From video information retrieval to hypervideo management

Olivier Aubert
LIRIS - FRE 2672 CNRS - Université Lyon 1
olivier.aubert@liris.univ-lyon1.fr

Yannick Prié
LIRIS - FRE 2672 CNRS - Université Lyon 1
yannick.prie@liris.univ-lyon1.fr

Abstract

The digital revolution that has somehow taken place with the World-Wide Web took advantage of the availability and interoperability of tools for visualisation and manipulation of text-based data, as well as the satisfying pertinence of search engines results that makes them usable by non-expert users. If we are to undergo such a revolution in the audiovisual domain, a number of issues mainly related to the temporal nature of audiovisual documents have to be resolved. In this article, we expose our view of the current state of Audiovisual Information Systems, and suggest that they should be considered as video information management systems rather than video retrieval systems, in order to foster new uses of video information. We define the notion of hypervideo that can be used as an analysis framework for this issue. We then describe our implementation of hypervideos in the Advene framework, which is aimed at DVD material annotation and analysis. We eventually discuss how the notion of hypervideo puts video information retrieval in a somewhat different light.

1. Introduction

The success of the WWW has given access to a wide collection of documents, mainly textual ones[1]. To us, *accessing documents* means: a/ the ability to visualise documents ; b/ the ability to search for documents (through search engines) ; c/ the ability to refer to documents (through hyperlinks) ; and d/ the ability to manipulate documents or parts of documents (by saving them, or by copy-pasting).

The worldwide success of the WWW does not only stem from its academic origin where scientists needed to share research information. It is also to a large extent due to its dissemination to a wider audience. This dissemination has been effective because of the availability and interoperability of tools for visualisation and manipulation of text-based data (e.g. web browsers + text processors), as well as the satisfying pertinence of search engines results that makes them usable by non-expert users. The current semantic web efforts aim at improving this interoperability.

While important research efforts are carried out to extend the four above capabilities to audiovisual documents, the same level of achievement has not yet been reached. Indeed, if some features are common to all types of documents (authorship, date of publication, etc), the most important ones are often specific. For instance, images are not composed of such useful items as words for texts, and the temporal nature of audiovisual documents induces original issues hardly translatable from textual documents. In this context, our team –working on the domains of document engineering, documentary information systems and annotation systems– is interested in the specificities of AudioVisual Information Systems (AVIS) and their emerging uses.

In this article, we expose in section 2 our view of the current state of AVIS, and suggest that they should be considered as video information management systems rather than video retrieval systems, in order to foster new uses of video information. The notion of hypervideo, which we propose in section 3, can be used as an analysis framework for this issue. Section 4 describes our implementation of hypervideos in the Advene framework, which is aimed at DVD material annotation and analysis. The last section is dedicated to discussing how the notion of hypervideo puts video information retrieval in a somewhat different light, and how the Advene project contributes to this evolution.

1 From now on, we will refer to them as "text-based documents", in opposition to "audiovisual documents", mainly based on moving images and/or sounds.

2. A view on Audiovisual information systems

AudioVisual Information Systems (AVIS) should provide means to search, retrieve and use audiovisual documents. They are still an active research domain [20, 7, 21] though some of them are already in wide use (Virage [24], used by CNN for instance). AVIS processing can be analysed using the following steps: indexing/retrieval, results selection, results exploitation. Indexing is the base upon which retrieval is built (CNN's video search proposes to retrieve video fragments based on the presence of keywords in text transcriptions). Once a (potentially large) set of results has been obtained, the user has to evaluate their relevance and select the most important ones (in CNN's video search, some context in the form of a keyframe and part of the transcription is provided). After the relevant fragments have been selected, they can be exploited. This step depends on the needs of the user but most of the time, it only consists in visualising the selected videos or fragments.

We will detail these steps and see that the last one is often overlooked and not well integrated with the others. We will then argue that indexing and retrieval should be integrated in the exploitation phase as a video processing cycle.

2.1. Video indexing and retrieval

Video retrieval is now recognised as a specific domain. For instance, the TREC conference series, which provides a test-bed for the comparison of information retrieval systems, integrates a video corpus since 2001, and has made the video track an independent workshop and evaluation process called TRECVID [13] since 2003.

Most AVIS use video and associated descriptors to access specific video documents or fragments. The range of video descriptors is very large, from very low-level features that are extracted automatically to more high-level indexing resulting in natural language descriptions. Indeed, while in textual documents, indexes are existing and objective units (words, characters), no such existing indexes are available for audiovisual documents [2]. They have to be extracted or manually defined, a process producing textual indexes (symbolic descriptors, transcriptions, textual annotations, ...) on which more common retrieval techniques can be used.

Low-level features (color or shape analysis, sound level analysis) generate information that has to be interpreted in a specific context (specialised domain) in order to be useful. Other automatic approaches (shot detection, closed-caption detection, voice processing for transcriptions) result in high-level information (shots, texts, etc.). Eventually, manual indexing produces synthetic information (structured descriptions or texts).

2.2. Query results selection

Once the interesting information is located, it has to be visualised in order to select the most relevant items. When dealing with text documents, it is easy to quickly scan a document or a fragment in order to determine if it is suited to the user's need. On the contrary, audiovisual documents possess a temporality which makes them inherently not instantly accessible. Surrogates [10] have to be used in order to summarise the video information and quickly analyse results of queries on audiovisual documents.

The visualisation of these collections commonly uses hypertext documents illustrated by keyframes from the videos and extracts from bibliographical descriptions. For instance, Graham *et al* [6] proposes a static view of a video document using key frames extracted from the video, in order to quickly browse through a number of video fragments. The VAST system, by Mu and Marchionini [10], also provides multiple means to quickly visualise video data either statically or dynamically (fast-forward, etc).

2.3. Query results exploitation

Once the query results have been selected, they must eventually be exploited. Dealing with text documents, or fragments thereof, is common these days: hypertext has become mainstream and the public daily uses text-based search engines and word processors for textual document generation using cut-and-paste and hyperlinking features.

The situation is different for audiovisual documents. In current AVIS, results often consist in collections of video documents (as in the Internet Movie Archive [8]), or of video fragments (as in CNN digital archive [24]). Their main use is often limited to a simple visualisation. More advanced uses such as editing in order to produce a new video need video editing tools that are not yet integrated with AVIS, although research is also active in this area [4, 22]. HyperHitchcock [19] proposes a means to create hypervideos featuring hyperlinks between video fragments. The creation does not however use an underlying retrieval engine, but is manually defined through an authoring software. The VideoZapper [3] approach allows to dynamically edit a video, based on previous user's experience. The output of this

system is then a new video, with selected fragments from a number of broadcasted sources.

2.4. Importance of usage

The studies of information retrieval systems often overlook the importance of the usage of the retrieved information. It should be stressed that the intended usage determines the descriptors that are used for the information retrieval, and even their interpretation. Moreover, for audiovisual documents, other uses than simple visualisation depend on the descriptors: they will often become indexes to specify links in the video, captions put on the video, etc. Video descriptors are thus determined by, and essential to, retrieval as well as visualisation in AVIS.

We claim that the main issue in AVIS is not video retrieval but video usage, which determines both retrieval and processing. Feedback from user studies, as done in [20], should guide the development of new systems. We must evolve from simple video retrieval using descriptors to building new documents from video and descriptors. Other authors share the same vision: Nack [12] also underlines the need for integrated video systems, from shooting to broadcasting and archival.

2.5. Proposal

As indexing and retrieval cannot be done without knowing the needs and intended usage, research in multimedia information retrieval should be closely interacting with the multimedia information visualisation domain, and gather information through user studies [20]. We need to build upon both video retrieval and hypermedia authoring in order to produce effective AVIS that will be more widely used, therefore providing more feedback on the needs and usefulness of these systems. As Frank Nack said, "the current goal of multimedia research is to make multimedia information pervasively accessible and useable" [11]. This extended use needs appropriate tools and also a large video corpus.

Moreover, innovative uses of video information can develop when given opportunities, just as what happened with text documents: hypertext brought for instance a new way of conceiving and using documents, which is now exploited by media, scientists, artists, etc. The notions of hyperlinks, of fragments addressing and reuse, should be employed in order to promote new uses of video material.

To address these issues, we propose to use the notion of hypervideo, which we will define more precisely in the next section. The following section will describe how the Advene project allows us to put hypervideos in practice.

3. Hypervideos

We first define an Annotated Audiovisual Document (AAD) as an audiovisual document augmented by an annotation structure, then a view as a way to visualise part of the AAD. An hypervideo is a specific view, using the annotation structure and giving access to the stream temporality.

Annotated Audiovisual Document. We call Annotated Audiovisual Document (AAD) an audiovisual document augmented with an annotation structure S_A with spatiotemporal relations to the document (i.e. some elements from the annotation structure are linked to spatiotemporal fragments of the document). The annotation structure that enriches/augments the document is composed of annotations a_i linked to spatiotemporal fragments f_i, and of a structure S defining relations between the annotations.

$$AAD = \{AVD, S_A\} = \{AVD, \{\{a_i, f_i\}, S\}\}$$

For instance (see figure 1), a 4-minutes long movie stored as an MPEG2 file can be annotated with a hierarchic structure that describes the logical scene/shot structure (a movie is made of scenes, each having a title and containing shots). Shots are temporally located in the document. The structure also holds a description of the different characters and their relationships (here, *Char1*, who is present on the screen during the third shot and part of the fourth, sends to *Char2* a letter, whose text is known and describes what happened during the first scene of the movie). The annotations a_i contain data about the shots and the characters, the associated fragments f_i are temporal fragments and the structure S contains the remaining data (movie, scenes, relationships, text of the letter, etc).

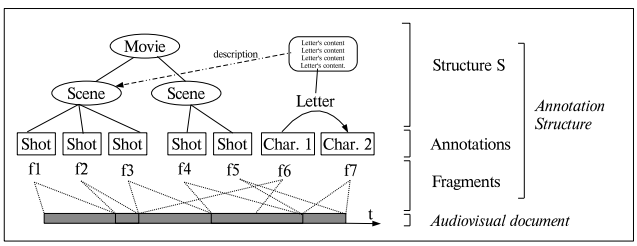

Figure 1: Example of Annotated Audiovisual Document.

Note that the definition of the annotation structure remains very generic on purpose: it holds all the information linked to fragments, as well as their interrelations. The nature of the information itself is not specified and can feature a temporal dimension, like an audio annotation for instance, or not.

View of an AAD. We call view of an Annotated Audiovisual Document any "way of visualising" it. A view is built 1/ from information taken from the document AVD, 2/ from the annotation structure S_A, and 3/ from its visualisation information. The visualisation information is always in part hard-coded in the visualisation tools/programs, and can also be specified by declarative means such as stylesheets.

Two criteria can be used to analyse views: first, the existence of the possibility to offer an access to the AAD temporal stream; second, the usage of informations coming from either the annotation structure S_A or the audiovisual document AVD, or from both. Table 1 presents different possibilities illustrated with examples based on the AAD described in figure 1.

The examples proposed in table 1 illustrate various ways of view generation using information taken from the AAD, but also what we called the visualisation-specific information of the tools. For instance, a standard video player will contain all necessary information in its source code (type B-); a stylesheet can be enough to generate a table of contents from annotations stored in an XML file (type C-). However, a specific tool is needed to extract screenshots from the movie if the view needs them (type D-). More complex views combine visualisation of audiovisual information and display of information taken from the annotation structure. They require new tools, for instance video players with enhanced capabilities such as the display of various kind of information (textual, graphical) on the video screen, original navigation capabilities, etc. They are the views that we call *hypervideos*.

Hypervideo. We call hypervideo any view of an AAD that on the one hand uses the annotation structure S_A *and* the audiovisual document AVD, and on the other hand gives access to the temporal stream of the AVD.

For instance, elaborating from the previous example (MPEG2 movie plus hierarchic annotation structure), it is possible to visualise the stream in a standard video player, with classic navigation capabilities (play, pause, fast forward, etc). This view can be called "simple navigation" (type B-). It uses the information from the audiovisual stream only, as well as the information built in the video player code (which we will call visualisation-specific information). Another view may use the annotation structure to present it in a web browser, each shot being represented by a key frame (for instance the first frame of each shot). This view (type D-) uses the annotation structure, the audiovisual document (to extract pictures) and, as visualisation-specific information, a stylesheet that specifies how to present the information and a tool to extract keyframes from the document.

This view is however not a hypervideo, because it offers no access to the original document temporality. The addition of the possibility to click on each keyframe to play the corresponding video segment makes it a hypervideo (type E-). Another example consists in using the annotation structure to enhance the original document while it is playing, through an instrumented video player. For instance, it can be a view constantly dis-

	No access to AAD temporality	Access to AAD temporality
AVD	A- Display of a static summary of the movie, as a table of screenshots extracted from the movie every 10 seconds, automatically generated. Visualisation of the sound envelope of the movie.	B- Simple visualisation of the movie in a basic video player, with regular controls (play, pause, start, fast forward, etc).
S_A	C- Visualisation of the information structure movie/scenes/shots. Visualisation of the character's names and of the text of the letter.	*Not applicable*
AVD + S_A	D- Display of a hierarchical table of contents of the movie, where each shot is represented by its first picture. Visualisation of the letter's content with 3 illustrating screenshots taken from the 3 shots of the first scene (see the *description* link).	E- Visualisation of a hypertext table of contents of the movie with links to the corresponding fragment of the video. Visualisation of the movie, captioned with the current shot number, with the possibility to navigate between shots. *On-the-fly* edition of the movie featuring only the shots where characters are present (shots 3, 4, 7).

Table 1: Five types of views for an Annotated Audiovisual Document

playing over the video the title of the current sequence as well as the number of the shot, and a link to the next and previous sequences.

The notion of hypervideo that we propose here is compatible with and extends the definition given in other works [17, 4]. It tries to set a unifying framework to build hypervideos, but also to analyse existing ones. There are indeed a number of existing and sometimes widely used hypervideos. The most well-known example is DVD menus, which provide a view on the video chapters with the possibility to directly play the interesting part of the movie. They can be very simple –chapter titles linked to the movie's chapters– or more elaborated: they can display in parallel reduced-size excerpts of each chapter, in order to provide more information about them. Moreover, subtitles are also a supplemental information that can be displayed on the video, thus providing a new view of the movie. Analysing DVDs as hypervideos explicits the metadata constituted by chapter definitions and subtitles, and suggests the idea of their reuse.

Considering existing media as hypervideos offers a unifying analysis framework that allows the comparison of different experiments and makes possible the transmission of experience from one to another.

For instance, the Hyperfilm [16] project proposes to enrich a video with links to either other parts of the same video or to external web documents. It does not however consider that the metadata embedded inside its documents may be exploited in other ways. Hypercafe [17] also explores hyperlinks in videos and tries to express the specificities induced by the spatial and temporal nature of documents. Both systems are however the result of an interactive edit process, and not directly built from requests results. The previously cited examples of HyperHitchcock [19] or video skimming [6] qualify as hypervideos according to our definition: they provide a hypertext visualisation of a video, allowing to access any identified fragment of the video. They are automatically generated, but do not provide the same interactivity when playing the video.

Thus some experience of hypervideo creation and interaction is already available, but its link to AVIS is not always explicit. The use of descriptors as indexes for new documents could be generalised, enriching the manual edited hypervideos with additional data. The bridge between information retrieval and hypervideos is not yet achieved. That is the reason why we propose in the Advene project to build upon the previous experience in hypervideos and to take advantage of structured metadata of audiovisual documents in order to integrate it.

4. Advene

The Advene project[2] aims at developing an open-source framework for hypervideo engineering, that allows to 1/ annotate audiovisual documents, i.e. to associate information to specific fragments of a video; 2/ provide augmented visualisations of the video that use the annotation structure; 3/ exchange the annotations and their associated visualisation modes independently from the original video, as documentary units called *packages*. It should meet the four aspects of document access presented in the introduction: visualise audiovisual documents, search them, hyperlink them and manipulate them through dynamic selection and edit.

The basic principles of Advene are 1/ to accommodate various needs in its annotation structure as well as in its visualisation means[3]; 2/ to facilitate the development of video usage at a simple user level and foster new usages; 3/ to use a video corpus that is accessible and interesting: movies edited as DVDs.

The main application domains of the project directly depend on the chosen audiovisual corpus, namely movies available on DVDs. This makes it useful for language teachers using the movies as pedagogical supports, moviegoers wishing to exchange movies analysis and discourses, etc. The genericity of the Advene model makes it usable in other domains such as humanities (using video support).

The choice of dealing with DVD movies is preponderant in the Advene project, even though it imposes strong constraints due to the technical capabilities of DVDs. One of the roots of the success of the WWW lies in its widespread adoption by a great number of users, which leads to the development of many tools to dig a vast mainly text-based corpus. In order to observe similar developments in the audiovisual domain, we need to have such a large user base and corpus. But the rights management of audiovisual data is stronger and more strictly enforced. One of the solutions is to build an accessible corpus [20, 8] from either public documents or self-made documents. It is however limited in its scope and cannot foster the interest of a large community. DVDs on the other hand constitute a large video corpus, disseminated worldwide, maybe with some variations depending on the location but globally homogeneous. In addition, it is available to a large user base. The criteria for a widespread usage are thus met: a large user base accessing a large, common corpus. The

2 http://liris.cnrs.fr/advene/

3 "[there is a] wide variety of needs from users, and so the interface should have some kind of flexibility (like the agile views)." (James Turner [23]).

condition is that every person owns a copy of the DVD she is interested in. Its enhanced use, through annotation and visualisation tools, is then an authorised, private use of the media.

The development of new usages of audiovisual material depends on the ability of the system to address the needs of a variety of persons. Hence, as we have seen in section 2, two major elements have been taken into account: the flexibility in the design of descriptors used to enrich the audiovisual document and the versatility of the proposed visualisation means.

4.1. The Advene model

Advene is based upon a model, with a focus on simplicity over completeness, that defines the available elements in the Advene framework. The model only specifies how to link a piece of data to a fragment of an audiovisual document. The handling of the data structure is then handled by specific plugins for each type of data.

The basic element in the Advene model is the annotation. An *annotation* is simply a piece of data linked to a part of a document called a *fragment*. Annotations can be put in relation with one another by means of n-ary *relations*, which can also possess a content describing them.

Annotations and relations are not unstructured: their content type and their relation structure are identified and constrained by *annotation types* and *relation types*. An annotation type possesses a name and defines a content-type for its annotations in the form of a MIME type (`text/plain`, `text/xml`, `audio/*`, etc.). If the type is `text/xml`, it can be more precisely constrained by a structure description (*e.g.* DTD, XML Schema, ...). A relation type also possesses a name and defines a content-type for its instances. In addition, it specifies the number of participating annotations and their respective types.

Annotation types and relation types can be seen as elementary elements of a specific analysis. Therefore, they can be grouped in a unit called a *schema*. A schema represents a certain point of view on the analysed document, and may be reused with other movies.

One of the most common examples is the Shot/Sequence analysis. It can be implemented with a schema called *Cutting*, which defines the following annotation types: *Movie*, *Sequence* and *Shot*. It also defines a *isPartOf* relation type that links an instance of *Sequence* to the *Shot*s it contains. The *Movie* type defines a structured content-type that will hold information about the movie: director, release date, reference URL. The *Sequence* type also defines a structure content that will hold the sequence number and the sequence title. Eventually, the *Shot* type holds the following information: the shot number, and the description of the shot. Figure 2 sums up this schema.

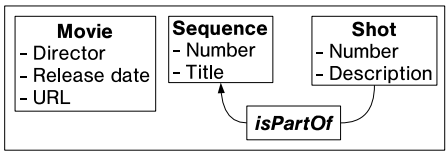

Figure 2: The *Cutting* schema

Using this schema, it is possible to annotate a movie and use the information in various ways that will be described in the following sections.

As we previously argued, an important aspect is to provide the user with facilities to define its own visualisation means. The Advene model integrates a notion of *view* that defines a way to visualise elements from the model. The model does not impose a specific mechanism but the prototype described in section 4.2 defines for the moment two different types of user-definable views, in addition to the GUI elements: a static one, using a standard web browser to visualise documents generated from the hypervideo data, and a dynamic one able to react to specific events during the video play. They will be described further.

The field of study of the Advene project being AVIS, the Advene model also integrates the notion of *query*, that allows to select elements from the model. The model only specifies that queries return a set of elements from the model, in order to accomodate multiple query models. Details are left to the implementation, which can propose multiple ways to represent and execute queries. The current implementation proposes a simple filtering system, that selects from a set the elements matching a condition. This method covers the basic needs of the current experiments. But this issue is still a work in progress.

All these elements (views, queries, schemas, annotation- and relation-types, annotations and relations) form a consistent set: views and queries depend on the schemas, schemas are defined to achieve a specific goal. The elements are gathered in a single representation unit called *package*. An Advene package thus contains the description of a hypervideo: the schemas defining the annotation structure, the annotation structure itself, queries to manipulate it and views to visualise it. A package can be shared by users in order to be simply visualised or to be edited.

As some schemas –and their associated views– may be reused (*Cutting* is very generic for instance), it is possible for the creator of a new package to reference external packages so that she does not have to redefine them. Such a reference is called an *import*. The intended use of this functionality is to make it possible to have a repository of common, reusable schemas that would allow people (moviegoers, teachers, students, etc) to share their analyses of movies.

4.2. Visualisation

One of the main contributions of Advene is to integrate the issue of visualisation in the AVIS. The Advene model only specifies that a view has a content-type and generates data from the annotated document. In the Advene prototype, two types of views that users can specify themselves have been implemented, in addition to the programmed, ad-hoc GUI views (such as timeline, etc). They provide a base for the generation of hypervideos. We quickly describe them here, and more details can be found in [1]. Figure 3 illustrates the exploitation of the different types of views: in the upper left-hand corner, a web browser displays a static, User-Time Based view generated from the package's data. The lower right-hand corner presents a dynamic, Stream-Time Based View, captioning the video with the content of selected annotations. In the lower left-hand corner is an ad-hoc view, presenting the different annotations on a timeline.

Ad-hoc views Ad-hoc views are GUI interfaces to the data. They cannot fully qualify as user-definable views, as they are defined by a source code that only a programmer can modify. However, they can be parameterized to offer some variations in the display of the data. For instance, the transcription view displays annotations of the same type by concatenating their content in a text window. During the movie play, the currently active annotation is highlighted. This view proposes a number of parameters (the separator used to concatenate the annotation's contents, a toggle indicating wether to display timestamps or not, etc).

User-Time Based Visualisation. The first kind of user-definable view, called *User-Time Based Visualisation* (noted UTBV), may be seen as a static view. It is in fact the definition of a hypertext document, whose temporality is the one of the user visualising it.

The User-Time Based Visualisation deals with presenting the available data (from the package and from the audiovisual document) as hypertext documents. The interaction with the user is done through a standard web browser, that connects to a simple web server integrated in the Advene application. The natural support for it is (X)HTML and more generally XML, which can be displayed by any web browser and can easily be generated. We have chosen a technology issued from the Zope platform [9]: a template language called TAL (*Template Attribute Language*) coupled with an expression syntax for data access called TALES (*Template Attribute Language Expression Syntax*). The couple TAL/TALES presents a number of interesting properties: simplicity, availability of useful tools and libraries, validity of the XML templates and model abstraction.

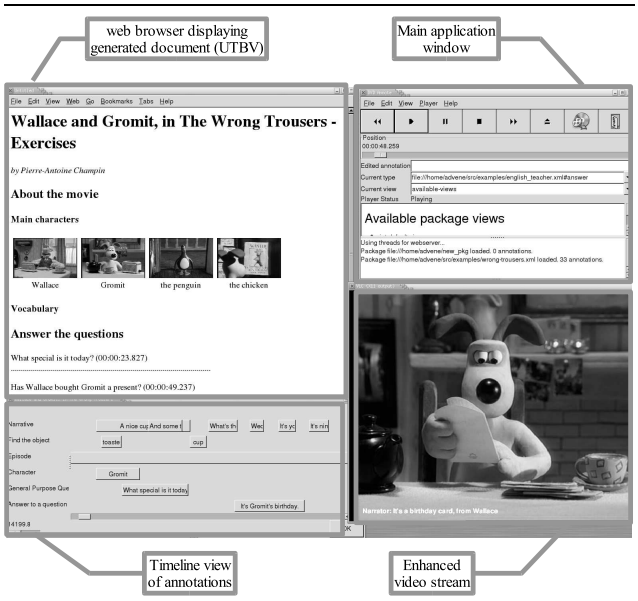

Figure 3: Multiple views of a single Advene package used in a language teaching context

Its simplicity mainly stems from its template-based nature: a TAL template is a valid XML document where some tags feature specific attributes. It has been designed so that it does not interfere with existing XML tools, especially WYSIWYG editors. The user only has to design a template document and afterwards insert simple processing instructions such as text substitution or basic loops.

The model abstraction is provided by the TALES component, which references elements in a data model using a path-like syntax which should be easily understandable by users. Moreover, the exposed structure represents the model itself (as seen through the library) rather than its XML representation. This makes it both more flexible and more intuitive.

Using the *Cutting* schema presented in figure 2, it is possible to easily generate a shot-by-shot table of contents of the movie with links to the corresponding fragments of the video. Figure 4 presents the ac-

tual template code used. The application of this template to data conforming to the *Cutting* schema generates a HTML table, which can be displayed in a standard web browser, containing the following information for each shot: shot number, shot duration, shot description with a link to the corresponding video fragment. The attributes defined in the `tal:` namespace are TAL processing instructions whose path-like values are TALES expressions, that allow to designate elements from the Advene model.

```
<table>
<tr tal:repeat="a here/annotationTypes/shot/annotations">
<td tal:content="a/content/parsed/number">Shot number</td>
<td tal:content="a/fragment/formatted/duration">Duration</td>
<td><a tal:attributes="href a/fragment/mediaurl"
       tal:content="a/content/parsed/description">
       Shot Description</a></td>
</tr></table>
```

Figure 4: Template for the table of contents

In the current Advene implementation, the HTML code of the templates must be hand-edited, or copy/pasted from existing examples. A WYSIWYG editor is currently being integrated into the application in order to make the design of simple templates more accessible.

Stream-Time Based Visualisation. In the second kind of user-definable view, *Stream-Time Based Visualisation* (noted STBV), the temporality of the resulting document is more related to the time of the audiovisual document than to the time of the user. It can be seen as a video augmented with additional capabilities. Two main approaches can be used to deal with dynamic presentation of content: scheduled or event-based [18]. SMIL for instance uses a hybrid approach, mixing determinate timings (scheduled) with undeterminate ones (event-based).

In the Advene prototype, we based the implementation on an event-based model, using the Event-Condition-Action (ECA) paradigm. The ECA model [14] is used in many applications such as databases or to define the filtering capabilities of mail client software.

Our use of ECA-rules is not meant as a full-fledged composition language, but as a simple means to achieve goals expressed by the users, using an understandable formalism. With it, we can quickly imagine and design new visualisation modalities for hypervideos. For instance, using both annotations and relations, it is possible to generate on the fly an alternate cut for a movie: annotations define the sequences, and relations define their order. Our first experiments with English-language teachers showed us that a common functionality is to pause at the end of a sequence and possibly loop over it. Using the adequate rules, we have been able to quickly implement this behavior.

A STBV is defined by a set of rules. Each rule can be triggered by the occurrence of an event, generated by the application and related to the activation of annotations during the movie play or the changing state of the player (pause, play, etc). When activated, the rule checks that some user-specified conditions are met, and executes actions accordingly. For instance, using the example presented in section 4.1, it is possible to design a view that allows to navigate between shots. Figure 5 presents the GUI that allows the user to define the corresponding rule: when the beginning of an annotation occurs, and if the type of the annotation is *Shot*, then caption the annotation with the shot number (extracted from the annotation's content) and display a popup dialog proposing to the user to skip to the next shot.

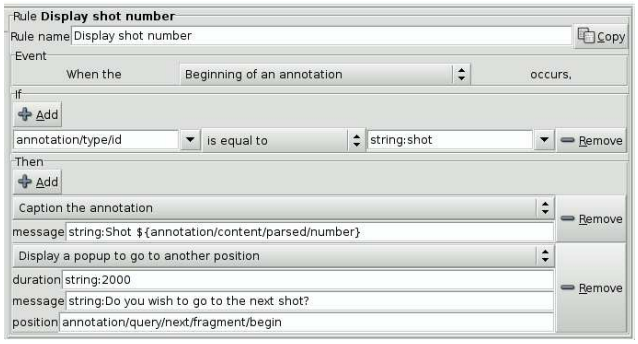

Figure 5: Example of Stream-Time Based Visualisation rule

No effort is made by the application to ensure the global validity of a set of rules, except for basic safeguards. A great deal of flexibility is given to the user, which may allow the definition of conflicting rules.

To implement such dynamic views, the Advene application integrates an enhanced video-player, controlled through a CORBA bus, that offers standard and not-so-standard capabilities: navigation in the movie stream, screenshot capture, text captioning, display of SVG graphics on the screen, etc. These capabilities are then exposed to the user through a number of possible actions. The currently implemented actions allow to:

- control the standard functionalities of the player: start/stop the player, go to a given position, modify the volume, etc.

- control the extended functionalities of the player: take a screenshot, display a text caption, display SVG graphics over the video.
- interact with the user: log messages, display dialog popups with video navigation options, etc
- control the application itself: activate another dynamic view.

The infrastructure of Advene makes it easy to propose and implement new types of actions, thus offering an excellent playground for testing new hypervideo interaction modalities.

4.3. Discussion

The model proposed in Advene aims at investigating some of the issues related to the management of video in Audiovisual Information Systems, through the notion of hypervideo. Advene proposes a simple but flexible annotation scheme, that should accomodate the needs of various categories of users. Moreover, visualisation of data is integrated in the model. More importantly, users have the ability to define their own ways of visualising the data, either by defining HTML templates or by specifying dynamic rules.

Our first experiments with the Advene prototype gave rise to some interesting issues related to hypervideo management, from information retrieval to information visualisation. In the context of AVIS, a query is applied on an Annotated Audiovisual Document, represented as a package accompanying the audiovisual document. The status of the Advene package depends on its use. Holding annotations, queries and visualisation definitions, it can be seen as a document in itself. It can also be seen as a document generator as each hypervideo generated from a package is a document.

As a document, it is meant to be shared and possibly modified. It is possible to create a repository holding Advene packages so that people interested in the same analysis of DVD movies can share their work, or use the same base (for instance, scene/shot packages of movies are of general interest). In such a repository, queries on the package's database would result in either a collection of packages, or a new package importing data from the matching ones. A query result then becomes a new document. It can thus be reused and shared between users, and possibly augmented, bringing new possibilities into the management of video databases.

As a document generator, it features multiples views that can be extended to fit the needs of the different users, reusing the same data or adding more data. This possibility to adapt the visualisation and the data to the various needs of the users is essential to propose a viable system. The generated hypervideos are a representation of the query results. Their non-linear nature, thanks to the hyperlinks, makes them objects distinct from standard audiovisual documents.

Visualisation of hypervideos puts forward a number of issues, either technical or conceptual. On the technical side, the need of adequate video software with extended capabilities, as identified in section 3, may prevent a large use of hypervideo software. The Advene project brings a contribution by providing open-source software built upon a cross-platform open-source video player [5]. The same principle is applied for instance by the authors of the Annodex technology [15], who propose another approach for annotating audiovisual documents where the audiovisual document and the annotation structure are merged in a unique stream. Moreover, annotating a video with audiovisual content raises the question of mixing multiple audiovisual streams. More conceptual issues deal with the relevance of video material in a multimedia context, how it is perceived by the user, what new uses can be brought by hypervideo. These questions are also in the current and future scope of investigation of the Advene project.

5. Conclusion

In the context of AudioVisual Information Systems, we have identified the different phases that occur during their exploitation and underlined that one of their shortcomings lies in the low level of integration between the different phases. More precisely, we have argued that usage determines the retrieval phase as well as the exploitation phase. We have then defined the notion of hypervideo that we propose as a unifying framework to analyse and build new interaction modalities with enriched video information. We have eventually presented the Advene project that explicitly implements the notion of hypervideo and is meant to be accessible to a wide audience.

Indeed, the Advene model does not impose a specific indexing scheme, but rather proposes a flexible model able to hold various types of information, thus accommodating the various needs of the users. Advene aims at providing a platform for exploring new uses of video by allowing the users to specify their own views and data model for audiovisual information. Using DVDs, it offers access to a large corpus and covers the various aspects needed for audiovisual document exploitation: visualising, searching, linking/referring and editing.

While hypervideos exist, they are not necessarily meant as such. They are often intended as a visualisation means, without taking into account the addi-

tional information stored in its structure. By defining hypervideos as views upon an Annotated Audiovisual Document, we want to emphasise the importance of the additional data in the processing of video documents. Hypervideos are then more than edited versions of videos with supplemental hyperlinks: they become complex documents featuring multiple and extensible visualisation means. Moreover, in the context of AVIS, hypervideos can be generated by queries, as the display of the query results (selection phase), or as their intended exploitation (exploitation phase), using both the audiovisual material and the descriptors used for the search. In other terms, search descriptors are used not only to search documents but also to *build* them[4]; conversely, visualisation descriptors – i.e. descriptors primarily used for hypervideo document rendering– are also used as indexes.

We think that using the concept of hypervideos offers a unifying framework to analyse and extend AVIS. It stresses the importance of annotation data in the process of video retrieval and results visualisation and exploitation. The Advene project allowed us to experiment with the hypervideo concept and to validate its applicability. We are currently experimenting with different categories of users in order to validate the generic applicability of hypervideos as well as their usability by non-expert users.

References

[1] O. Aubert, P.-A. Champin, and Y. Prié. The advene model for hypervideo document engineering. Research Report RR-2004022, LIRIS, 2004. 19 pages.

[2] G. Auffret and Y. Prié. Managing Full-indexed Audiovisual Documents: a New Perspective for the Humanities. *Computer and the Humanities, special issue on Digital Images*, 33(4):319–344, 1999.

[3] M. Boavida, S. Cabaço, and N. Correia. Videozapper: A system for delivering personalized video content. *Multimedia Tools and Applications Journal*, 2004.

[4] T. Chambel and N. Guimaraes. Context perception in video-based hypermedia spaces. In *Proceedings of the thirteenth conference on Hypertext and hypermedia*, pages 85–94. ACM Press, 2002.

[5] H. Fallon, A. de Lattre, J. Bilien, A. Daoud, M. Gautier, and C. Stenac. *VLC User Guide*. VideoLAN Project, 2003.

[6] J. Graham, B. Erol, J. Hull, and D.-S. Lee. The video paper multimedia playback system. In *Proceedings of the eleventh ACM international conference on Multimedia*, november 2003.

4 As a comparison, for text documents in the semantic web, indexes are used to search data as well as to construct result documents.

[7] A. Hauptmann, R. Baron, W. Lin, M. Chen, M. Derthick, M. C. R. Jin, and R. Yan. Video classification and retrieval with the informedia digital video library system. 2002.

[8] Internet movie archive, 2004. http://www.archive.org/movies/.

[9] A. Latteier and M. Pelletier. *Zope Book*. http://www.zope.org/, 2003.

[10] X. Mu and G. Marchionini. Enriched video semantic metadata: Authorization, integration, and presentation. In *ASIST 2003 Annual Meeting*, 2003.

[11] F. Nack. *Understanding Media Semantics*. ACM Multimedia Tutorial, 2003. http://www.acm.org/sigmm/mm2003/t11.shtml.

[12] F. Nack and W. Putz. Designing annotation before it's needed. In *Proceedings of the ninth ACM international conference on Multimedia*, pages 251–260, Ottawa, Canada, 2001.

[13] N. I. of Standards and Technology. Trec video retrieval evaluation, 2004. http://www-nlpir.nist.gov/projects/trecvid/.

[14] N. W. Paton, editor. *Active Rules in Database Systems*. Springer, 1999.

[15] S. Pfeiffer, C. Parker, and C. Schremmer. Annodex: a simple architecture to enable hyperlinking, search and retrieval of time–continuous data on the web. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 87–93, 2003.

[16] M. Pollone. Hyperfilm: video hyper-media production. Technical report, Hyperfilm project, 2001. http://www.hyperfilm.it/.

[17] N. N. Sawhney, D. Balcom, and I. E. Smith. HyperCafe: Narrative and Aesthetic Properties of Hypervideo. In *UK Conference on Hypertext*, pages 1–10, 1996.

[18] P. Schmitz. Unifying scheduled time models with interactive event-based timing. Technical report, Microsoft Research, 2000.

[19] F. Shipman, A. Girgensohn, and L. Wilcox. Image annotation and video summarization: Generation of interactive multi-level video summaries. In *Proceedings of the eleventh ACM international conference on Multimedia*, november 2003.

[20] L. Slaughter, G. Marchionini, and G. Geisler. Open video: A framework for a test collection. *Journal of Network and Computer Applications*, 3(23):219–245, 2000.

[21] A. Smeaton, H. Lee, and K. M. Donald. Experiences of creating four video library collections with the fischlar system. *Journal of Digital Libraries*, 2004.

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

[23] Symposium on understanding video, 2002. http://ils.unc.edu/idl/video_symposium02/.

[24] Virage video systems, 2004. http://www.virage.com/.