2 Recueil

Un recueil permet de regrouper dans un unique document tous les éléments Advene (schémas, vues, annotations, etc.) produits ou utilisés par la lecture active d'un ou plusieurs DVDs, afin d'en faciliter l'utilisation et l'échange. Il est identifié par un URI. L'auteur et la date de création du recueil sont représentés comme méta-données à l'aide des éléments du Dublin Core.

Un recueil contient également une liste de tous les recueils dont il importe des éléments (cette liste n'est pas représentée dans le diagramme UML). Entre autres bonnes propriétés, cette liste permet d'établir immédiatement les dépendances entre recueils. Un autre rôle important de cette liste est qu'elle affecte un *identifiant d'espace de noms* à chaque recueil importé; ceci permet d'éviter les conflits de noms en préfixant l'identificateur de tout élément importé par celui de son espace de noms (ce mécanisme est très similaire à celui des espaces de noms XML [1]).

<sup>3</sup>Le flux audiovisuel est lui-même référencé par chaque annotation, quoiqu'il ne soit pas contenu dans le recueil.

<sup>4</sup><http://dublincore.org/>, Dublin Core Metadata Initiative

### 2.2.3 Annotations, Relations

Les annotations sont des données (appelées le *contenu* de l'annotation) attachées à des portions adressables du flux audiovisuel, que l'on appelle *fragments*. En plus des attributs `id` et `meta`, les annotations ont donc un attribut `fragment` et un attribut `content`.

Notons que l'identification d'un fragment dans un document audiovisuel numérique dépend de la structure spécifique de ce dernier. Dans une simple vidéo, seule une mesure temporelle (ou éventuellement en nombre d'octets) permet d'adresser un point du flux. Les DVDs, en revanche, sont structurés en titres et en chapitres. La norme MPEG-4 permet même d'adresser des objets (fragments spatio-temporels) dans le flux audiovisuel. Advène utilise pour l'instant des fragments définis comme intervalles entre deux bornes temporelles, mais permet l'extension à d'autres types de fragments.

Le support DVD offre de nombreuses options de visualisation : le même film peut en général être visionné dans différentes langues, avec ou sans sous-titres, parfois même selon différents angles de caméra. Ces paramètres définissent un *contexte* de visualisation. Une annotation posée dans un certain contexte ne sera pas forcément pertinente dans un autre contexte : par exemple, une annotation portant sur la qualité du doublage ne sera pas pertinente quand le film est visualisé avec la bande son originale. C'est pourquoi les annotations ont un attribut `context` permettant de représenter le contexte de visualisation dans lequel elles ont été posées. Comme pour les fragments, cette notion de contexte peut être étendue.

Par définition, les annotations sont en relation avec le flux audiovisuel, par le biais de leur attribut `fragment`. Il peut également être utile d'exprimer le fait que des annotations sont en relation les unes avec les autres. C'est le rôle de la classe `Relation` : une relation n'est pas reliée directement au flux, mais elle est liée à plusieurs annotations (les *membres* de la relation). Les relations peuvent également contenir des informations additionnelles par le biais de leur attribut `content`.

Un recueil peut définir ses propres annotations et relations, ou les importer depuis un autre recueil. Cependant, toutes les annotations et toutes les relations doivent correspondre à un schéma du recueil (voir ci-après).

### 2.2.4 Types

Toutes les Annotations et Relations ne sont pas équivalentes entre elles. Les types permettent de contraindre la structure d'annotations ou de relations jouant le même rôle.

Un type d'annotations (classe `AnnotationType`) définit le *type de contenu* que les annotations de ce type peuvent avoir. Dans le prototype Advène, le type de contenu est spécifié par un type MIME [3] (`text/html`, `audio/*`, etc.). Si le type de contenu est `text/xml`, il peut éventuellement être précisé par une DTD, un schéma XML ou Relax NG<sup>5</sup>.

Les types de relations (classe `RelationType`) définissent le nombre de membres, *i.e.* le nombre d'annotations qui doivent participer à une telle relation. Il définissent aussi le type d'annotation auquel chaque membre doit appartenir. Enfin, puisque les relations peuvent également avoir un contenu, les types de relation définissent un type de contenu, de manière similaire aux types d'annotations.

### 2.2.5 Schémas

Les schémas d'annotation (classe `Schema`) contraignent les annotations et les relations qu'un recueil peut contenir. Plus précisément, ils sont des collections de types d'annotations et de types de relations, regroupés sur la base des tâches de lecture active qu'ils permettent d'accomplir. Par exemple, les types d'annotations et de relations utiles pour l'édition d'un film (comprenant notamment les types d'annotations *Plan*, *Séquence*, et le type de relations *Transition*) seront regroupés dans un schéma *Édition* ; un autre schéma *Analyse technique* contiendra les types utiles à l'analyse des techniques cinématographiques (mouvements de caméra, lumière, ...); un troisième contiendra les types utiles à l'exploitation d'un film en cours de langue, etc.

Ce regroupement n'est bien sûr pas exclusif : un type utile à plusieurs tâches, défini dans un premier schéma, pourra ensuite être importé dans d'autres schémas. Par exemple, le type d'annotation *Plan*, appartenant au schéma *Édition*, sera réutilisé dans le schéma *Analyse technique* pour désigner les plans où ont lieu certains mouvements de caméra.

---

<sup>5</sup>En fait, ce principe peut être étendu à n'importe quel type MIME disposant d'un mécanisme similaire pour contraindre une classe de documents.

Un schéma et les types qui les composent possèdent un titre et une description. Ces méta-données ont pour rôle d'expliquer la façon d'utiliser les éléments.

### 2.2.6 Requêtes

Une requête (class Query) permet d'accéder à une liste d'éléments Advene (de n'importe quel type décrit dans cette section). Elle peut éventuellement être paramétrable. L'attribut `content` contient une description de la requête dans un langage spécifié par son type MIME. Le seul langage de requête implémenté pour le moment est le langage python, une implémentation immédiate puisque le prototype est lui même écrit en python. Des langages de requête plus simples d'utilisation devront être développés et implémentés à l'avenir.

Alors que les schémas offrent un premier moyen de structurer les annotations, les requêtes permettent quant à elles une structuration plus souple (en fonction du pouvoir d'expression du langage de requête fourni). Cette structuration sert ensuite de base aux hypervidéos générées grâce aux vues (cf. ci après). D'autre part, les requêtes permettent de structurer les annotations *a priori*, c'est à dire au moment de la création des schémas, mais elles peuvent aussi être créées *a posteriori*, pour structurer de manière totalement *ad hoc* un ensemble spécifique d'annotations. Dans le premier cas, elles peuvent, comme les schémas, être définies dans un recueil spécifique destiné être importé par toute personne souhaitant les utiliser dans son propre recueil.

### 2.2.7 Vues

Les vues (classe View) sont le dernier élément de la chaîne permettant de générer des hypervidéos à partir d'un recueil Advene. Les requêtes permettent de sélectionner un ensemble d'éléments d'un recueil. Les vues permettent quant à elles de *présenter* ces éléments de différentes manières, soit globalement statique, soit globalement dynamique (cf. ci-après). Conformément à la vocation de généralité d'Advene, une vue peut être décrite dans n'importe quel langage de présentation identifié par son type MIME. Dans le prototype actuel, les vues globalement dynamiques sont définies par le biais d'un langage de type événementiel (événement-condition-action) tandis que les vues globalement statiques sont définies par le langage TAL.

Le langage TAL a été développé pour la plateforme applicative Zope<sup>6</sup>. Le souci de modularité et de simplicité d'utilisation ayant présidé à son développement nous a fait adopter ce langage comme langage de description de vues pour Advene. En effet, sa syntaxe a facilement pu être adaptée à la structure spécifique des recueils Advene. Par ailleurs, TAL permet une intégration transparente avec des outils d'édition WYSIWYG. On peut donc envisager des éditeurs instrumentés permettant à un public novice de créer des vues simples, par réutilisation de vues existantes. Enfin, TAL sert principalement à générer des documents HTML (vues globalement statiques), mais peut générer n'importe quel type de document XML. Il permet donc de générer, par exemple, un document SVG ou SMIL[14], et permet donc en principe être utilisé pour décrire des vues globalement dynamiques<sup>7</sup>.

## 3 Visualisation, temporalité et détemporalisation

Nous proposons dans cette partie l'état actuel d'une réflexion en cours sur les différents niveaux de temporalités en jeu dans les systèmes d'information audiovisuelle, en essayant de prendre en compte les temporalités des documents audiovisuels, des utilisateurs, et bien entendu des constructions que sont les hypervidéos. Nous en profiterons pour présenter quelques vues d'ores et déjà implantées dans le prototype Advene.

### 3.1 Retour sur les temporalités du document original

Rappelons tout d'abord quelles étaient les trois temporalités liées aux documents audiovisuels que nous évoquons dans l'introduction : temporalités du flux d'abord, du document ensuite, diégétique enfin.

Il nous semble que les deux dernières diffèrent radicalement de la première. Elles peuvent en effet être considérées en première analyse comme ayant trait à la nature documentaire des documents audiovisuels comme manifestations humaines, et étudiées dans des cadres équivalents à ceux utilisés pour les documents textuels à « textualité séquentielle ». Ainsi, on pourra considérer une ou plusieurs structures documentaires pour un

---

<sup>6</sup><http://www.zope.org/Wikis/DevSite/Projects/ZPT/TAL>

<sup>7</sup>Ceci n'est pas totalement exploitable en pratique car le prototype Advene n'est pas capable d'interpréter le langage SMIL.

document audiovisuel, ainsi que de multiples structures d'analyse, qui s'appuieront toutes sur les annotations de fragments du flux, ainsi que les relations entre ces annotations. Cette structure globale d'annotation constitue la structure analytique d'enrichissement du flux.

La temporalité du flux, au contraire, ressort du caractère proprement physique de l'expérience audiovisuelle, et est représentée canoniquement en machine par la description du flux audiovisuel comme flux d'images et flux sonores superposés. La génération d'une animation multimédia entraînant une expérience audiovisuelle (par exemple un cube qui se déplace) ressort de la même temporalité en terme de rendu, tandis que le codage en machine diffère (puisqu'il s'agit alors de décrire ce qui se passe, et comment l'obtenir). Ajoutons que certaines annotations peuvent de droit avoir un contenu pouvant être considéré comme ayant une temporalité de flux (par exemple une annotation sonore).

Les temporalités du document audiovisuel annoté ressortent donc de deux catégories de temporalité, analytique d'une part, physique d'autre part, qui se caractérisent en machine par des structures et des codages différents.

### 3.2 Flux temporels pour l'analyse des hypervidéos

Considérons maintenant les hypervidéos qui sont des document multimédias générés à partir d'une ou plusieurs vues, une vue étant elle-même composée à partir des ces deux types de codage.

Les vues sont des hyperdocuments multimédias car elles peuvent être composées de textes, d'images, d'extraits AV plus ou moins longs, et être interactives (interaction au sein d'une vue, ou pour passer d'une vue à l'autre, auquel cas l'hypervidéo est composée de plusieurs vues). Notons que certains types de vues peuvent très bien ne pas faire appel au document audiovisuel annoté (le DVD dans Advène) pour être rendues.

Les hypervidéos construites à partir d'un flux annoté, ainsi que les temporalités induites sont potentiellement d'une très grande complexité. Ceci nous amène à proposer un cadre d'analyse dans lequel nous puissions à la fois considérer le point de vue de la machine sur les hypervidéos générées, et celui de l'utilisateur pour la réception de celles-ci.

Nous considérons ainsi deux flux temporels, qui sont le *flux du jeu* et le *flux de l'utilisateur*.

Le flux du jeu ressort du temps propre du document multimédia généré. De ce point de vue, on observe l'enchaînement des événements automatiques ou déclenchés par l'utilisateur. Le début et la fin de l'affichage d'éléments (fragment de vidéo, texte, image...) composant le document hypervidéo constituent par exemple des événements.

Certains éléments du flux du jeu seront des éléments audiovisuels (morceaux du document AV annoté, annotations AV, éléments dynamiques générés). On pourra alors si besoin y considérer des sous-éléments et événements (par exemple les images, et les événements d'affichage de nouvelles images). Cependant, le caractère particulier de l'expérience audiovisuelle nous incitera à les considérer globalement le plus souvent.

Certains éléments du flux du jeu seront des éléments plus « conceptuels », car ils auront été définis à partir des structures d'annotation et des définitions des vues (par exemple le contenu d'une annotation, ou bien une image correspondant à la fin d'un fragment).

Le flux de l'utilisateur ressort du temps vécu du document multimédia généré et est en relation avec le flux de conscience de l'utilisateur. On observe le document vécu par l'utilisateur de manière passive (en termes de soumission à ce qui se déroule automatiquement : AV, changement automatiques ; et aux possibilités d'actions) et de manière active (actions effectives de l'utilisateur).

Nous considérons en première analyse qu'il existe deux grands types de documents vécus, qui diffèrent suivant le degré d'intervention nécessaire de la part de l'utilisateur.

La première catégorie est celles des documents « globalement statiques », qui se base essentiellement sur une présentation hypertexte standard des annotations et du contenu AV du document annoté, avec des possibilités d'actions hypertextes. Alors la visualisation du document résultat dépend entièrement des actions de l'utilisateur, la temporalité du document vécu dépend essentiellement de lui. Nous avons également défini cette catégorie par le terme de UTBV pour *User-Time Based Visualisation* – visualisation basée sur le temps de l'utilisateur.

La deuxième catégorie est celle des documents « globalement dynamiques », dans laquelle la soumission de l'utilisateur à la temporalité induite par le flux du jeu est forte. Par exemple, un lecteur vidéo est instrumenté

pour présenter des annotations comme incrustation sur la vidéo. Nous avons également défini ce mode par le terme de STBV pour *Stream-Time Based Visualisation* – visualisation basée sur le temps d’un flux.

Ces deux grandes catégories peuvent être rapprochées de la distinction qui est faite dans [2] entre « pages centrées vidéo » et « pages centrées texte » et devront être raffinées. Par exemple, le degré de soumission (faible ou fort) au flux du jeu pourra dépendre de l’importance (de la taille) du lecteur vidéo « enrichi » (flux AV), ou de l’importance du flux événementiel qui doit être suivi (rapide, lent).

Pour résumer cette section : nous proposons pour l’analyse des hypervidéos <sup>8</sup> de distinguer deux flux temporels : le flux de l’utilisateur et le flux du jeu. Selon le point de vue considéré (utilisateur/programme), on pourra alors considérer le document hypervidéo résultant comme une « hypervidéo vécue » ou bien comme une « instance d’hypervidéo ».

Dans certains cas — les plus simples — ces flux se recouperont. Ce sera par exemple le cas des document statiques, ou bien de la simple lecture d’un document audiovisuel (lecteur standard). Dans d’autres cas, les deux catégories nous seront utiles, dès qu’il s’agira de combiner éléments statiques et dynamiques, et d’en considérer aussi bien la production que la réception.

### 3.3 Temporalités des documents annotés et hypervidéos

Après avoir dans les sections précédentes considéré d’une part les temporalités du document original annoté (physique/analytique), d’autre part les flux temporels nécessaires à l’analyse et à la conception d’hypervidéos (flux du jeu/flux de l’utilisateur), nous pouvons présenter quelques exemples d’hypervidéos mêlant ces diverses considérations. Ce sera l’occasion de présenter quelques-unes des vues déjà implémentées dans le prototype Advène, mais aussi de discuter les possibilités offertes par le modèle que nous proposons.

Ces exemples se basent sur deux schémas d’annotation :

- un schéma « Découpage en plans », permettant de poser des annotations de type Plan, dont le contenu est une brève description textuelle de ce plan, et des relations entre plans permettant d’indiquer leur ordre diégétique ;
- un schéma « Exploitation pédagogique », fournissant un type d’annotation Référence, dont le contenu est l’URL d’une ressource concernant la séquence annotée, et un type d’annotation Épisode découpant de film en unités pertinentes sur le plan pédagogique. D’autres types, non détaillés ici, permettent la mise en place d’exercices à propos du film.

Dans de nombreux cas, les annotations ont un contenu textuel plus ou moins structuré (texte simple, HTML, XML), à partir duquel il est possible de générer des documents *globalement statiques*. Ces documents peuvent également exploiter des images extraites à la volée du DVD pour illustrer ces documents. Par exemple, les annotations de type Plan peuvent être utilisées pour générer une liste chronologique (temps de la narration) des plans, avec leur première image et leur description. Une telle vue est une *spatialisation* du temps du film. Si on exploite des relations du même schéma, on peut aussi envisager d’ordonner cette liste suivant l’ordre diégétique. Un autre exemple de vue globalement statique, utilisant le schéma pédagogique, est la génération d’une feuille d’exercice et du corrigé à partir des annotations dédiées posées sur le film.

Dans le cas le plus simple de vue *globalement dynamique*, l’hypervidéo produite a une temporalité découlant directement de celle du flux annoté. Par exemple, le film annoté par ses plans peut être visualisé en affichant, en sous-titre, la description de chaque plan au début de celui-ci. Avec le schéma pédagogique, cette temporalité peut être plus ou moins altérée : les annotations de type Épisode peuvent entraîner un arrêt sur image à la fin de chaque épisode, réclamant un redémarrage manuel. Les annotations de type Référence peuvent quant à elles être présentées sous forme d’hyperliens, permettant de passer d’un instant du film à un autre, voir à un tout autre document.

Enfin, comme pour les vues globalement statiques, la structure des hypervidéos produites par les vues globalement dynamique peut être basée sur la structure induite par les relations entre annotations, « déstructurant » ainsi la temporalité initiale du flux annoté. Dans notre exemple de l’annotation en plans, les relations d’ordre diégétiques peuvent être utilisées pour générer à la volée un re-montage virtuel du film, oubliant l’ordre de la narration au profit de l’ordre des événements.

La figure 2 présente un exemple d’utilisation pédagogique d’Advène, faisant figurer trois modes de visualisation : d’une part, une visualisation globalement statique à l’intérieur d’un navigateur web, présentant une

---

<sup>8</sup>Mais également afin de disposer d’un cadre de conception.

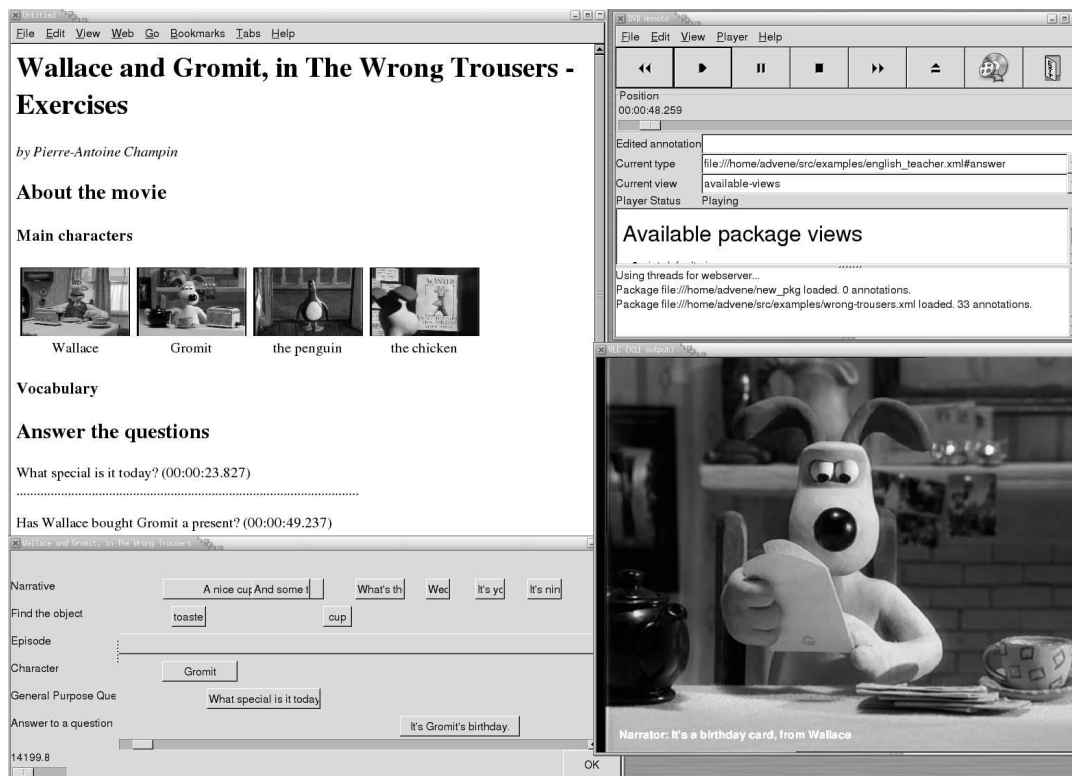


FIG. 2 – Utilisation pédagogique d'Advene

feuille d'exercice utilisant le contenu de certaines annotations (de type Question), enrichi de vignettes tirées du DVD. D'autre part, une autre visualisation globalement statique, utilisée principalement lors de l'édition d'annotations, présentant l'ensemble des annotations sur une ligne de temps. Enfin, la dernière visualisation, globalement dynamique, affiche certaines annotations (selon leur type) comme sous-titres sur le rendu de la vidéo.

## 4 Travaux proches

Nous présentons rapidement quelques travaux proches, en considérant d'une part ceux qui sont plus axés sur les modèles de description, d'autre part ceux qui s'intéressent plus aux interfaces de visualisation. Il est cependant entendu qu'une description vise souvent à fournir l'architecture d'une interface, tandis qu'une interface de visualisation s'appuie obligatoirement sur un modèle de description des documents audiovisuels.

### 4.1 Modèles de description

MPEG7 [11] vise à enrichir les documents AV en ajoutant des annotations de types variés. Cependant, ce format n'intègre pas de notion de visualisation : il se contente, par conception, de fournir un support de stockage des informations, charge aux concepteurs d'applications de les exploiter. Nous postulons avec Advene que la visualisation est liée plus ou moins fortement avec les annotations, et que tout du moins, il est important de prendre cet aspect en compte lors d'un processus d'annotation. D'autre part, le format informatique utilisé par MPEG7 étant XML, nous pouvons assurer une interopérabilité avec Advene : une simple vue Advene permet effectivement de générer un fichier MPEG7 contenant les annotations désirées.

Le projet OPALES [10] se place dans le cadre d'un travail collaboratif sur un corpus de vidéo. Il suppose une infrastructure matérielle dédiée, et son modèle largement réflexif est séduisant. Cependant, il semble difficilement diffusable vers le public visé par Advene : des utilisateurs non-avertis utilisant un matériel standard.

Le projet ATLAS [7] tente de fournir une abstraction couvrant un large spectre d'annotations linguistiques,



afin d'assurer une interopérabilité de différents corpus. Il reste contraint au domaine linguistique par le formalisme sous-jacent (le modèle de graphe d'annotations), mais fournit une base intéressante, notamment dans sa définition très flexible des régions (l'équivalent des Fragments d'Advène), qui peuvent adresser différents types de données (flux audio ou vidéo, zone géométriques, etc).

## 4.2 Interfaces de visualisation

Plusieurs outils existent qui fournissent un sous-ensemble des fonctionnalités proposées par Advène. Le projet le plus proche est Anvil [6], qui fournit un modèle générique d'annotation, dont on peut enrichir les types, orienté sur l'analyse des dialogues multimodaux. Néanmoins, il reste très orienté comme support d'analyse et n'offre pas de possibilités évoluées de visualisation comme les modes globalement statiques ou globalement dynamiques que nous proposons. D'autre part, il repose techniquement sur la plate-forme Java Media Framework qui ne permet pas à notre connaissance de contrôler un lecteur de DVD, qui est le domaine d'application que nous visons. D'autres projets tels que MagPie [9] ou Transana [13] permettent d'enrichir un document AV par le biais d'annotations, textuelles le plus souvent mais également audio. Cependant, l'unique exploitation envisagée est un rendu de ces annotations au cours de la visualisation du support AV annoté, alors que l'on peut entrevoir d'autres utilisations, comme nous l'avons vu pour Advène, et la contrainte technique (utilisation de JMF) demeure.

En matière de rendu enrichi de vidéo, le projet Hyperfilm [5] explore des pistes plus originales, dont on peut retrouver l'équivalent dans certaines idées d'Advène. L'outil permet notamment d'inclure des liens actifs dans les vidéos, afin d'offrir à l'utilisateur des pistes alternatives de visualisation d'un média. Cependant, comme les projets précédemment évoqués, il se limite à un mode de rendu des annotations (rendu lié au déroulement du support AV).

## 4.3 Temporalité et hypermédias

Parmi les nombreux travaux existant sur la question, certains auteurs ont cherché à catégoriser différents temps, aussi bien liés aux documents (temps narratif, diégétique, etc.) qu'à la lecture elle-même. Par exemple, [8] différencie le « temps interface », lié à la lecture, du « temps cognitif », lié au récit. Le temps interface peut se s'analyser suivant les « temps mécanique » (attente d'une réponse de la machine ou du réseau); « temps de lecture » (passé à lire l'hypertexte); et « temps interactif » (passé à interagir avec le document). Cette analyse se base sur des soumissions plus ou moins grande de l'utilisateur à l'hyperdocument généré, mais ne considère pas le cas particulier de la vidéo, et de sa temporalité physique particulière. On pourrait sans doute considérer un « temps de visionnage » dont la soumission serait intermédiaire entre les temps mécanique et de lecture.

Si la notion de flux de l'utilisateur nous semble peu utilisée dans la communauté hypermédia, les études sur les temps des documents s'intéressent plus à ce que nous avons appelé flux du jeu. [4] différencie par exemple le « temps d'exécution » (*runtime*) du « temps du document » (ligne de temps du concepteur). Le flux de l'utilisateur n'est ainsi pris en compte qu'implicitement à travers l'influence qu'il a sur le flux du jeu.

## 5 Conclusion

Nous avons décrit dans la première partie de cet article les différents éléments du modèle Advène pour représenter des annotations audiovisuelles, et présenter celles-ci à l'utilisateur sous la forme d'hypervidéos, documents hypermédias générés à partir du document audiovisuel original annoté et des annotations. Les hypervidéos sont des documents multimédias complexes, du fait de la part importante prise par l'élément vidéo, qui possède des temporalités particulières. Afin d'être à même de construire des hypervidéos conduisant à des utilisations novatrices du matériel audiovisuel, nous proposons un modèle d'analyse de celles-ci suivant différentes temporalités. Le début de notre modèle d'analyse est présenté dans la deuxième partie de cet article, et illustré par quelques scénarios d'utilisation de l'outil Advène, et hypervidéos associées.

Les développements du modèle et du prototype Advène se poursuivent. Nous disposons d'une base solide, tant au niveau du modèle que de l'architecture, pour explorer les possibilités des hypervidéos. Le cadre des réflexions ébauché dans cet article devra être affiné afin de soutenir notre recherche en termes de conception et d'utilisation de ces nouveaux supports.

## Références

- [1] Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in XML. Recommendation REC-xml-names-19990114, World Wide Web Consortium, 1999.
- [2] Teresa Chambel and Nuno Guimaraes. Context perception in video-based hypermedia spaces. In *Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 85–94. ACM Press, 2002.
- [3] N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part Two : Media Types. RFC 2046, Internet Engineering Task Force, 1996.
- [4] Lynda Hardman, Jacco van Ossenbruggen, K. Sjoerd Mullender, Lloyd Rutledge, and Dick C. A. Bulterman. Do you have the time? Composition and linking in time-based hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia : returning to our diverse roots*, pages 189–196, Darmstadt, Germany, 1999.
- [5] Hyperfilm. Video hyper-media production, 10 2003.
- [6] Michael Kipp. Anvil - A Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of Eurospeech 2001*, pages 1367–1370, Aalborg, Sept 2001.
- [7] Christophe Laprun, Jonathan G. Fiscus, John Garofolo, and Sylvain Pajot. A practical introduction to ATLAS. In *Language Resources and Evaluation Conference (LREC)*, 2002.
- [8] Marjorie C. Luesebrink. The moment in hypertext : a brief lexicon of time. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space*, pages 106–112, Pittsburgh, Pennsylvania, 1998.
- [9] Media Access Generator (MAGpie), 2003.
- [10] Marc Nanard and Jocelyne Nanard. Cumulating and sharing end users knowledge to improve video indexing in a video digital library. In *Proceedings of the first ACM/IEEE-CS joint conference on Digital libraries*, pages 282–289, Virginia, USA, Jun 2001.
- [11] José María Martínez Sanchez, Rob Koenen, and Fernando Pereira. MPEG-7 : The Generic Multimedia Content Description Standard, Part 1. *IEEE Multimedia*, 9(2) :78–87, 2002.
- [12] Nitin Nick Sawhney, David Balcom, and Ian E. Smith. HyperCafe : Narrative and Aesthetic Properties of Hypervideo. In *UK Conference on Hypertext*, pages 1–10, 1996.
- [13] C. Thorn. Creating New Histories of Learning for Math and Science Instruction : Using NVivo and Transana to manage and study large multimedia datasets. In *Conference on Strategies in Qualitative Research*. Institute of Education, University of London, 2002.
- [14] W3C. *Synchronized Multimedia Integration Language (SMIL 2.0)*, 2001. <http://www.w3.org/TR/smil20/>.