# On the Articulation of Document models and Ontological Models: Integration of Semantic Web Technology in an Annotation-based Hypervideo System

Pierre-Antoine Champin, Yannick Prié
Université de Lyon, Lyon, F-69003, France ;
Université Lyon 1, CNRS UMR5205,
LIRIS, Villeurbanne, F-69622, France
first-name.lastname@liris.cnrs.fr

April 6, 2007

**Abstract**

In this article, we question the notion of "semantic multimedia": to us, multimedia semantics resides in the models and the tools that are used to manipulate multimedia documents. Document models and document manipulation tools are therefore knowledge containers of such semantics, but there versatility and pragmatic efficiency is done at the expense of lacking formal semantics. On the other hand, ontological models offer formal and explicit semantics, but are not adapted to document processing and presentation. Our claim is that both types of models should be used for what they are best suited: document models for manipulating document structures, and ontological models for inference-based processing, explicit knowledge sharing and reuse, etc. An important issue is then to define and study precisely the notion of document and ontological model articulation. We propose a basic conceptual framework as a first step towards that goal, identifying modality, directionality and focus as relevant features of model articulations. To illustrate and study how model articulation prove useful in the field of semantic multimedia, we also present our own document model and prototype for video annotation and hypervideo design (Advene), and we consider three different cases of articulations. Each case study is illustrated with one or several examples working within the ontological extension of our prototype, providing consistency checking, reporting, advising and advanced queries.

# 1 Introduction

The context of this paper is related to the current development of novel applications for audiovisual (AV) documents manipulation and audiovisual information systems [3]. More precisely, it addresses multiple ways one can use and reuse AV documents within a hypermedia framework so as to create hypervideos [4, 7]. Hypervideos are useful in many application domains such as teaching (web lectures), human activity analysis (study of human interaction patterns), cinema (film critics), leisure (movie co-watching on the web), etc.

This need for advanced functionality in new or already existing applications often requires also new document models and tools to handle those [8]. The result are flourishing media semantics we have not been envisioning several years ago. Hence these models and tools put in evidence new "useful semantics" for AV documents. Our focus is therefore not directly to fill in the "semantic gap" between documents designed for human consumption and software agents handling them. We are more interested in studying 1/ the emergence of new practices, and the consequent uncovering of where AV documents semantics resides for users, as they perceive them in applications, and 2/ the modeling of AV documents associated to these practices [26].

In our long-standing research project on hypervideos, we adopt a document-based point of view related to hypermedia construction. In this approach AV semantics resides in the way users perceive the transformation of AV documents performed by document manipulation tools (authoring and presentation tools). Moreover, the semantics are described in models and schemas that allow practical applications to share and reuse those documents. AV semantics are therefore inscribed in the *document models* for AV document description and hypervideo building, and in the *annotation structure* they allow to add to AV documents.

However, because the AV semantics we capture within document models are not strictly defined, those models can prove weak when it comes to describing more formally defined semantics. In this article, our concern is then to study to what extend a certain category of models related to the Semantic web —that we call *ontological models*— can prove useful for *complementing* AV documents models and supporting hypervideo-related practices. For that purpose, we introduce the notion of *model articulation*, namely the articulation between audiovisual document models and ontological models.

The structure of the article is as follows. In section 2 we introduce the notion of document model, mainly focusing on audiovisual document models and discuss a number of such models proposed in the literature. We then present the Advene project for hypervideo study, whose document model is sufficiently simple and versatile for our prototype to be used in various applications. Section 3 is dedicated to the general issues of articulation between document and ontological models. In that discussion we present our proposal of a framework for studying different categories and modalities of model articulation and use it for positioning a number of already established works. We then present in section 4 how the Advene prototype has been enhanced in order to enable the articulation of its model with the OWL ontological model. In section 5 we study three types of articulations experimented in the Advene project, as well as working applicative examples related to film analysis. Finally we conclude and present future research directions.

## 2 Audiovisual document models

The notion of digital document can be considered from three main points of view [19]: the form, the sign and the medium. The form approach studies the document as a digital object with a structure that can be manipulated. The sign approach concerns the semantics expressed in the document, and the way it is expressed. The medium approach questions the document's status and usage in the context of social relationships.

We are here mainly concerned with the form and the sign approaches, reified into different structures, relying on different models. As a first approximation, document models are good candidates for the form, while ontological models seem well to describe the sign. In this section, we will focus on the former. The remainder of the paper will study the articulation between both kinds of models.

## 2.1 Document models

We define a *document model* as a model that describes and constrains concrete document structures through which documents can be manipulated and presented to a user. For instance HTML or MPEG standards are document models that describe, respectively, concrete web hypertext or AV documents.

Document manipulation tools are designed to store, present, search and transform document structures, according to document models. Most document models share a common feature: they have been designed pragmatically in order to sustain document-related needs (e.g. be able to present video in a hypermedia CDROM for MPEG1). Hence, there is always a tradeoff between what is present in the document model (what constitutes the document), and what is left to the document manipulation tools[1].

While document models provide minimal semantics, which constrains associated tools to some extent, those tools are often (and are expected to be) used in unforeseen ways, creating new uses and hence new semantics for document structures. For example, most word processing tools provide a function to highlight text. When people write a collaborative document, they often use that

[1]or, as a middle way, to other ancillary models. XSLT and CSS are example of such ancillary models that complement XML and HTML, but are distinct from them, and allow presentation information to be shared across documents.

function to leave comments to co-authors, even though a specific function is usually available, but less accessible. Hence we consider that the semantics of document models largely resides in the associated document manipulation tools and the practices they induce.

## 2.2 Audiovisual document models

Audiovisual document models are document models that concern AV documents. We can identify three kinds of models.

**Basic audiovisual document models.** The most obvious AV document models are basic models for describing AV documents as synchronized streams of images and sounds, such as MPEG or AVI. These models are related to tools for playing these documents, with basic control and navigation facilities related to their temporal character (e.g. changing playing rate or seeking a given timecode in the document).

**Annotation-based audiovisual document models.** However, we are most interested in AV document models that allow to describe videos and their non-basic uses, such as video navigation, precise AV information retrieval or personalized AV documents generation. For this, we need models that are capable of taking into account the notion of temporal annotation, i.e. data related to a (spatio-)temporal *fragment* of the AV stream. Hence the structure of such documents enriches the intrinsic temporality of basic AV models with the notion of AV fragments, and allows to take into account related information. Simple examples of such additional information are subtitles or chapters in a movie, which are available in DVDs or file formats such as Quicktime[2] or Matroska[3].

[2] `http://www.apple.com/fr/quicktime/`
[3] `ttp://www.matroska.org/`

The tools based on such models offer means for presenting or manipulating enriched AV documents, taking advantage of the annotation structure (e.g. present subtitles on the playing movie; use the chapters to build a table of content; generate a re-montage of the document related to the subtitles containing a particular word). Such novel kinds of documents are called *hypervideos*, for they extend standard video playing with additional uses.

Annotation-based AV models are more or less specialized, depending on their scope. For instance, the popular SRT format[4] for subtitles, read by most video players, allows to attach textual content to fragments of a video. The Annodex Project model (CMML [20]) allows to attach to an AV stream annotations that contain HTML, to be displayed as such in dedicated applications (for instance the annodex Firefox plugin). Web-applications like Mojiti[5] and BubblePLY[6] allow to enhance videos from external sources with elaborate captions and hyperlinks. The MPEG7 model [22] proposes many kinds of standard annotation types (mostly focusing on low-level descriptors automatically extractable from the audiovisual document) related to the many applications that where considered at design time (e.g. surveillance, retrieval, tailored presentation...).

To sum up, annotation-based AV models offer the possibility to define 1/ fragments of the AV document, 2/ annotations as data containers and 3/ relations between annotations. Annotations and relations compose the *annotation structure* of an AV document. The information of the annotation structure is used together with the information of the AV stream to build *hypervideos* as audiovisual-based hypermedia views [4], see figure 1.

Annotation-based document tools offer several services. Some are related to selection in the annotation structure (queries on the annotations, the relations, the AV fragments) and presentation of the hypervideo (views). Other allow

[4] `http://en.wikipedia.org/wiki/SubRip`
[5] `http://mojiti.com/`
[6] `http://www.bubbleply.com/`

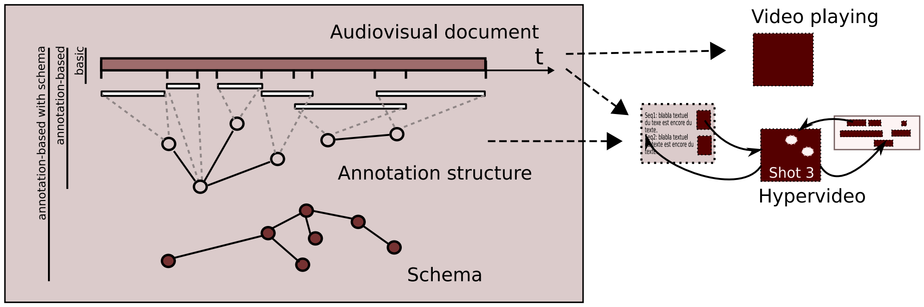


Figure 1: Left: AV document models (simple, annotation-based, annotation-based with schema). Right: Simple models lead to basic video playing, while annotation-based models allow to build hypervideos, i.e. hypermedia documents based on video documents.

to compute temporal relations between fragments; for instance, checking that subtitle fragments do not overlap, or that the whole length of the video is covered by contiguous chapter fragments.

**Annotation-based document models with schemas.** Most annotation-based document models are linked to precise practices and applications, that constrain the annotation structure in a unique way. They do not feature any notion of customized integrity constraint. Hence, some annotation-based document models go one step further by adding the notion of *schemas*, that allow authors to add *explicit* constraints to the annotation structure.

To some extent, the DDL (Description Definition Language) of MPEG7 offers means to define schemas: additional annotation types with a customized structure. Of course, those additional types require ad-hoc tools in order to be exploited. Our own model, the Advene model (see next section), defines a schema as a set of annotation types and relation types corresponding to a given usage, but also feature a notion of *view* to specify how those new types are to be presented.

## 2.3 Advene model

The Advene project aims at providing tools to build, exchange, enhance and customize critical analyses about video documents, e.g. hypervideo film analyses that are shared in a community. Analyses are built upon annotations linked to the fragments of the video. The Advene prototype[7] provides means to create and modify annotations, as well as to specify how they should be rendered in meaningful ways. Instead of exchanging the sole final form of an analysis, the Advene project makes it possible to rather exchange annotations and the specification of their visualisation, thus allowing end-users to customize data and visualisations in order to fit their needs. For this, it keeps metadata separated from the audiovisual document, so that they can evolve and be exchanged independently from the movie, in order to eschew the issues of copyright.

This section presents the Advene model as an annotation-based AV audiovisual model with schemas: annotation structures are constrained by explicit schemas, that can be created by users, shared and reused by others.

**Annotation structure** In the Advene model, described more precisely in [4], we define an *Annotated Audiovisual Document* (AAD) as an audiovisual document augmented with an annotation structure. The annotation structure consists of *annotations* and *relations* between annotations. Both may contain data. Annotations are linked through a temporal fragment to a specific portion of the AV document. The structure of data contained in the annotations and relations is not specified by the model: it can be any type of data (simple text, structured information, audio documents, office documents...).

This very generic model can be further structured: *annotation types* constrain the kind of content (with a MIME-type specification) held by annotations. Multiple annotation types can be used to describe a number of analysis facets.

[7]Open source multi-platform prototype available from `http://liris.cnrs.fr/advene/`

*Relation types* constrain the types of annotations that can be linked by a relation, as well as an optional content MIME type for relations.

As annotation types and relation types define a certain point of view in the document analysis, they are grouped as meaningful sets called *schemas*. An Advene schema thus defines annotation types and relation types that form together an analysis framework.

Let us illustrate this through a simple example. Suppose we want to analyze a movie along its structure and its story. For this we can define two advene schemas. The *Structure* schema will define two annotation types (*Sequence* and *Shot*) as well as their relation type (*contains*, expressing that a movie sequence is composed of shots). The *Story* schema, that builds upon the first one defines two other annotation types (*Place*, *Character*) and two additional relation types relating Shots to Places and Characters, respectively. Figure 2 sums up these schemas, and present an example of complying annotation structure. The use of the *Story* schema consists in: 1/ annotating a representative fragment of the movie that depicts each Place and Character, then 2/ relating those annotations to every Shot situated in the corresponding place, and involving the corresponding character.

One of the design goals of Advene is to allow users to specify themselves how they want the annotations to be rendered. For this, the Advene model defines two notions: queries and view. *Queries* define a way to select elements from the annotation structure. *Views* define a way to display information from the audiovisual document and the annotations.

We identified and proposed three main types of views in the Advene prototype (see section 4.1 for some technical details). *Static views* are standard hypertext documents (texts, images, links), for instance a summary of the film sequences that gives access to the video at the right timecodes. *Dynamic views*

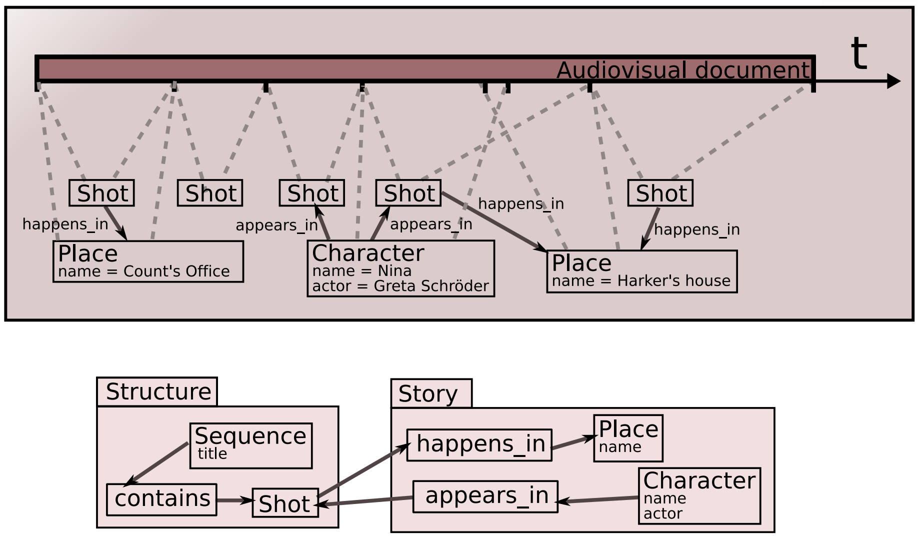


Figure 2: Top: an AV document is annotated with an annotation structure composed of 8 annotations and 5 relations. Bottom: annotation types and relations types are defined in two Advene schemas.

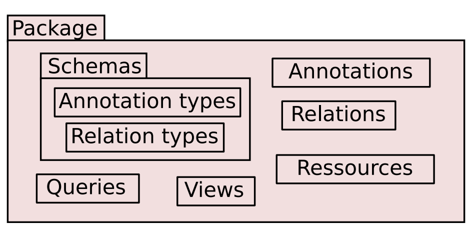


Figure 3: Overview of the Advene model.

are enhanced playings of the video, for example, each time a new sequence begins, its title is displayed on the stream, and the user has the possibility to navigate directly to one of the shots it contains. On the contrary to static and dynamic views that can be designed by the user, *ad-hoc views* are hard-coded GUI components of the application, for example, a timeline view presents all the annotations of a movie, offers zoom, playing, editing facilities, etc.

A hypervideo in Advene is a set of views that on the one hand use information from both the audiovisual document *and* the annotation structure, and on the other hand gives access to the temporality of the audiovisual document [4].

Figure 3 gives an overview of the different elements of the Advene model. They are stored in documents called *packages*, that hold all relevant information (schemas, annotations, queries, views and resources[8]) allowing to exchange, modify and visualise the metadata associated to an audiovisual document[9].

**To summarize our reasoning so far.** A document model is a pragmatically designed model, the semantics of which mainly residing in the tools that use the actual document structures obeying that model. In the audiovisual domain, beyond simple stream description, annotation-based AV document models feature the notion of annotation (and related notions: fragment, annotation content, annotation structure) that allows to build hypervideos. Some models are provided with schemas that constrain how the annotation structure is build.

[8]A resource is any file that may be of use within the package: stylesheet, PDF document, OWL file, etc.

[9]It can be compared to a Microsoft Access file that contains both a database and the queries and forms allowing to use it.

# 3 Articulating document models and ontological models

We now discuss ontological models and the notion of model articulation.

## 3.1 What are ontological models

Ontologies are defined by Gruber [14] as explicit specifications of a conceptualization, i.e. formal descriptions of the elements structuring a domain of interest. Description Logics (DL), of which the ontology language for the web OWL [11] is derived, are a popular formalism for representing ontologies. That family of languages focuses on a strictly defined semantics, in order to allow software tools handling ontological structures to make *inferences* about what is formally entailed by a given structure.

More precisely, efforts in the DL field have been motivated by the fact that previous languages (frame languages, semantic networks), having more loosely defined semantics, showed discrepancies among the inferences performed by different implementations [5]. Knowledge sharing was hindered by those discrepancies, since inference is the primary task intended for ontological structures. The precise semantics of DL, and the provably sound and complete inference algorithms associated with them, allow ontological structures to be shared with the certainty that they will draw to the same consequences for any partner. A side effect is that the shared meaning of ontological structures is purposely limited to what the formal semantics allows to infer. There is no room for ambiguity or connotations.

Another feature of DL-based ontological models is their ability to reason under the open-world assumption: constraints may not be explicitly satisfied, as long as they are not explicitly violated. Although this is sometimes considered as an irreconcilable difference with closed-world models, we consider on the contrary that it is an additional degree of flexibility. Indeed, nothing prevents a particular ontological structure to be artificially "closed", i.e. to be added negative assertions for everything that is not explicitly stated. But this is not required, and openness can be kept wherever deemed useful.

We therefore consider that there is a tension between the underlying approaches of document model and ontological models, beyond the classically stressed out difference between the syntactical and semantical level. We have argued that loose semantics is a *feature* of document models, allowing diverse (and sometimes unexpected) uses of documents. On the other hand, the ontological models insist on a strict semantics, allowing powerful inferencing, but requiring a more controlled (and often expert) use of ontological structures.

## 3.2 Model articulation

There is nevertheless an urge to merge the benefits of both worlds, bringing the ease of use of documents to ontological structures, and the power of inference to document structures. We are interested in articulating document models and ontological models in the field of AV documents.

We call a *model articulation* any process allowing to consider the same document along two different points of view. The benefit is that users wanting to manipulate the document may then choose between the two models, and associated tools, the most appropriate for the task at hand. The conceptual structure resulting from that articulation is represented in figure 4.

Note that model articulation is not new to the field of document models. Import/export filters are a common, though very basic, form of articulation between two document models. XSL-T [9] specifies a way for XML documents to declare a transformation method (stylesheet) into a human readable structure[10].

[10] Actually to *any* kind of document structure, though its primary focus is presentation

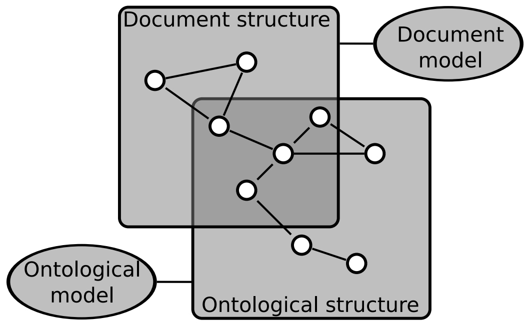


Figure 4: Articulated structures obeying two different models

More recently, the notion of microformat [17] has received much attention: additional document structures are embedded in HTML documents, usually by means of the versatile *class* attribute. Articulation with ontological models has also been considered: GRDDL [10] specifies a way for XML documents and schemas to declare transformation methods into RDF graphs, while RDFa [1] is a kind of microformat for RDF.

In the field of AV documents, most efforts on model articulations have been conducted between the MPEG7 document model and Semantic Web ontological models (RDFS and OWL) [15, 24, 16, 13]. Those works aim at translating MPEG7 structures into a language with a formal semantics. Their rationale is that such ontological languages are better fitted for machine processing.

**A conceptual framework for considering articulation.** In order to study and compare model articulation approaches, we propose to consider several dimensions along which those approach may vary. Table 1 gives an overview of how presented articulation approaches fit into that framework.

First, the articulation may be based on the *mapping* of existing structures into another model (XSL-T, mappings of MPEG7) while other tend to *embed* one model into extensibility "slots" of the other one (e.g. microformats using the *class* attribute of HTML). Note that this distinction is not always clear-cut: GRDDL examples are usually hybrid, using both the original document structure and excerpts of RDF added to the document, to produce an RDF graph.

Second, most efforts in articulating models have a strong *directionality*, meaning that they consider a source and a target model. This is obviously true of import/export filter, but even the recent works cited before consider only one-way transformations: e.g. HTML to structured data for microformats, XML documents to RDF graphs for GDRRL. A few approaches however, consider more feedback between the two models, thus tending towards two-ways articulations. An example is [24], where MPEG7 structures are generated from an ontology, in order for a archivist to feed that ontology with document manipulation tools, and then allow inferences to be performed in the corresponding ontological structure.

| Approach | Models | Modality | Directionality | Focus |
|---|---|---|---|---|
| Microformats | HTML / various | extend | 1-way | HTML |
| GRDDL | XML / RDF | hybrid | 1-way | RDF |
| [15] | MPEG7 / DAML+OIL | map | 1-way | RDFS |
| [13] | MPEG7 / OWL | map | 1-way | OWL |
| [24] | MPEG7 / OWL | map | 2-ways | both |

Table 1: Features of different model articulation approaches

Related to the notion of directionality is the notion of *focus*. Microformats claim to be designed for humans first and machines second, putting a strong emphasis on the source model (HTML) rather than the target models (structured data formats), aiming at easing the work of document authors. On the other hand, most works targeting ontological models put the stress on those models, with the goal of enhancing automated indexing and retrieval processes.

# 4 Ontological Advene

The basic framework we propose for considering model articulation types is quite general and applies to many models and documents types. However, our particular scope concerns audiovisual documents and our Advene document model. Before we study several cases of articulations in Advene, we need to present some details about the Advene prototype, its ontological extension, and the general ontology-based tasks it features.

## 4.1 Views and queries

The Advene model is fairly generic. Some decisions regarding the implementation of query or view languages had to be made in the prototype, regarding views and queries.

**Views in the Advene prototype.** As stated earlier, the prototype proposes three types of views: ad-hoc views (GUI views), static views (HTML templates) and dynamic views (set of rules allowing to dynamically modify the movie rendering).

Ad-hoc views are programmed views built in the GUI, that the user can configure. They feature standard views found in audiovisual software (time-line views, hierarchical data view, transcription view...).

Static views are XML (usually XHTML) templates that can be applied on the data. We are reusing the ZPT (Zope Page Templates) template system from the Zope platform [27]. This template system is oriented towards XML templates edition, using attributes in a dedicated namespace as processing instructions. Thanks to this attribute-based approach, both templates and result documents are valid XML documents, which allows us to process them with standard XML processing tools.

Dynamic views are able to dynamically change the way the movie is played, based on the annotations' content. Using a rule-based model similar to the filtering capabilities of e-mail software (Event-Condition-Action [18]), dynamic views allow the user to specify various actions to be executed when some events occur. The actions range from simple VCR-like functionality (pause, go to a position, stop...) to more elaborate video control (display captions – text or graphic – on the video, get a snapshot...), and also provide user-interaction facilities (information popups, navigation popups offering to go to another position...). The events are triggered by the annotation structure (annotation begin, annotation end...) or by user actions (player pause, player start...).

**Queries in the Advene prototype.** Queries offer a way to select elements from the annotation structure. A simple query implementation has been integrated in the prototype, using the same framework as the dynamic views: elements matching a given condition can be extracted from a given set of elements. This approach has proved flexible enough to accommodate various needs in our experimentations: selecting elements based on their contents, their temporal relationships (through Allen relations) or their relations.

## 4.2 Architecture of the Advene prototype

The open-source Advene prototype reuses standard software components: it embeds the versatile, open-source and cross-platform VLC video player [12], uses the ZPT template model from Zope, and uses a standard web browser to visualise the rendition of the ZPT templates. Figure 5 provides an overview of the prototype architecture (left-side of the figure).

The Advene prototype has been written in python, which proved an excellent choice for rapid development and experimentation, so as to foster the emmergence of new practices [8]. It provides a testbed for the development of new ideas in the field of multimedia annotation handling and visualisation. It is being used in ongoing collaborations with researchers in human interactions (who study video recordings) or movies study (e.g. Ciné Lab project[11]), as well as by individual researchers that use audiovisual material.

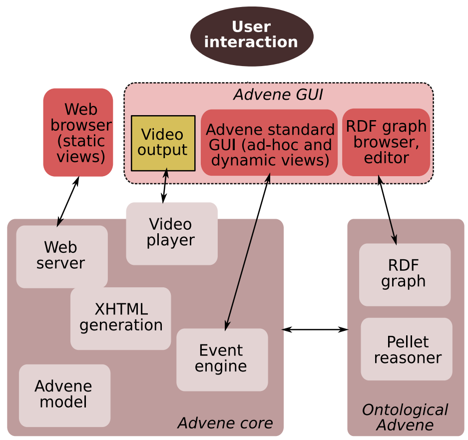


Figure 5: The Advene prototype architecture (left) and its ontological extension (right)

## 4.3 Ontological extension to Advene

Following our work on document model and ontological model articulation, and the basic conceptual framework we proposed in order to analyze it, we designed an "ontological extension" to the Advene prototype. This extension is meant to rapidly and conveniently study some of the articulations and related applications we proposed.

Figure 5 (right side) describes how it integrates in the application architecture: RDF graphs are handled in a dedicated component. This component communicates with an OWL reasoner (Pellet [23]) for inferential queries to the RDF graph. A part of the Advene GUI is extended so as to provide means to

[11]http://liris.cnrs.fr/advene/cinelab.html

browse and/or edit RDF graphs (for the moment being, we just implemented a tree-view of a graph).

## 4.4 Ontology-based tasks

A not exhaustive list of the tasks that can be sustained by that enhance prototype is given below (see [2] for more detail):

**Consistency checking.** Although the Advene prototype can check some basic integrity constraints, mapping its structures into OWL allows a schema designer to add finer constraints. Pellet can then check that those constraints are verified by testing the formal consistency of the resulting ontological structure. Users annotating a video can then check that they are correctly using a given schema; should the consistency checking fail, it is likely that the views associated to the schema would give unsatisfactory results.

**Reporting.** Integrity constraints may not always be implemented in a way that makes the resulting ontology inconsistent. Indeed, it may be more practical to assign the faulty elements to a specific class, and ask the inference engine all the instances of that class. This allows more accurate error detection: a schema designer may provide users with static views enumerating all faulty elements, and even distinguish between different levels or kinds of fault.

**Advising.** Under the open world assumption, missing elements can be reasoned about by the inference engine. This allows schema designer to define expected *annotation patterns*, and to assist the user in filling in those patterns by creating new annotations respecting the constraints that can be inferred by Pellet.

**Advanced querying.** The rich semantics of ontological structures allows to question them in various ways. The SPARQL language [21] is supported by Pellet to query the deductive closure of the RDF graph. The ontological extension of Advene uses this feature to provide an OWL-dedicated kind of queries.

# 5 Model articulation applied in Advene

The ontology-based tasks we just presented can be further refined considering specific articulations and applicative examples. We describe in this section practical articulations of the Advene document model with OWL ontological models. Some of these articulations have been tested while others are still under development. All aim at studying the emergence of multimedia semantics as a hybrid between the loose flexible semantics of document structures and the formal semantics of ontological structures. We will use as a running example the Advene package annotating Murnau's Nosferatu [12], that uses schemas similar to the *Structure* and *Story* schemas decribed before.

## 5.1 Mapping document structures

A first way to articulate the Advene model with ontological models is to *map* the document structure of an Advene package to an ontological one. In Advene, static views (producing any kind of XML document) are a straightforward means to achieve such a mapping.

The mapping can be more or less generic depending on the target ontological model. At one end of the spectrum, one can build an ontology of the Advene document model and map any package to an ontological structure conforming to that ontology[13]. At the other end of the spectrum, one can propose

[12] `http://liris.cnrs.fr/advene/examples.html`

[13] A package defining such a generic view can be downloaded at `http://liris.cnrs.fr/advene/packages/generic-owl-export/0.1/`

a schema-specific mapping, targeting a specialized ontology expressing the specific semantics of that schema. Any intermediate position is of course possible; useful mappings will generally combine information from the annotation structure itself with the particular semantics of a given schema. [16] have actually experimentally demonstrated the relevance of mixing an audiovisual ontology with a domain ontology for the description of AV documents.

Let us note that the mapped ontological structures are highly constrained, both by the constraints bearing upon the document model, and by those enforced by the mapping itself. Hence, even if the user is not an expert of ontological models, the mapping gives a reasonable guarantee on the relevance of the ontological structures it produces, and hence on the relevance of the inferences performed on it.

Finally, Advene structures (annotation types, relation types or schemas) can even be *specifically* designed to be mapped into an ontological structure. The annotation type *Character* is an example of such an ontology-driven model. It is based on the FOAF ontology[14]: annotations of this type were intended from the start to represent instances of the FOAF class *Person*, and the relation with the AV fragment, the FOAF relation *depictedBy*. This approach is similar to the two-ways articulation proposed in [24]: depending on the used tool, the user will have a document-based or ontological point of view on the data, but no point of view is "more valid" than the other, with a *focus* on both models at the same time.

**Application.** In the Nosferatu package, we map Advene structures into an ontological model using three vocabularies: the generic ontology of Advene structures, the FOAF ontology to represent more specifically annotations of type *Character* and the relations between them, and an ontology of our own for rep-

[14] `http://xmlns.com/foaf/0.1/`

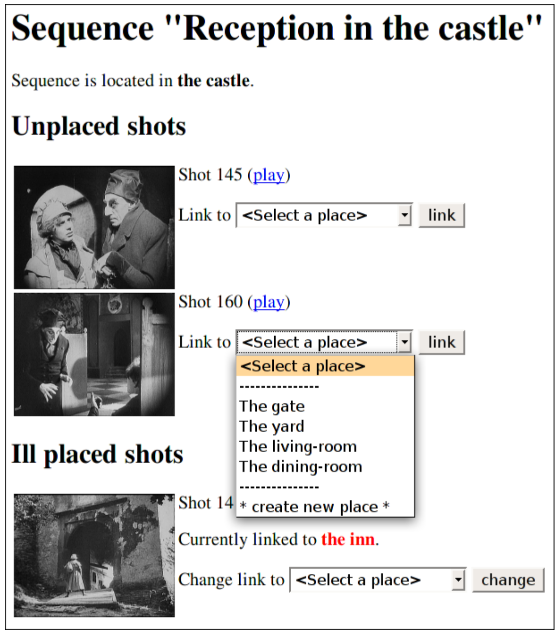


Figure 6: A reporting and advising view using inference of Places

resenting annotations of type *Place*. More precisely, the Places in the Nosferatu package have been organized, in the ontological model, by a *sub-place* transitive relation, involving places that do not appear as annotations (e.g. Transylvania, which has for sub-places the castle and the inn). Disjointness between places has also been stated (e.g. the castle and the inn, Transylvania and Germany).

Inference engines can then perform non-trivial computations on the mapped ontological structures. We can state and check, for example, that all the Shots in the Sequence "Reception in the castle" happen in a sub-place of the castle[15] (the yard, the living room...). This constraint can be used to perform *consistency checking* on the annotation structure.

It actually proves more practical to define the class of *IllPlacedShots*, and declared that all Shots of the Sequence "Reception in the castle" are ill placed if

[15] in DL syntax, this would be expressed like
$\exists contains^{-}.\{seq\_reception\_in\_the\_castle\} \sqsubseteq \exists happens\_in.(\exists sub\_place\_of.\{castle\})$

located outside the castle. Then a *reporting* view can be produced to list all such ill placed Shots. Inference can further be used to build a list of possible Places for those Shots (figure 6) and *advise* modifications in the annotation structure. It can be seen that places such as Hutter's house or the transylvanian inn, which are not sub-places of the castle, are not proposed.

### 5.2 Embedding ontological structures

A second possible articulation of models is to *embed* RDF/OWL as a content type for document elements (annotations, relations, resources). Inferences, especially consistency checking, can then be performed on the content of each individual element. It can also be performed on the merged content of all such elements, in order to take into account all the information attached to the video.

But the interesting feature of distributing the OWL content across several elements is that one can also check the consistency of the merged content of only a *part* of those elements, grouped according to some aspect of the document structure: annotations overlapping a given fragment, annotations in relation with one another, elements matching a given Advene query, etc. (see figure 7). That way, the user can have a fine control on the elements of the document model that will be shared with the ontological model (central part on figure 4).

In such use cases, it is not required that the ontological structure distributed in document elements was globally consistent; only local (or contextual) consistency may be desired. Note that other works [6] tackle the problem of contextual reasoning, but purely from the point of view of ontological models, assuming a familiarity of the user with those models. We, on the other hand, propose to define the notion of context by means of the document structures, with all the subjectivity they allow. We envision that this bond to document structures can make it easier for novice users to handle ontological structures and inferences.

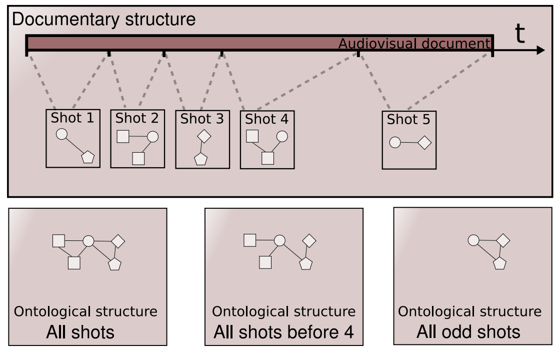


Figure 7: Top: a movie is annotated with *Shot* annotations, whose contents are RDF/OWL graphs. Bottom: examples of three different ontological structures built from three queries on these annotations.

**Application.** The most straightforward application of that "local" reasoning is to use temporal relations. One could for example describe, with an ontological structure, the action of each shot in the content of the corresponding Shot annotation[16], then check the *consistency* of "all that has been stated up to a given moment". One could also look for Shots whose content is not entailed by the merged content of all the preceding Shots, in order to *report* the Shots inflecting the storyline (those where something unexpected happens). One could also decide to use different types of annotations to represent the knowledge of different characters, and compare their respective points of view at some point of the story (are they consistent? is one entailed by another?).

But temporal relations are not the only ones that can define a reasoning context; any Advene relation can be used. We can for example reason on all the Shots happening in a given Place or involving a given Character, and infer some local truth in those Shots that are not always verified elsewhere. For example, one could check that Hutter is always feeling fearful in the living room of the count's castle, or that the count always appear during the night. An appropriate Advene query may allow a finer definition of refining context; one could check,

[16]provided an adequate ontology.

e.g., whether Hutter is always fearful in any indoor sub-place of the castle[17].

Note that our goal with such *advanced queries* is not to replace modal, epistemic or temporal DLs [5], but rather to anchor ontological reasoning in tried document-related practices. The *focus* of that kind of articulation is clearly on the document model.

## 5.3 Manipulating ontological structures as such

A specific feature of AV document models *with schemas* (like Advene), is that they provide a mean to attach additional semantics to their structure. A consequence is that, while simpler models may only provide additional information by embedding it (e.g. in annotations), one can consider that models with schemas may convey additional information in their structure. Hence, the boundary between *mapping* and *embedding* is somewhat erased.

Let us take for example a closer look at the annotation type *Character*. On the one hand, literal properties like *name* or *title* are more naturally represented in the content of corresponding annotations. On the other hand, relations between Characters, while they may also be embedded in annotation content, are preferentially reified as Advene relations, so as to be usable by document manipulation tools as well.

**Application.** The same hybrid (between mapping and embedding) design has been used in articulating the Advene model with the ScholOnto ontological model [25]. ScholOnto is an ontology for modelling and reasoning about arguments; we plan to use it to represent argumentative film analyses. There is no unique way to link an argument to the video, hence to unique way to represent ScholOnto structures in the Advene model. An AV fragment may support a statement (the basic element of an argument in ScholOnto) or a rela-

[17]there is actually one shot where he is not...

tion between two statements; some statements may not be supported by any AV fragment (but rather be considered as common knowledge or as a consequence of several other statements). Furthermore, an AV fragment may correspond to several different statements : what a character says, what he or she means, what the director wants to convey...

Here the complexity of the task makes ontological tools more suitable than document manipulation ones. We will provide the user with a dedicated interface for building an argument, *focusing* mainly on ScholOnto concepts, and transparently turning them into Advene structures in the most appropriate way (which is a mix of structural mapping and embedding in the annotations). The user may in turn use the produced document structures with classical Advene tools, but will not have to manage the complex production process of that structure.

To sum up this section, we position in table 2 our three application examples in our framework for studying model articulation.

| Application | Modality | Directionality | Focus |
|---|---|---|---|
| FOAF mapping | map | 2-way | both |
| "local" reasoning | embed | 2-way | Advene (document) |
| ScholOnto | hybrid | 1-way | OWL (ontological) |

Table 2: Features of our application examples

## 6 Conclusion and future work

In this article, we have first presented how we question the notion of "semantic multimedia": to us, multimedia semantics resides in the models and the tools that are used to manipulate multimedia documents. Document models and tools are therefore knowledge containers of such semantics, but there versatility and pragmatic efficiency is done at the expense of lacking formal semantics. On the other hand, ontological models offer formal and explicit semantics, but are not adapted to document processing and presentation. Our claim is that both types of models should be used for what they are best suited: document models for manipulating document structures, and ontological models for inference-based processing, explicit knowledge sharing and reuse, etc. An important issue is then to define and study precisely the notion of document models and ontological model articulation. We have proposed a basic conceptual framework as a first step towards that goal, identifying modality, directionality and focus as relevant features of model articulations. To illustrate and study how model articulation prove useful in the field of semantic multimedia, we have also presented our own document model and prototype for video annotation and hypervideo design (Advene), and we have considered three different cases of articulations. Each case study has been illustrated with one or several examples working within the ontological extension of our prototype, providing consistency checking, reporting, advising and advanced queries.

Future work implies deeper theoretical investigation on the notion of model articulation, and the study of the relations between document structures, models and semantics, and actual practices of document uses. On the more practical side, we will continue our developments of the Advene prototype and its ontological extension, so as to study precisely the various cases articulations we proposed. Applications will mostly focus on film study with two main directions, within the Ciné Lab project[18]. The first is related to how to describe and share ontologically described practices (complementing document schema-based description with ontological means of checking their good usage). The second concerns hypermedia ontology-based argumentation in film analyses.

[18] `http://liris.cnrs.fr/advene/cinelab.html`

## References

[1] Ben Adida and Mark Birbeck. RDFa primer 1.0, embedding RDF in XHTML. W3C Working Draft, 2007. `http://www.w3.org/TR/xhtml-rdfa-primer/`.

[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. From video information retrieval to hypervideo management. In *Corimedia, the international workshop on multidisciplinary image, video, and audio retrieval and mining*, Sherbrooke, Canada, Oct 2004. 10 pp.

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

[5] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[6] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. Contextualizing ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(4):325–343, October 2004.

[7] Teresa Chambel, Carmen Zahn, and Matthias Finke. Hypervideo design and support for contextualized learning. In *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, pages 345–349, Joensuu, Finland, august 2004.

[8] Pierre-Antoine Champin and Yannick Prié. Models for sustaining emergence of practices for hypervideo. In Peter King Marc Nanard, editor, *Workshop On Semantically Aware Document Processing And Indexing (SADPI 07)*, may 2007.

[9] James Clark. XSL Transformations (XSLT) Version 1.0. Recommendation REC-xslt-19991116, Word Wide Web Consortium, 1999.

[10] Dan Connolly. Gleaning resource descriptions from dialects of languages (GRDDL). W3C Working Draft, 2006. `http://www.w3.org/TR/grddl/`.

[11] Mike Dean and Guus Schreiber. OWL Web Ontology Language. W3C Recommendation, `http://www.w3.org/TR/owl-ref/`, Feb. 2004.

[12] Henri Fallon, Alexis de Lattre, Johan Bilien, Anil Daoud, Mathieu Gautier, and Clément Stenac. *VLC User Guide*. VideoLAN Project, 2003.

[13] Roberto García and Òscar Celma. Semantic integration and retrieval of multimedia metadata. In Siegfried Handschuh, Thierry Declerck, and Marja-Riitta Koivunen, editors, *5th International Workshop on Knowledge Markup and Semantic Annotation (SemAnnot 2005)*, Galway, Ireland, November 2005. CEUR Workshop Proceedings.

[14] Tom R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

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

[16] 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.

[17] Rohit Khare. Microformats: The next (small) thing on the semantic web? *Internet Computing, IEEE*, 10(1):68–75, 2006.

[18] Norman W. Paton, editor. *Active Rules in Database Systems*. Springer Verlag, New York, 1999.

[19] Roger T. (coll) Pédauque. Document: Form, sign and medium, as reformulated for electronic documents. Pre-publication, `http://archivesic.ccsd.cnrs.fr/sic_00000594`, July 2003.

[20] 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.

[21] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Working Draft, 2007. `http://www.w3.org/TR/rdf-sparql-query/`.

[22] 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.

[23] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 2006. (to appear).

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

[25] V. Uren, S.J. Buckingham Shum, G. Li, and M. Bachler. Sensemaking tools for understanding research literatures: Design, implementation and user evaluation. *Int. Journal of Human Computer Studies*, 64(5):420–455, 2006.

[26] Jacco van Ossenbruggen, Frank Nack, and Lynda Hardman. That obscure object of desire: Multimedia metadata on the web, part 1. *IEEE MultiMedia*, 11(4):38–48, 2004.

[27] Zope Corporation. *Zope Page Templates reference*, 2004. `http://www.zope.org/Documentation/Books/ZopeBook/2_6Edition/AppendixC.stx`.