Models for sustaining emergence of practices for hypervideo

Pierre-Antoine Champin
Université de Lyon, Lyon, F-69003, France ;
Université Lyon 1, CNRS UMR5205,
LIRIS, Villeurbanne, F-69622, France
pierre-antoine.champin@liris.cnrs.fr

Yannick Prié
Université de Lyon, Lyon, F-69003, France ;
Université Lyon 1, CNRS UMR5205,
LIRIS, Villeurbanne, F-69622, France
yannick.prie@liris.cnrs.fr

ABSTRACT

The work presented in this paper aims at covering several domains: hypervideo modelling, document annotation, and practices sharing and emergence. It is based on the Advene project, providing a model and a prototype for creating, rendering and sharing annotations of audiovisual documents. After a presentation of the notion of multi-structurality in documents and a definition of hypervideos, we present the original Advene model. We then discuss some limitations observed in our model, and introduce a new model for hypervideos. We finally discuss related work with respect to our model, and raise the problem of the emergence of semantics in videos and hypervideos.

1. INTRODUCTION

The work presented in this article covers several research domains. The first domain deals with multimedia document modelling, more precisely hypervideo (as audiovisual-based hypermedia documents) modelling. The second has to do with document annotation, both on the practice (active reading, interaction) and modelling side. The third is more concerned with knowledge sharing, community, innovation and document genre emergence.

Our work is also grounded in the Advene project[1] we have been carrying for several years at LIRIS laboratory. Advene is at the same time a general project on hypervideo and active reading; a model suited to video annotations and hypervideo construction; and a prototype used in several application domains such as video-based interaction analysis or film critics.

We are therefore more interested in personal annotation practices and innovation for hypervideo design which leads to emergent semantics and structures than with top-imposed structure models and semantics. In such a context, our approach on structure-based document manipulation is directed towards personal practices and semantics which can lead by dissemination and emergence to interpersonal practices, semantics, structures and models and then to new document structures and genre stabilization.

The article is organized as follows. We first discuss the notions of document semantics and multi-structured documents in the context of video and hypervideo documents. We then present the current Advene model for annotation and hypervideo construction, which is implemented in the Advene prototype. After a discussion on this first model, we describe its ongoing evolution in a second Advene model. Eventually, we discuss the various concepts we have presented in the article regarding some of the conference issues.

2. STRUCTURES, HYPERVIDEOS AND SEMANTICS

2.1 Multi-structured documents

Digital documents are described using digital logical structures upon which their rendering will be calculated: a XHTML document describes a structure of various kind of logical elements; a PNG image describes an array of colored pixels, an MPEG video describes a structure of synchronized images and sound. Renderers of digital documents (browsers, multimedia players...) use both the structural information in the logical forms of the documents (xhtml, png, mpeg files) and their hard-coded computational rendering information to present the physical form (or the rendered form) of the document.

When it comes to considering the addition of information to documents so as to encompass their basic rendering and their basic experience, other digital structures are needed. For instance, adding a rhetorical information structure to a linear document allows to search its argumentation and to navigate along its arguments, provided that tools capable of using such an added structure are available.

Hence there always exists a tension between the basic digital logical structure and the basic rendering of documents, that correspond to widely recognized documents forms (web documents, video documents, etc.), and the fact that supplementary structures and renderings can be considered. If these reveal successful, then new basic structures will arise, and the corresponding new basic rendering, that incorporate and enhance the previous form of the document (see the evolution of HTML along the 90'). But it can also be

[1] Annotate Digital Video, Exchange on the NEt, http://liris.cnrs.fr/advene

useful to keep the separation between the basic form of the document and its enrichment, and to allow separate tools to give together global rendering (eg. RDF statements in a web documents will not be considered by browsers, but by browsers' plugins).

In the digital document world, a document form is then something that is sufficiently stable to be widely acknowledged, so that everybody agrees on its structures and ways of rendering[2]. But a document form is always challenged by new structures that build upon its basic structure, and new tools that extend or cooperate with basic related tools[3].

Identifying and using an additional logical structure in a document means having the means to inscribe it as a structure related to the basic structure of the document, and tools to make use of it. Simple informal annotation is the simplest way to do so. Adding for instance textual information to a fragment of a document means building a structure (a set) of annotation on a set of fragments defined in the terms of the basic logical structure. Tools for annotating and annotation rendering help build and present the new structure, most of the time by presenting the annotations in the context of the fragments they annotate. Of course, there are much more complex ways to add structure to basic logical structures, for instance so-called "semantic annotation" as considered in the semantic web [17].

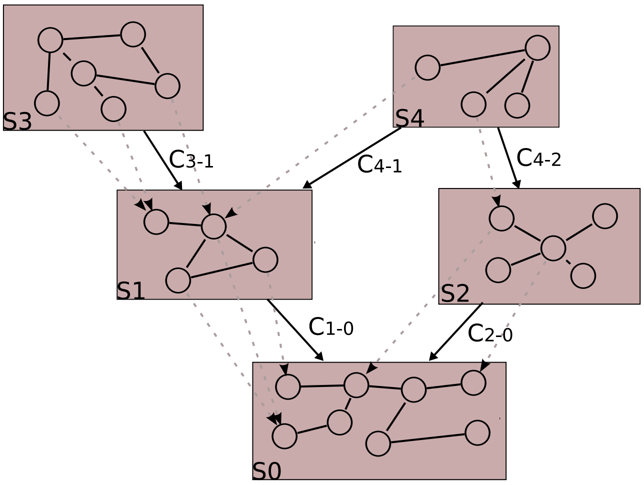

Figure 1: A model for multi-structured documents. S_0 is the base structure. A correspondence function C explains how structures build on other structures.

So as to clarify the study of multiple structures in documents and multi-structured documents, we proposed in [1] a conceptual framework and a model. A multi-structured document is presented as a directed labeled graph of structures, each one being itself a graph (figure 1).One structure S_0 is considered as the base structure, upon which other structures S_i can be built. A correspondence between structures describes how elements from a structure at level i are related to elements from its sub-level structures at level $i-1$ to 0. Each structure S_i is then related to the structure S_0, directly or indirectly.

This model takes into account the fact that basic logical structures S_0 have to be defined for any document, while other structures are build upon it, extending it in various directions, for various purposes. The model also acknowledges the fact that all the structures of a document actually define one global digital structure S, and that each structure S_i indeed corresponds to a consistent subset of it, regarding a certain activity that can be conducted with it.

2.2 Hypervideos

In [3], we defined *hypervideos* as hypermedia documents built upon audiovisual linear documents. Users experiencing hypervideos are given various possibilities of navigating the stream, either from and to static pages (such as tables of content or clickable dialogs), dynamic interfaces (such as timelines) or the stream itself (intra-stream navigation, with hotspot clicking for instance).

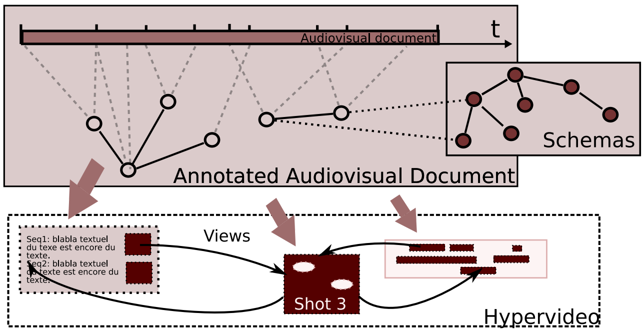
Audiovisual document
t
Schemas
Annotated Audiovisual Document
Seq1: blabla textuel du texe est encore du texte.
Seq2: blabla textuel du texte est encore du texte.
Views
Shot 3
Hypervideo

Figure 2: Hypervideos are audiovisual-based hypermedia documents composed of a set of inter-related views. Views use the annotation and the relation in the annotation structure of the annotated audiovisual document. Schemas constrain the annotation structure.

Hypervideos are rendered using the information from an *annotated audiovisual document*, i.e. from both audiovisual documents (moving images and audio) and from an annotation structure with which it is enriched.

The *annotation structure* is a set of annotations and relations between them. *Annotations* are pieces of information that annotate a fragment of the video document. For instance, an annotation can annotate a shot in a movie, and contain information on the lightning and the camera work for that shot. Another annotation can annotate a whole sequence. *Relations* between annotations can also hold information. For instance, two successive shots can be linked with a relation describing how the transition is done between them (e.g. cut, dissolve...).

We define *schemas* as constraints on the annotation structure. For instance, a "Decomposition" schema can specify that one should annotate a video document with annotations of type "Shot" (containing several attributes, one being "CameraWork") and "Sequence" (containing a textual

[2] This often leads to normalizing the basic structures in formats that permit interoperability of tools.
[3] Some Web 2.0 mashups fall into such category.

description of what happens in that sequence), and relations of type "NextShot" between two "Shots" (containing a description of the shot transition).

We finally define a *view* as a way of rendering information from the annotated audiovisual document. Therefore a view can render information from the annotations (e.g. a table of content built from the "Sequence" annotations), from the stream (e.g. a classical video player), or from both (e.g. a video player allowing shot and sequence navigation, or a table of content displaying images or sound extracted from the stream). A hypervideo is defined as a consistent set of views related to one or more annotated audiovisual documents.

There are therefore four poles in our way of conceptualizing hypervideos: 1/ the videos themselves, 2/ the annotations structures that enrich video information, 3/ schemas that specify how to build annotation structures, 4/ views that specify how to render hypervideos, see figure 2.

2.3 Structures and semantics of hypervideos

Having presented how we consider multiple structures in documents, and the notion of hypervideo, we are now able to focus on multiple structures in the particular context of hypervideos.

Considering an annotated audiovisual document, we will define S_0 the basic structure of the audiovisual document as a sequence of images synchronized with sound, which allows us to define fragments in it. The annotation structure (the graph of annotations and relations) can then be considered as one supplementary structure S_a, that corresponds to all the annotations that have been added to the audiovisual document. But we can also decide that there are multiple S_i structures, both constructed upon S_0 on the one side (because annotations are directly linked to the stream through fragments) and upon other structures if needed, on the other side.

For example, S_0 being the structure of a movie as a video stream, S_1 could be the decomposition of the movie (movie, sequences, shots), S_2 could add the occurrences of the characters to the shots, S_3 could build directly upon S_0 so as to present the mood of the movie, see figure 3.

Having annotated a video so as to create an annotated audiovisual document, views can be defined, that specify how the information of the video (corresponding to S_0) and the annotation structure (S_a or $\{S_i\}$) can be rendered into a hypervideo.

We said earlier that textual annotation was the simplest way to extend the structure of a document, with a simple basic rendering. An equivalent in the context of hypervideo consists in basically annotating the audiovisual document with textual items related to time intervals, that are presented in a horizontal timeline displaying the temporal extension of the annotations as boxes and their content as labels on the boxes. In such a context, S_a is a very basic structure related to S_0, and the timeline view presents annotations equally regardless of their content. The user may only fill that additional structure with his textual annotations, and render it with the predefined timeline view provided to him.

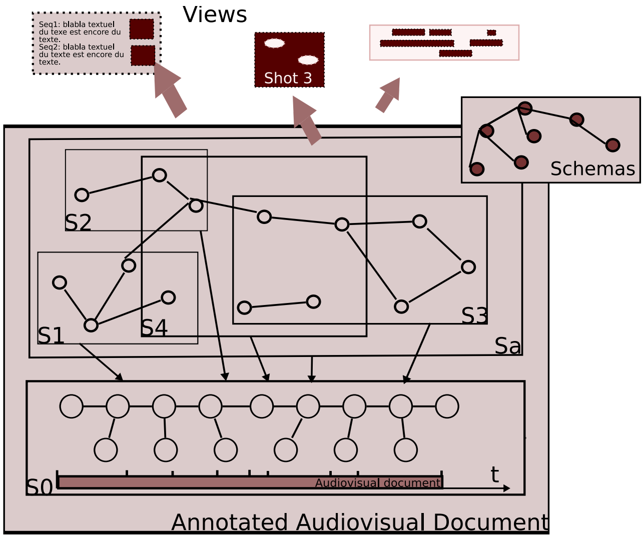
Views
Seq1: blabla textuel du texe est encore du texte.
Seq2: blabla textuel du texte est encore du texte.
Shot 3
Schemas
S2
S1
S4
S3
Sa
S0
Audiovisual document
t
Annotated Audiovisual Document

Figure 3: Multi-structures in hypervideos. S_0 represents the video document. S_a is the annotation structure, into which one or several multi-structures can be defined. Structures in S_a are defined within practices together with their associated schemas and views.

But more complex views can be defined, for which the rendering of the structure S_a can partly be defined by the user, who can then specify more precisely what to do with the annotations, depending on their content, fragment duration, or any discriminating information. In that case, we can say that the user both defines the annotation structure and the way this structure will be rendered, contributing to the proposal of both a new structure and a new rendering for video documents.

Moreover, as schemas constrain the annotation structure, they are part of the means a user can abstract and reify the way he has annotated, and control the way he will further annotate. Schemas hence represent a means to describe how to annotate a video in order to ensure that the annotation structure can be rendered in a hypervideo through a set of views.

We therefore claim that the semantics of the annotated audiovisual document resides resides in the annotation structure (data added to the video, enriching the document structure), in the views (the way the document structure is rendered) and in the schemas (the way the enriched structure is abstractly defined, which could be thought of as a format). That semantics is not totally explicit nor formal, but it indeed resides in those elements for they are used to sustain a real *practice*. For instance,

- if we have a schema that defines a "Decomposition" of a movie in sequences and shots;
- if, for a particular movie, we have defined several "Sequence" and "Shot" annotations and "NextShot" relations according to that schema; and
- if we have defined three views, the first for a "Sequence summary" of the movie, the second for a "Sequence

and shot navigation player", the third being an illustrated list of one's ten preferred shots;

- then we have defined a new practice for this very movie experience based on its decomposition and on one's favorite shots.

The semantics related to this practice resides in the "Decomposition" schema together with the three views, together with the annotations and the relations. If we disregard the third view, then we can consider that we have defined a more general practice related to *any* film decomposition, which semantics can be abstracted as residing in the "Decomposition" schema and in the "Sequence summary" and "Sequence and shot navigation player" views, being exemplified by the actual annotations of our particular film.

Accepting that the semantics resides in reified practices means that we can consider as many substructures S_i in S_a as there are identified practices in the hypervideo. An annotation structure is therefore composed of multiple, possibly overlapping structures, possibly corresponding to different schemas, relevant in different usages (e.g. a "character + shot substructure" differs from a "shot substructure", both being used within different renderings, for instance one oriented towards narration study, the other towards film making study). Multi-structurality is therefore natural in the hypervideo model we propose.

We consider that we are just at the beginning of the development of new forms of multimedia documents such as hypervideo documents[4]. Our claim is therefore that so as to favor the emergence of useful hypervideo practices, we should *under-specify* hypervideo models in order to offer freedom to define whatever users want to describe (abstract knowledge: schemas, direct knowledge: annotations, use knowledge: views). Of course, this entails that means should be given to share multimedia semantics so that users can reuse and construct upon what other users have done. This approach is different from a top-down approach which would imply trying to specify (and normalize) usages before they even exist, a problem encountered by MPEG7 [18]. The user should have the freedom to define her own semantics with all the concepts/tools at her disposal. Doing this will define the semantics of the video and hypervideo documents, as it will define both structures and means of using them, be it for indexing documents or corpuses of documents, retrieval, manipulation, generation, etc. Semantics will then be defined in the different ways of using the structures and by the actual uses of the different structures.

The remainder of this article is dedicated to the presentation of two models for hypervideo documents representation in Advene (current and future models). These models have been designed keeping in mind our general objective of favoring the emergence of multimedia semantics as explained above.

3. ORIGINAL ADVENE MODEL

This section describes the Advene model implementing the notion of hypervideo described in section 2.2. Figure 4 gives a global view of that model as a UML class diagram. We describe in the following the role of each class.

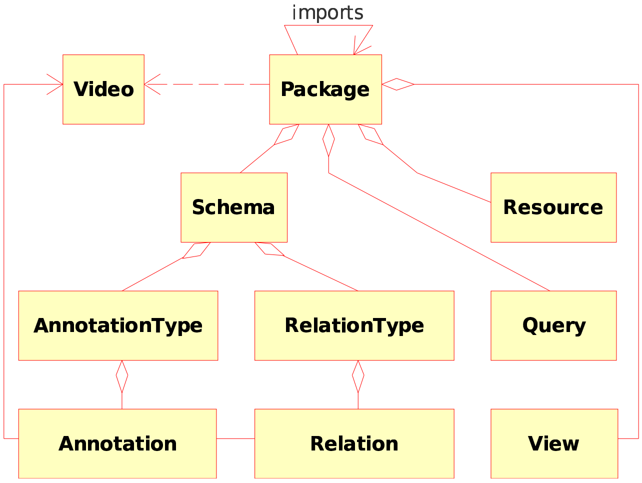

Figure 4: The first Advene model

3.1 Annotation structure

In Advene, information is attached to a video under the form of annotations. More precisely, an *annotation* is constituted of a *content* attached to a *fragment* of the video. The content can be any piece of information: plain or rich text, sound, image, structured data (XML, RDF...), or even another video. The fragment is minimally defined as a temporal interval over the video, but can possibly be enhanced by other information: a spatial region, a specific angle, soundtrack or subtitle track (in DVDs), etc.

Annotations can in turn be linked to other annotations by *relations*. A relation can link an arbitrary number of annotations (the *members* of the relation), and can optionally have a content of its own (a piece of data attached to the relation itself).

3.2 Schemas

Every annotation and relation belongs to a unique *type*. Each type has a name, a short description of the intended semantics of its annotation or relations. Annotation types furthermore constrain the kind of content that its annotations can have (basically as a MIME type). Relation types constrain the number of members, the types of the members, and the kind of content that its relations can have. According to our hypervideo model, annotation types and relation types are not defined independently but in the context of *schemas*.

For example, the schema "Decomposition" described in section 2.2 would contain two annotation types and one relation type. Annotation type "Shot" would contrain the content of its annotations to an ad-hoc structured type (like an XML schema dedicated to movie description), while annotation type "Sequence" content type would be plain text. Relation type "NextShot" would constrain its relation to have

[4] We can identify at least two reasons for it. First, the relative novelty of digital audiovisual documents, which have become massively widespread only for a short time. Second, the particular status of audiovisual compared to text: texts, strings and alphabetical characters just do fit better into computers than images and sounds.

exactly two members of type “Shot”, with a content taken in an ontology of video transitions. The description of the schema would further document the intended use of those type (e.g. “NextShot” relations should only appear between successive shots).

3.3 Designing hypervideos

Annotations and relations constitute, with the annotated video, raw material for new hypervideos. *Views* are the components which use that material to render it into the actual hypervideo. There is no assumption in the Advene model about the format or modalities of the data produced by a view: it could for example be a hypertext, an image, a viewing of (parts of) the video, possibly augmented with information on or around the video. It could also be any composition of such elements. Views can be interactive, and can possibly lead to other views. According to our definition, a hypervideo is consituted of a set of related views linking to each other.

Though the structure of schemas and types presented before can be used in the design of views, it is sometime useful to access more specific sets of elements to be rendered. For example, a view might be using only “Shot” annotations indicating a travelling; another view might be interested in annotations from any type, but containing the word ”Lyon” in their content. For this purpose, *queries* can be used to filter the whole set of elements into a subset relevant to one or several views.

Finally, some views may require some data from neither the video nor the annotation structure: a CSS stylesheet, an image... Such accessory contents are stored in *resources.* In contrast to annotations and relations, resources have no explicit link to the video.

3.4 Exchanging hypervideos

A self-sufficient set of the Advene elements described above is called a *package.* A package can be stored as a digital document in order to be easily edited, shared and reused. It can be used either to render the hypervideos defined by its views, or merely as base for building more hypervideos from its components.

An important feature of Advene is that the package is not supposed to contain the annotated video. In order to avoid the legal problems linked with copyrighted documents, we assume that each user of a package has otherwise acquired the same video as the others. Note that some views might not use at all the video, hence be usable even by a user owning no copy of the annotated video.

Another focus of Advene is the sharing of annotations and other elements. Imagine Alicia defining the “Decomposition” schema described above, and annotating her favourite movie with it. She also defines a view listing all the shots in a web page. Alicia shares her package on her website. Brian is interested in the schema and the view, but would like to reuse it to annotate his own favourite movie. He does so and shares his package as well (together with information about the DVD edition he annotated so that interested users may buy the same one in order to reuse that package). Note that Brian didn’t have to define a new view, since Alicia’s view (applied to his annotations) gave a satisfying result: their practices (and underlying semantics) are not significantly different. Chris likes Brian’s movie as well, but would like to elicit the diegetic chronology between the shots (that movie has a lot of flashbacks). She reuses Alicia’s schema and enhances it with the “Diegetic order” relation type (between two “Sequence” annotations, describing the approximate amount of diegetic time between the sequences), and adds such relations between the sequences defined by Brian. She then defines a new view to play the movie in the diegetic order, and another view to add a subtitle to the movie everytime the diegetic chronology differs from the montage chronology of shot sequencing.

Although that scenario could be realized with a lot of copy-paste (and although in most situations, users will actually do it that way), Advene packages are able to *import* elements from other packages. Imported elements are not stored in the importing package, but retrieved from the defining package. This allows for modular design of packages, especially in the case of general purpose schemas and views.

3.5 Prototype

The Advene model has been implemented in the Advene prototype for video annotation and hypervideo design. The open source multi-platform prototype can be freely downloaded from our website, together with some example packages. As described in [3], the general architecture features a python application, a enhanced video player and a webserver. Three categories of views have been defined:

- Ad-hoc views present the annotation structure and the video in pre-programmed components of the application; e.g. a complex timeline showing and giving access to the annotation structure.
- Dynamic views use the temporal playing of the video stream, in the enhanced player, as an event source. Events can come from user interaction (play, stop, fastforward) or from the annotation structure (encoutering the start or end of an annotated fragment). They trigger various multimedia presentation directly in the player or in the application; e.g. a view that present, as a caption on the video, the title and number of a shot each time an annotation of type “Shot” occurs.
- Static views are template-based XHTML documents, provided by the web server to a web browser. They query and present information from the annotation structure and the stream; e.g. a table of the sequences, presented with their title and a keyframe.

Access to the annotation structure and to the video player has been defined in the Advene prototype using the path-based TALES expression syntax.

4. REQUIREMENTS

The prototype described in section 3.5 has already been used within different video active reading application fields, mainly film study and interaction study, in collaboration with specialists. User feedback has allowed us to greatly improve the ergonomy of the prototype, both to make it more accessible to users without much technical background, and to provided desired functionalities to more advanced users. However, some of the requirements expressed by users have been hampered by the underlying model of Advene.

4.1 More flexible organization of the annotation structure

Annotations and relations must belong to an annotation type and a relation type, respectively. However, that constraint has proved too strong for some users of Advene, using annotation in a note-taking way, where the schema is designed concurrently to the annotation process itself. Such users need to create *untyped* annotation, then decide later how to organise them into types and schemas.

It appears furthermore that the specification of annotation types mainly by the kind of content they can hold is a fairly technical one, while types are rather intended to capture semantics of the annotation structure. We envision that in some applications, annotation types might accept heterogeneous content types (although this need has not been expressed yet by any user).

Finally, annotations and relations could sometimes be grouped according to criteria that are independant of their type, but that can not be automatically computed by a query.

4.2 Relating views with schemas

Although views are typically designed in conjunction with schemas, the Advene model offers no means to elicit that relation between them (at least not besides their appearing in the same package or the textual documentation of schemas). The same is true of queries and resources, on which views can implicitly depend without any mean to make that dependency explicit. Such explicit knowledge would however help software in providing a better assistance to user when sharing, reusing and adapting packages.

4.3 Importing annotations/relations

While the model theoretically allows a package to import any element from another package, the implementation does not allow annotations nor relations to be imported. This stems from the fact that, while it is possible to indicate which schema or view one wants to import, identifying individually every annotation to be imported is not convenient. On the other hand, importing systematically all the annotations from a packages does not seem satisfying either, and even schemas or annotation types are too coarse-grained in some situations. This problem raises again the need of being able to group annotations and relation with more ad-hoc criteria.

4.4 Metadata about the video and multiple video

The Advene model offers no real means to describe the video annotated by a package. For the moment, that information is mainy conveyed by other means: in specific views of the package, for the user to read, or on the web page where the package is shared. This lack of video representation also hinders the possibility to annotate multiple videos in a package. Though possible in theory, this is highly inconvenient from the point of view of application developpers, and hence not implemented in the prototype.

5. MODEL EVOLUTION

In this section, we propose an evolution of the Advene model. This new model is represented as an UML class diagramm in figure 5.In the following, we describe that model by focusing on the differences with the old one.

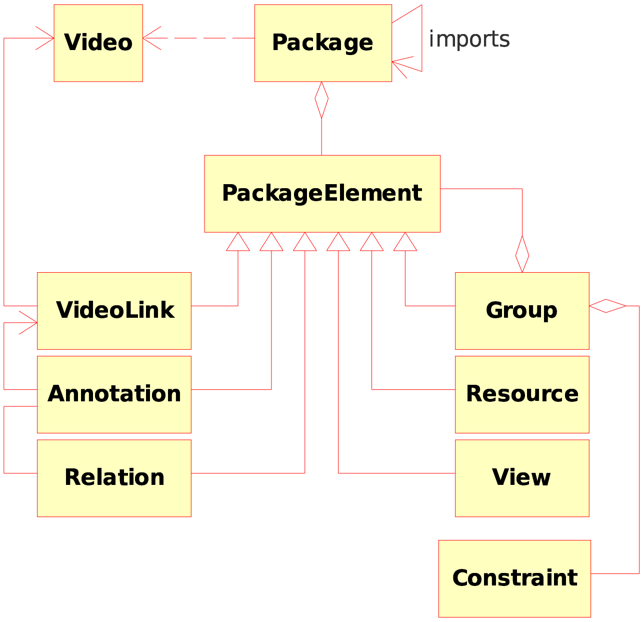

Figure 5: The future Advene model

5.1 Packages and package element

The notion of package has not changed. It is still composed of various elements, depends on one or several videos, and imports other packages.

A salient difference between both diagramms is the presence of the new PackageElement superclass of all the elements contained in a package. This difference is however merely a notation convenience, since all components of a package could have had a common superclass in the first model. We didn't represent it because it had no conceptual role. On the contrary, the PackageElement class in the new model is in relation with the Group class, which we will discuss later.

5.2 Annotation structure

The modelling of the annotation structure is essentially the same as in the old model. A minor difference is that annotations and relations are no longer required to have a type (see section 4).

A more important difference is that video are now represented in the package (class VideoLink). Note that this new class does not hold the actual video, still for legal reasons. However, it may contain arbitrary meta-data allowing to identify the video. Those meta-data can help checking that a video is the same as the one used by the author of the package (e.g. MD5 hashcode of the video) or at least might be "compatible" (e.g. precise duration). They can also help retrieving a copy of the video (e.g. ISAN number, VoD vendor reference...). Note also that this explicit representation of video links makes it easier to annotate several videos in a single package.

5.3 Groups

Groups have replaced, in the new model, a number of different concepts from the old one: schemas, types and queries. A group is defined as a collection of arbitrary package elements. Note that the same element can belong to several groups. As we will discuss in the section 6, we foresee that groups may actually be used to implement many actual or future practices. A group has a name and a textual description allowing its creator to explain the rationale of the group. We now discuss several kinds of groups.

5.3.1 User-defined groups

As their name implies, user-defined groups have their elements explicitly set by the user, who can both add individual package element and include all the elements from other groups. Furthermore, the user can express a set of *constraints* that all elements of the group must satisfy. It allows the user to formalize and make operational, to a certain extent, the intended semantics (or prescribed use) of the group. Authoring tools should be able to check those constraint in order to notify the user if the group contains an incorrect element or if an incorrect element is to be added. It is not expected that tools *prevent* the addition of an incorrect element: indeed, we already argued in favor of *a posteriori* specification. The addition of an element violating the constraints may mean that the elements should not be added, but it may as well mean that the constraints need to be reconsidered.

5.3.2 Queries

Queries in the new model are very similar to those from the old model. The difference is that they are homogeneous to groups (a set of elements). Hence they are now considered as a special kind of group, defined in intension while user-defined groups are defined in extension. Queries are to user-defined groups what *virtual folders* are *saved searches* are to folders in some operating systems or e-mail softwares[5].

5.3.3 Imports

From the point of view of a package, all the elements from an imported package are a contained in a specific *import* group. This comprise all the groups defined in the imported package, so every subset of its elements deemed relevant by the imported package's author is accessible by the importing package as well.

5.4 Structuring with groups

We now describe how groups (as means of structuring annotation structure and schemas) can be used in a number of ways to enable practices suggested in our definition of hypervideo, practices required by Advene users, and other, yet to be invented, practices.

Note that user interfaces can assist those practices (with *ad hoc* metaphores and widgets) or even enforce them. It may indeed be a good thing to present novice users with a directive and restrictive interface that do not give access to the full genericity of the model, in order to guide such users in identified good practices; this is what so called *wizards* do in many software applications. Expert users can nevertheless use functionalities beyond the ones offered by wizards.

Furthermore, a generic model opens the way to inter-operability between applications. This is all the more important in the field of audiovisual documents that practices are not yet well defined, and individual tools are bound to be incomplete with regard to a user's needs.

5.4.1 Types and schemas

A schema has been defined as a consistent set of annotation types and relation type. We can see it as a special case of user-defined group, provided that the *constraints* on that group impose that it contains only annotation types and relation types — and provided that the practice prescribed by the schema is documented in the group's description.

It appears that annotation types and relation types themselves can be defined in term of user-defined groups: an annotation type is nothing more than a set of annotations, constrained to have a particular content type. Respectively, a relation type is a set of relations with constraints on the number and types of their members. Groups actually offer means to specify more precisely the semantics of types (depending of course of the expressive power of the language used to describe constraints). Annotations and relations may not be immediately typed (as required by some users), and may easily change type (provided that they comply with the new type's constraints). Moreover, a specific annotation type can be used in several schemas, corresponding to its use and reuse across different practices.

5.4.2 Tags

Tags have become a popular means to index and share content in many applications, especially on the web [7]. User-defined groups are similar to tags: they have a name, assumed to be descriptive of the elements its contains, and an element can belong to an arbirtary number of groups (as an element can hold an arbitrary number of tags).

Inside a single package (possibly edited by several users), the mechanism of user-defined groups can be seen as a tagging system. According to the classification proposed in [7], that system allows free-for-all tagging (the Advene model does not include access limitations to elements of a packages), using the set-model (an element can not be several times in a group).

Note however that this analogy becomes more problematic when sharing and importing packages comes into play. Groups are local to a package: even if two packages use the same name for a group, both groups are different. In a tagging system, when two users use the same word, the tags are viewed as identical. This is a central feature of tagging system for it enables sharing and emmergence of folksonomies, but is not appropriate for other intended uses of groups in Advene. An appropriate instrumentation of groups may however help to achieve a real tagging system across imported packages.

[5] We can also draw a parallel with tables and views in relational database management systems, though one must not confuse them with *views* in our model

5.4.3 Hierarchy

Though elements of a package can belong to an arbitrary number of groups, user-defined groups can also be used as *folders*, with a strict hierarchical structure. This can be achieved by encoding the hierarchy in the names of the groups: group `A_B` would e.g. be considered as a "subgroup" of `A` that would appear to be simply named `B`. This technique is sometimes used in tagging systems to introduce a hierarchy of tags. It is also used in the current Advene prototype for resources. Of course, an appropriate interface is required to "decode" those names and render it properly as a hierarchy.

6. RELATED WORKS AND DISCUSSION

A problem with multimedia documents, in particular audiovisual documents, is that the exchanged forms of those documents convey almost no explicit information about the authoring process. Not only does this make it difficult to index and retrieve those documents for common uses, but it also hinders the emergence of new practices.

In this section we discuss the Advene model and compare it with other existing works. We first focus on the structural aspects of multimedia document engineering. We then raise the question of the semantics of audiovisual documents, and the current and future status of hypervideos as documents.

6.1 Multimedia document engineering

A central feature of the Advene model (both the original and the new one) is the clear separation between the audiovisual document and its metadata, and the fact that the latter allows for multiple different structures, with different intended uses. Many systems today do not allow such a separation: subtitles and chapter structures in a DVD are integrated in the data stream. This makes them hardly manipulable, and indeed their reuse appears to be neither envisionned nor desired. The Annodex format [10] also aims at embeding (rendering oriented) metadata in the audiovisual stream. SMIL [19], a format for freestanding metadata, is merely a presentation format, without a proper notion of annotation.

On the other hand, the MPEG7 format [12] provides a separate storage for metadata, enabling multiple annotation dimensions, but tools allowing to use it [8, 11, 13, 15] are not widely used yet. Recent web-applications like Mojiti[6] and BubblePLY[7] allow to annotate videos from external sources in order to enhance them with elaborate captions and hyperlinks. Those annotations are nevertheless designed for a specific rendering, and their sharing in order to be reused is not emphasized.

Note that multi-structurality was not a design constraint on the advene model. It is rather a consequence of the underspecification of the model with respect to pre-defined structures, in favour of means to define one's own structures. As we demonstrated in the previous section, this enables the individuation of different, possibly overlapping, structures. An Advene package has therefore more the status of a document generator (by rendering one or several structure through views) than of a document. However it has its internal structure (annotations, groups, views) allowing the reuse of all or parts of it.

We also acknowledge the fact that the requirement for genericity and reusability induces a tension with simplicity and usability. We try to keep the Advene model generic enough to enable innovative practices, but still able to capture the semantics of common usage. The model proposed by [9] is much more generic, but specific structures have therefore to be buried in the content of information units rather than explicit in the model.

6.2 Semantics of videos and hypervideos

We do not consider semantics as an intrinsic property of audiovisual documents, as implied by the notion of *role* in [9]. We are rather interested in their "operational" semantics: the semantics is linked to the use of videos, it emerges from (actual and yet to be defined) practices and practitioners' needs.

Considering multi-structurality gives us a means to grasp that semantics. The original temporal structure of audiovisual documents stems from the most basic usage — linear playing. New structures are set up for new usage, and their link with previous structures is a trace of the latter being reused to construct the former: semantics evolves as structures evolve and superimpose on others. Interactions with the structures through views creates in turn a new document, with its own semantics.

Formalizing semantics of documents is the primary concern of research on the Semantic Web, and a number of works have considered using ontologies to structure [4, 16, 6] and annotate [5, 14] multimedia documents[8]. We have begun to study the integration of Semantic Web technologies and tools into Advene: in [2], we propose to use specific views to render annotation structures in OWL, and we use an OWL-inference engine to implement *semantic queries*. The new Advene model presented in this paper fits well with our previous propositions: groups could for example be used to mimic the lattice of concept of an ontology.

7. CONCLUSION

In this paper, we have first presented a discussion on document structure and genre evolution, in relation with previous work on multi-structurality of documents, and a definition of the emerging notion of hypervideo. We then presented the original Advene model for hypervideos engineering, that has been implemented and used in different fields. Feedback from users and further researches have allowed us to state a number of further requirements for the model, and we have presented an evolution of the Advene model. We finally discussed that model by comparing it to other works in the fields of multimedia authoring and annotation.

Work in the Advene project will continue in several directions, within several fields of audiovisual active reading practices related to film analysis for critics or teachers[9] and in-

[6] `http://mojiti.com/`

[7] `http://www.bubbleply.com/`

[8] See also the Multimedia Semantics incubator group: `http://www.w3.org/2005/Incubator/mmsem/`

[9] Through *Cinélab*, a french Research Agency funded project

teraction analysis for humanities researchers[10]. We consider practical field work an important constraint when it comes to elaborating models for hypervideos, a kind of document that do not really exists with widespread practices and genres yet.

At the theoretical level, we are currently working on "light" knowledge models for audiovisual active reading, and trying to get further theoretical developments into the notion of "practice and semantics" we proposed here, in relation with the various structures we identify in hypervideo descriptions. On the formal semantics side, we are also actively working on integrating some semantic web technologies into Advene, studying how documentary structures and semantic structures can mix and/or cooperate within documents and practices.

The implementation of the next Advene model has begun, which will lead to an evolution of our Advene tool for hypervideo creation and rendering. This prototype is used as a testbed for rapidly prototyping our various ideas for hypervideos within our various research directions. Hence we hope to participate to the emergence of hypervideos as new kinds of documents, with their own semantics, structures and practices.

8. REFERENCES

[1] R. Abascal, M. Beigbeder, A. Bénel, S. Calabretto, B. Chabbat, P.A. Champin, N. Chatti, D. Jouve, Y. Prié, B. Rumpler, and E. Thivant. Documents à structures multiples. In *SETIT 2004*, Mar 2004.

[2] Olivier Aubert, Pierre-Antoine Champin, and Yannick Prié. Integration of semantic web technology in an annotation-based hypervideo system. In *SWAMM 2006, First International Workshop on Semantic Web Annotations for Multimedia, 15th World Wide Web Conference*, may 2006.

[3] Olivier Aubert and Yannick Prié. Advene: active reading through hypervideo. In *ACM Hypertext'05*, pages 235–244, Salzburg, Austria, Sep 2005.

[4] Jane Hunter. Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In *International Semantic Web Working Symposium (SWWS)*, Stanford, Aug 2001.

[5] Antoine Isaac and Raphaël Troncy". Using several ontologies for describing AV documents : a case study in the medical domai. In *2nd European Semantic Web Conference, Workshop on Multimedia and the Semantic Web*, Heraklion, Crete, May 2005.

[6] Faith Lawrence, Mischa M. Tuffield, Mike O. Jewell, Adam Prügel-Bennett, David E. Millard, Mark S. Nixon, Monica Schraefel, and Nigel R. Shadbolt. OntoMedia - Creating an Ontology for Marking Up the Contents of Heterogeneous Media. In *Proceedings of Ontology Patterns for the Semantic Web ISWC-05 Workshop*, Galway, Ireland, 2005.

[7] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, ToRead. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, Odense, Denmark, may 2006.

[8] Xiangming Mu and Gary Marchionini. Enriched video semantic metadata: Authorization, integration, and presentation. In *Proceedings of the Annual Meeting of the American Society for Information Science and Technology*, pages 316–322, 2003.

[9] Marc Nanard, Jocelyne Nanard, and Peter King. IUHM: a hypermedia-based model for integrating open services, data and metadata. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*, pages 128–137, 2003.

[10] Silvia Pfeiffer, Conrad Parker, and Claudia Schremmer. Annodex: a simple architecture to enable hyperlinking, search and retrieval of time-continuous data on the web. In *5th ACM SIGMM International workshop on Multimedia information retrieval*, pages 87–93, 2003.

[11] Silvia Pfeiffer and Uma Srinivasan. TV Anytime as an application scenario for MPEG-7. In *Workshop on Standards, Interoperability and Practice, ACM Multimedia 2000*, Los Angeles, Oct 2000.

[12] José María Martínez Sanchez, Rob Koenen, and Fernando Pereira. MPEG-7: The Generic Multimedia Content Description Standard, Part 1. *IEEE Multimedia Journal*, 9(2):78–87, Apr-Jun 2002.

[13] R. Schroeter, J. Hunter, and D. Kosovic. Vannotea - a collaborative video indexing, annotation and discussion system for broadband networks. In *Workshop on "Knowledge Markup and Semantic Annotation"*, 2003.

[14] G. Stamou, J. van Ossenbruggen, J.Z. Pan, and G. Schreiber. Multimedia annotations on the semantic web. *IEEE Multimedia*, 13(1), mar 2006.

[15] Tien Tran-Thuong and Cécile Roisin. Multimedia modeling using mpeg-7 for authoring multimedia integration. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, 2003.

[16] Raphaël Troncy. Integrating Structure and Semantics into Audio-visual Documents. In *Second International Semantic Web Conference (ISWC2003)*, pages 566–581, Sanibel Island, Florida, USA, 2003. Springer.

[17] Victoria Uren, Philipp Cimian, Josè Iria, Siegfried Handschuh, Maria Vargas-Vera, Enrico Motta, and Fabio Ciravegna. Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Journal of Web Semantics*, 4(1), 2006.

[18] J. van Ossenbruggen, F. Nack, and L. Hardman. That obscure object of desire: multimedia metadata on the web, part-1. *IEEE Multimedia*, 11(4), dec 2004.

[19] W3C. *Synchronized Multimedia Integration Language (SMIL 2.0)*. W3C, 2001. `http://www.w3.org/TR/smil20/`.

with partners Centre Pompidou, Forum des Images and Antenna Audio and work with cinema teachers at CRDP Lyon (Pedagogical Ressources Department)

[10] IncorporAction project, with ICAR - CNRS laboratory.