CHM: an Annotation- and Component-based Hypervideo Model for the Web

Madjid Sadallah (msadallah@mail.cerist.dz)
DTISI - CERIST
Alger, Algérie

Olivier Aubert (olivier.aubert@liris.cnrs.fr) and Yannick Prié (yannick.prie@liris.cnrs.fr)
Université de Lyon, CNRS Université Lyon 1, LIRIS, UMR5205, F-69622, France

Abstract. Hypervideos are hypermedia documents that focus on video content. While they have long been deployed using specialized software or even hardware, the Web now offers a ground for them to fit into standardized languages and implementations. However, hypervideo design also currently uses very specific models limited to a single class of documents, or very generic hypermedia models that may not appropriately express their specific features. In this article we describe such features, and we introduce CHM, an annotation-driven and component-based model to conceptualize hypervideos through a high level operational specification. An extensible set of high level components is defined to emphasize the presentation and interaction features modeling, while lower level components offer more flexibility and customization opportunities. Being annotation-based, the model promotes a clear separation between video content/metadata and their various potential presentations. We also describe WebCHM, an implementation of CHM with standard Web technologies that provides a general framework to experiment with hypervideos on the Web. Two examples are provided as well as a preliminary usage study of the model and its implementation to validate our claims and proposals.

Keywords: Annotation, Advene, CHM, Hypervideo, Component, Time and synchronization, WebCHM

H.5.1: Multimedia Information Systems/Video;
H.5.4: Hypertext/Hypermedia/Architectures, Navigation:

1. Introduction

Since its inception, and even before the term was coined by Ted Nelson (Nelson, 1965), the hypermedia idea has largely been implemented and used in various forms, the Web being one of its latest and most successful incarnations. Video becoming one of the major mediums accessed through the Web (at least in terms of volume), its integration in the Web experience is a key issue, all the more relevant that technologies and standards have now reached a maturity regarding the handling of audiovisual content. Integrating video into hypermedia spaces raises specific issues, due to the semantic gap between the low-level bit stream and the high-level human interpretation of video semantics and to video temporality, which adds another dimension to the already multiply-dimensioned hypermedia world.

With the aim to foster and study such integration, the Advene[1] team has been interested since 2002 in the expression of discourses that take advantage of hypermedia techniques to support points of view upon video material, such as film analysis or linguistic/gesture analysis. More precisely, we study and design systems that allow to build and share *hypervideos* as hypermedia documents created upon videos.

Based on this experience, we identified that one of the issues in building hypervideos was the complexity of the design task, which emphasized the need for a *dedicated operational hypervideo model*. Hypervideo, as a specialization of hypermedia, presents certain characteristics that we detail in this article. Currently, hypervideo design uses very specific models limited to a single class of documents, or very generic and all-purpose hypermedia models that do not take advantage of hypervideo characteristics and thus do not directly provide features aimed at hypervideo.

To address these issues, two main contributions are presented in this work. Our first contribution is the proposal of CHM, an annotation-driven and Component-based Hypervideo Model that has been designed so as to lower the above-mentioned usability barrier. An earlier version of the proposed model has been roughly presented in (Sadallah et al., 2011). CHM intends to lay out a dedicated hypervideo conceptual framework built around a component-oriented logical structure organized as a hierarchical tree of components with a focus on extensible and high level built-in components. CHM follows a view-based approach since it explicitly separates the document content from the potential presentations through the use of annotations. By emphasizing the importance of attaching annotation structures to video streams and generating variety of renderings, CHM offers a new point of view on video-centered hypermedia systems.

Our second contribution is the development of WebCHM, a practical implementation of CHM for Web hypervideo creation and rendering. Using standard Web technologies, WebCHM development is intended to validate the applicability of CHM principles to the design of online hypervideos. Moreover, the framework illustrates practically how the higher-level instrumentation of Web technologies can reduce the accessibility barrier of hypervideo design on the Web.

[1] Advene (*Annotate Digital Video, Exchange on the NEt* - `www.advene.org`) designates a project, a data model, as well as an open source, multi-platform application for creating hypervideos.

The article is organized as follows. Section 2 introduces hypervideos, presenting classical use-cases, main characteristics and common components. Existing models for specifying hypervideos are then presented in section 3, and the importance of annotations-based hypervideos is underlined. Section 4 presents the Component-based Hypervideo Model (CHM) and its principles through structural, spatial, temporal and linking aspects. In Section 5, a Web-based implementation of CHM is presented along with illustrative examples. A preliminary user study is presented in Section 6 while Section 7 discusses our proposals.

2. Hypervideos

2.1. A Definition

Within hypermedia documents, videos are commonly embedded as atomic, sequential, static and not easily navigable clips mainly used as support to give a better idea of a concept (Navarrete and Blat, 2002). Such integration of videos in hypermedia systems is not as deep as it should be, an issue that hypervideos address. A hypervideo can be seen as a kind of video-centered hypermedia document, that brings additional capabilities to videos allowing more elaborated practices and improved interactivity (Hammoud, 2006).

Several definitions of hypervideo exist (Chambel et al., 2004), depending on different points of view. While some authors emphasize the hypermedia aspect of adding information to digital video so that users can activate video hyperlinks and access additional rich content (VideoClix.tv, 2012; InnoTeamS GmbH, 2012), others highlight the storytelling aspects of dynamically creating non-linear and user-defined navigation paths into the videos (Sawhney et al., 1996). In this article, we simply define hypervideo as being an *interactive video-centric hypermedia document built upon audiovisual content.*

2.2. Some Use-Cases for Hypervideos

The large applicability spectrum of hypervideo covers various areas.

Interactive movies. At the core of many hypervideo projects is the aim to develop novel ways to navigate within movies. For instance, Aspen Movie-map (Lippman, 1980) allowed users to navigate the street of Aspen, Colorado. HyperCafe (Sawhney et al., 1996) targeted new kinds of cinematic experiences by offering filmmakers the possibility to define different narrative sequences, and viewers the choice of the path to follow.

Augmented documentaries/movies. Augmenting video experience is another privileged use case for hypervideo. For instance, interactive television features additional and interactive content synchronized with a live transmitted program (Hoffmann et al., 2008). Interactive documentaries (Shipman et al., 2003a) allow the author to recombine video segments and expose alternate representations of video details with extra content. The Mozilla drumbeat "Web made movies" project (Mozilla, 2012b), that uses the popcorn.js framework for implementation, explicitly fosters the development of interactive documentaries.

Learning. In training and scholar systems, hypervideo supports the creation of rich and realistic learning environments (Tiellet et al., 2010) by offering dynamic visualizations and facilitating reflective learning and cognitive flexibility (Zahn et al., 2004). Hyper-Hitchcock (Girgensohn et al., 2004) is a hypervideo environment that enables direct edition of a particular form of hypervideo called "detail-on-demand video", which allows a single link out of the currently playing video to provide more details on the content being presented. This form of video can be used for instance as representation for an interactive multi-level video summary (Shipman et al., 2003b) by including multiple video summaries as well as navigational links between summary levels and the original video. Based on guidelines supported by media and cognition theories, HVet (Tiellet et al., 2010) is an example of a hypervideo learning environment for veterinary surgery. It provides a rich and flexible environment with the potential to support different cognitive modes, learning styles and phases. Greater control and autonomy are granted to the learner to explore video data complemented with external materials and presentation/navigation aids like maps and history.

Marketing. Some hypervideo providers, like *Adivi* (InnoTeamS GmbH, 2012) or *VideoClix* (VideoClix.tv, 2012) allow to design online video clips that link to advertising or e-commerce sites, or provide more information about specific products. For example, HyperSoap (Bove et al., 2000) is a short soap opera program in which a viewer can use an enhanced remote control to select clothing, furniture, and other items, and get information about them. Object identification and tracking algorithms are used in the hypervideo authoring process.

Video active reading. In the general case active reading consists in reading documents and producing objects (most commonly annotations) that will be then used to carry on the reading activity, or to produce some output document. This activity is routinely carried on texts, which can be commented with notes and graphics that serve as guides in the reading activity, allowing to quickly identify interesting parts for further browsing, as well as potential content for producing output from the reading activity (analyses, comments, etc.). Applied to video documents, active reading mainly consists in creating annotations that comment fragments of the audiovisual documents, and using these annotations both to carry on the document consultation/navigation, in a more structured way, and to produce new hypervideo documents combining annotations with the audiovisual document. Advene (Aubert and Prié, 2005) and Anvil (Kipp, 2001) are typical annotation-based video active reading systems.

2.3. Main Characteristics of Hypervideos

The existing hypervideo systems described above share a number of characteristics, and feature common display components. We describe in this part some of the common properties of hypervideos [2], and then describe commonly found components.

Interactivity. Hypervideos combine the powerful impact of video with hypermedia facilities into new interactivity forms featuring richer navigational possibilities. Doing so, it brings interactivity to video by integrating it in hypermedia spaces where it can be structured through hyperlinks defined in space and time (Chambel et al., 2011). Hypervideo documents provide navigational possibilities through hyperlinks, which anchors are time-based (or even spatio-temporal), and whose destinations can be inside the video itself, within another video or anything in the hypermedia space. A generic system like Youtube annotations (Fagá et al., 2010) offers means to define deep video linking through annotations. HotVideo (Finke, 2004) is an implementation of hypervideo that extends the concept of hyperlinks from text and images to any dynamic video object, making it possible to link parts of the digital video to different types of media.

Non-linearity. A high degree of flexibility is granted by hypervideos to compose video-based documents that foster knowledge perception by promoting an *active* reading experience and reflecting the audience dynamic engagement and influence on the document storyline (Hoffmann et al., 2008). Different exploration patterns can be proposed to the user through the definition of advanced features like video montage and inter-connections, and synchronous/simultaneous display of different videos.

[2] Common hypermedia concerns might be more pronounced in hypervideo due to the important risk of overstraining the cognitive capacities of users and putting them under time pressure during navigation (Tiellet et al., 2010). The potential of increased cognitive load might lead to user disorientation (Chambel et al., 2004): in addition to the common space disorientation generally found in hypermedia, the time-based nature of audiovisual documents brings time disorientation as well as time pressure through time-limited interactivity opportunities

Enrichments. The presentation of the content of a video used within a hypervideo can be supplemented by many types of enrichments. Such enrichments can be external, such as a table of contents, or some further material (text, images, web pages), or appear as a synchronous display of textual (subtitles, captions, links, etc.) or graphical overlays (images, figures, other videos, etc.). SIVA Suite (Meixner et al., 2010) is a Flash-based authoring and presentation system for interactive video that features video enrichment, clickable video objects and selection buttons for following plotlines.

2.4. Common Components of Hypervideos

Articulating video content and navigational capabilities or enrichments leads to designing new kinds of interfaces and related interactions (Shipman et al., 2003a). Several years of hypervideo design have lead to the emergence of similar and recurring visualization and interaction patterns accessible through one or several *hypervideo components*. We have studied a number of existing systems and identified some of these common components in the following list, and sum up in table I their use or availability in different hypervideo systems. No entry means that the pattern is not explicitly mentioned in the literature to be supported by the system.

- *Video player + controls*: A video player is obviously always available. It is interesting to distinguish its controls, since they can sometimes be limited or completely disabled.
- *Timeline*: A timeline component is defined in our context as a spatial representation of temporally situated metadata, where the temporal dimension is projected onto one of the spatial dimensions. One of its most conventional forms is a horizontal timeline where time is represented on the x-axis and metadata can be categorized along the y-axis. Note that a basic video slider does not stand in this category, since it does not feature any metadata.
- *Textual overlay*: A textual overlay presents additional information, such as captions or other textual information, placed over a video.
- *Graphical overlay*: A graphical overlay displays graphics over the video, which can be used to designate specific parts of the image.
- *Hotspot*: A hotspot is a graphical overlay with hyperlinking features.
- *ToC*: A table of contents is a textual representation of the basic documentary structure of the video document.

– *Maps*: A map acts like a graphical table of contents displaying graphical representation of metadata. Image maps for instance can be composed of the most relevant video frames and may act like visual summaries of the video (Chambel and Guimaraes, 2002).

– *Transcript*: A transcript is a text generated from a textual transcription of the audiovisual document, that allows to navigate from the transcription to the corresponding time in the video, and possibly highlights in some way the text that corresponds to the video fragment being played.

3. Specifying Hypervideos

Specifying a hypervideo means describing it in a digital language so that the machine can compute it and present it interactively to the user. In this section we consider *ad-hoc*, multimedia-based and annotation-based specifications.

3.1. Ad-hoc specifications

Most of hypervideo authoring and reading tools (Sawhney et al., 1996; Girgensohn et al., 2004; Chambel et al., 2004) rely on *ad-hoc* specifications that are fitted to particular needs and lack genericity. These systems use specific approaches for abstracting the produced documents with generally implementation-based representations, resulting in informal models to describe hypervideos. As a consequence, features such as semantic description, enrichment, video fragmentation and composition, robust addressing conventions, linkage and fragment accessing are not widely granted. Moreover, since the implied representations are mainly technically driven, they are not adapted to the evolution of the hypervideo documents model: any evolution entails software re-engineering, hence limiting the fostering of the emergence of useful hypervideo practices and applications.

3.2. Multimedia-based specifications

We consider two kinds of multimedia-based specifications of hypervideos: those that rely on general multimedia models, and those that take advantage of generic multimedia programming languages.

General multimedia models. Following the Ted Nelson hypermedia model extension to include "branching movies" (Nelson, 1987),

Table I.: Recurring components in existing hypervideo systems

System	**Video Player + controls**	**Timeline**	**Textual Overlays**	**Graphical Overlays**	**Hotspots**	**ToC**	**Maps**	**Transcript**
Adivi (InnoTeamS GmbH, 2012)	Y	Y	Y	Y	Y	-	-	-
Advene (Aubert and Prié, 2005)	Y	Y	Y	Y	-	Y	-	Y
Anvil (Kipp, 2001)	Y	Y	-	-	-	Y	-	Y
HotVideo(Finke, 2004)	Y	-	-	-	Y	-	-	-
HVet (Tiellet et al., 2010)	Y	Y	-	-	Y	Y	Y	Y[a]
HyperCafe (Sawhney et al., 1996)	Y[b]	-	Y	Y	-	-	Y[c]	-
HyperFilm (Pollone et al., 2002)	Y	-	-	-	-	Y[d]	-	-
HyperHitchcock (Shipman et al., 2003a)	Y	Y[e]	-	-	-	-	Y[f]	-
HyperSoap (Bove et al., 2000)	Y[g]	-	Y	Y	Y[h]	-	-	-
Popcorn (Mozilla, 2012a)	Y	Y	Y	Y	Y	Y	Y	Y
SIVA Suite (Meixner et al., 2010)	Y	-	Y	Y	Y	Y	Y	-
VideoClix (VideoClix.tv, 2012)	Y	-	Y	Y	Y	Y	Y	-
VisualSHOCK Movie (America, 2000)	Y	-	-	-	Y	-	-	-
Youtube (YouTube, 2009)	Y	Y	Y	Y	Y	-	-	Y

[a] Not explicitly and not interactive
[b] No video control: "*The video sequences play continuously, and at no point can the user's actions stop them*"
[c] At the beginning: overview shot of the entire scene
[d] Link table
[e] With shots as tracks
[f] Workspace View
[g] Remote control
[h] Internal references only

many researches have addressed the field of interactive video. Usual approaches to theorize such documents have considered them from a very general hypermedia/multimedia perspective enhanced with specific features like video hyperlinks. Models and systems for continuous media integration in hypermedia were discussed since the Amsterdam Hypermedia Model (AHM) (Hardman et al., 1994) proposal, providing mechanisms for structuring, linking and synchronizing dynamic multimedia content. AHM, based on the Dexter reference model (Halasz and Schwartz, 1994), has been designed to cover all relevant theoretical concepts of hypertext and hypermedia systems, regardless of technical limitations; no modeling methodology supports all their features (Zoller, 2001). NCM (Nested Context Model) (Casanova et al., 1991; Soares and Rodrigues, 2005) is a hypermedia conceptual model that provides support for temporal and spatial relationship definition among media objects. It targets content and presentation adaptation, and also distributed exhibition of an application on multiple devices.

Generic multimedia languages and tools. Another type of multimedia-based specification consists in using general multimedia-oriented programming languages to implement hypervideo. We present several of these languages.

The Synchronized Multimedia Integration Language (SMIL) (Bulterman, 2008), inspired from the AHM model, is the W3C recommendation for building time-based multimedia documents, designed to allow great complexity and sophistication in time-based scheduling and interaction of multimedia components and documents. SMIL enables authoring of interactive rich media and multimedia presentations, integrating audio and video with images, text or any other media type. Although new and interesting features have been added along the versions, specific hypervideo support has not received much attention (Tiellet et al., 2010).

The Nested Context Language (NCL) (Silva et al., 2004) is a declarative language based on the NCM conceptual model for hypermedia document specification, with temporal and spatial synchronization among its media objects. It treats hypermedia relations as first-class entities through the definition of hypermedia connectors, and it can specify arbitrary semantics for a hypermedia composition using the concept of composite templates. NCL modules can be combined to other languages, such as SMIL, to provide new facilities. While SMIL provides high level constructs defining a restricted set of temporal relationships, NCL allows the creation of custom relationships from a toolkit of language primitives as objects (Jansen et al., 2010).

Traditionally, on the Web, content providers have also relied on third party pieces of software known as browser-plugins, mainly Adobe Flash, to embed multimedia content. These technologies raise many issues concerning security, reliability, integration and performance. For instance, their binary nature prevents the content from being accessible from screen readers, handled by assistive technologies, indexed by search engines or deep linked and syndicated (Jansen and Bulterman, 2009).

Finally, HTML5 (Hickson, 2011) targets native browsers support for audiovisual content and provides rich structured constructs for declarative Web documents description. Most media items are referenced by a URL, and not embedded in the presentation. While the flow-based layout can be enhanced with CSS, HTML5 does not provide declarative mechanisms for temporal composition (Jansen et al., 2010). Using the new HTML5 features, *Popcorn.js* (Mozilla, 2012a) is a framework, backed by the Mozilla Foundation, for connecting video to the rest of the Web, linking it into the hypertext world by pulling data from the Web and synchronizing it with video documents. It features various components that can display information from various sources (Google Maps, Twitter, Wikipedia, etc.) and can be customized through a JavaScript API.

3.3. Annotation-based Specifications

Annotation-based specifications of hypervideo take advantage of the notion of video annotation as metadata associated to video fragments, that can be further used as means to integrate video contents into hypervideos.

3.3.1. *Video Annotations*

The nature of video data makes it not directly suitable for traditional forms of data access, indexing, search, and retrieval (Li et al., 2010). Moreover, digital video documents have long raised concerns about how to link and navigate from and to precise parts, enrich and explain the contents, re-arrange or reveal story structures, etc. This motivated efforts towards the use of video annotations to index and retrieve video fragments on the one side, and to disclose, explain or augment the knowledge carried by the video on the other side.

In this article we define an annotation as a *piece of data associated to a video fragment* (Aubert and Prié, 2005), which is a logical video segment defined by start and end timecodes. Annotations can be generated through various means, from completely human-made annotations, precise and focused but tedious to produce, to automatically extracted annotations, cheaper to produce but less precise (Kokkoras et al., 2002; Vendrig and Worring, 2003). Intermediate approaches, such as computer-assisted manual annotation methods, can also be used (Davis et al., 2004). Annotation-based modeling of audiovisual documents adds a supplementary temporalized content layer on top of video documents (Aubert and Prié, 2005; Fagá et al., 2010) thus defining an infrastructure for augmenting video with meaningful data. Such data can be further used to manage and manipulate videos for analyzing, indexing, searching and generating hypervideos.

3.3.2. *Hypervideo as Annotation-based Video-centric Hypermedia*

Once a video is annotated, it is possible to take advantage of the annotations to build hypermedia visualizations that break the linearity of the video and create a non-linear information space, hence hypervideos (Miller et al., 2011). Designing hypervideos in this way emphasizes the importance of metadata and annotations in the process of creating video-centric hypermedia documents.

Most hypervideo editors use the concept of annotation and provide means to add such data to video. Systems like HyperCafe, HyperSoap, VisualShock MOVIE (America, 2000) and HyperFilm (Pollone et al., 2002) were among the earliest research projects to integrate video annotation as a core concept (Miller et al., 2011). Hyper-Hitchcock allowed viewers not only to interact with annotated videos but also to create one's own annotations and share them with others over the Internet in real-time. Tools such as Adivi (InnoTeamS GmbH, 2012), Advene (Aubert and Prié, 2005), Anvil (Kipp, 2001) and MediaDiver (Miller et al., 2011) provide the functionality to annotate videos with text, links or any rich content and to generate video centric presentations from these annotations.

While many works target the use of annotations for audiovisual document description and enrichment, the annotation structure is often tightly linked to the video when it is not completely embedded in the stream or in the player. This is the case for systems like Youtube that use annotations mainly as captions, branching anchors and graphical overlays over the video, with no clear separation between metadata content and its representation. Moreover, while systems like Hyper-Hitchcock (Girgensohn et al., 2004) use annotation as a key concept for hypervideo document design, their approach for annotation definition does not make use of any standard format or explicit data model (Alisi et al., 2009) and the implied uses and representations are mainly technically driven. Therefore, as these systems only provide a few of the necessary support for defining annotation-based hypervideo models, the current use of annotations prevents the emergence of fully annotation-based techniques for hypervideo design. Our Advene project (Aubert and Prié, 2005) is an effort to develop such a model and to propose solutions for annotating videos and generating hypervideos.

3.3.3. *The Advene annotation-based hypervideo conceptual model*
We have previously (Aubert and Prié, 2005) proposed a conceptual hypervideo model, focused on annotations. It is implemented in the Advene application, and constitutes a conceptual framework for the Component-based Hypervideo Model we will present in this article. We thus briefly expose this approach.

We first define (see figure 1) an *Annotated Audiovisual Documents Base* as a *set of video documents associated with an annotation structure*. Some elements (actual annotations) of the annotation structure feature spatio-temporal links within the AV document.

A *view of an AV Document Base* is then defined as a "*way of presenting*" it, combining information extracted from the video documents and information from the annotation structure. Two main characteristics can be associated to views: 1/ the balance of information obtained through either source (video documents and annotation structure): a view can be built exclusively from the AV document (its plain visualization), exclusively from the annotation structure (a plain table of contents of the video, generated from specific annotations), or using a mix from both sources; 2/ the possibility to access the AV document temporality (i.e. the ability to visualize the AV document and control its playing).

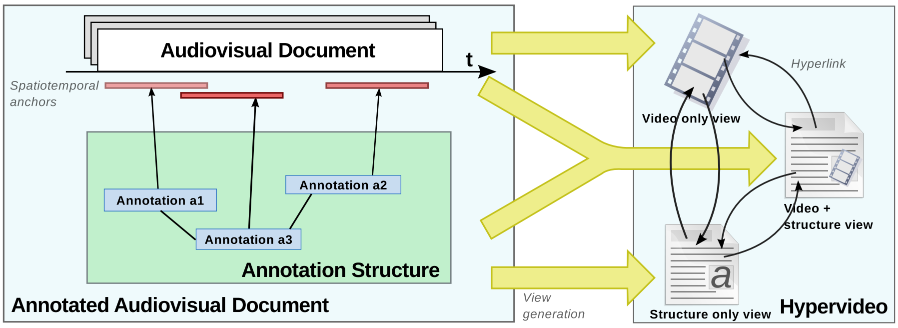

Figure 1. Annotation-based hypervideo in Advene: a *set of views* that 1/ use information from *both* the video documents *and* the annotation structure and 2/ give access to the audiovisual document temporality

We can then define an annotation-based *hypervideo* as a *set of views of the annotated audiovisual document base that on the one hand uses information from both the video documents* and *the annotation structure (and not exclusively from one source) and on the other hand gives access to the temporality of the video documents.*

As an illustrative example, let us consider a plain table of contents featuring links to the visualizations of corresponding fragments. Such a hypermedia document is composed of two views that are not hypervideos on their own, but becomes a hypervideo when combined. This approach that focuses on adding annotation content layers to videos and defining hypervideos as visualizations combining both metadata and videos, is very generic. It is also coherent with the characteristics of hypervideo we presented earlier.

The Advene software provides an implementation of our framework (Advene/Cinelab data model[3]), but as mentioned above, its genericity and flexibility hamper its usability, also limiting hypervideo design to using the standalone Advene software. From this assessment, we decided to extend our work by designing a more operational and visualization-oriented model of hypervideos.

4. CHM: an Annotation-driven and Component-based Hypervideo Model

CHM stands for *Component-based Hypervideo Model*. It is an annotation-driven and component-based model to conceptualize hypervideos and describe their main features.

4.1. Rationale

As we have seen in section 3, annotation-driven approaches present attractive alternatives to *ad-hoc* or multimedia specifications for hypervideo design, maintenance, exchange. With data separation from its possible visualizations, maintaining the document structure independently from the audiovisual stream is easier. This enforces security management by separating concerns between data and metadata, and enhances collaboration options by requiring only the annotation structure to be updated or exchanged. Such a data model also sustains metadata-based indexation and allows generation of multiple hypervideo presentations from the same annotation structure. CHM further builds on the annotation-based hypervideo conceptual framework of Advene.

Moreover, our approach is *component-based*, meaning that the logical view of a hypervideo is represented using a specification that tries to be concise and expressive, presented as a hierarchy of components. Such an approach eases the conceptualization and design of hypervideos by providing common "classical" components as presented in section 2.4.

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

It also allows the model to be extensible since new components can be created from existing lower-level ones.

Hypervideo is a specialization of multimedia and thus, it can be analyzed along the four dimensions of multimedia documents (Roisin, 1998): logical (document organization), spatial (graphical layout), hypermedia or links (relations between documents and document fragments) and temporal (temporal ordering of the document objects). Consequently, we provide within CHM representations for these multimedia dimensions.

The remainder of this section presents the CHM model in details, while the following section presents WebCHM, a Web implementation of CHM that acts as a proof of concept of CHM assumptions.

4.2. CHM Logical Model

4.2.1. *CHM Core Components*

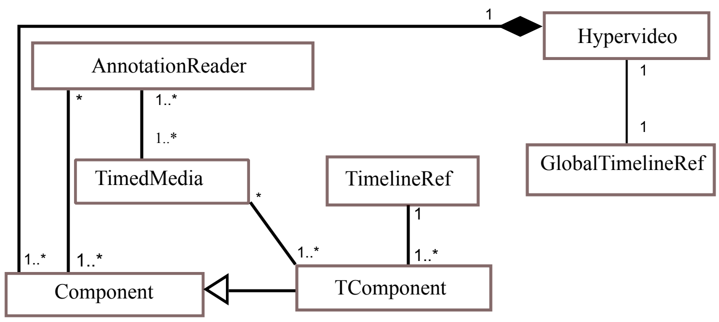

Figure 2. CHM Core Components

For the logical hypervideo representation, CHM uses the principle of nesting hierarchical high level components. Following this model, as shown in figure 2, a hypervideo is composed of a set of low and high level components, building blocks that represent formal information and composition units. Each `component` element is associated with a list of composition, placement, synchronization and behavioral attributes supplied by the author or retrieved from an annotation structure, through `AnnotationReader` components.

A hypervideo references at least one main audiovisual document accessed through the `TimedMedia` element that addresses a temporalized stream, audio or video. A TimedMedia component has an intrinsic duration; when played through a player component, it conveys a timing capability to the document, expressed by a virtual reference - an abstract clock - called *TimeLine Reference* (TLR). The TLR is used to synchronize the display of rendering components, following the

CHM temporal model that we present in section 4.5 and using timing metadata provided through `AnnotationReader` components. It can also be manipulated by some components (typically by the player control component, but also by any other component) to modify the rendering of the document. While a generic component within a hypervideo may not relate to any TLR, and if so is said to be time-independent, components bound to a TLR are specialized `TComponent` elements with synchronization capabilities.

Many players (therefore, many TLRs) may be present within the same document, defining different hypervideo sub-documents possibly spatially and/or temporally related. This allows synchronization between numerous hypervideo sub-documents and to use one to enrich another. This synchronization is expressed in term of temporal constraints between the corresponding TLRs and is handled by a global timeline reference, addressed by the `GlobalTimelineRef` component

4.2.2. *CHM Plain Components*

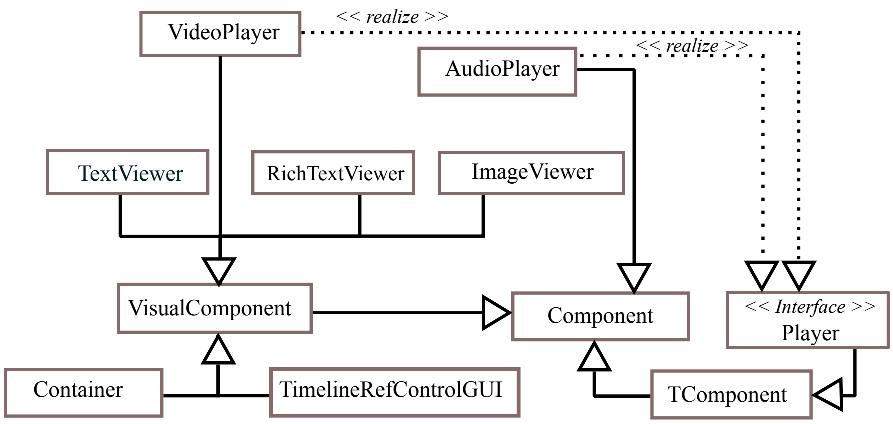

Figure 3. CHM Plain Components

Figure 3 presents the basic components that form a hypervideo. While `Component` elements are generic for handling data, content with visual manifestation is rendered through `VisualComponent` elements. Presentation specification attributes are associated to components and can be used by the rendering engine. Specific synchronized *display components* offer interactive interfaces for rendering temporalized data, provided as annotations. Multiple *AnnotationReaders* (see 4.3) can dispatch annotation data, either user supplied or possibly automatically extracted from the media elements (textual transcription, screenshots, videoclip references, etc.).

Among plain display components, the continuous media players such as `VideoPlayer` and `AudioPlayer` present a generic `Player` interface for rendering and interacting with content. Document content viewers such as `TextViewer`, `RichTextViewer` and `ImageViewer` allow the display of the textual and graphical content, retrieved from the annotation structure or defined as presentation information. The rich text viewer is a general container for heterogeneous content like HTML pages, RSS feeds and generic XML-based content. For instance, a synchronized *Wikipedia* content retrieved by a set of URLs can provide wider information about different topics. The `Container` element is an abstract receptacle for components grouping in order to ease their spatial clustering and to unify their processing. The `TimelineRefControlGUI` element allows the definition of a graphical user interface for controlling and interacting with the TLR.

To enhance video accessibility, *Text Captions*, *Graphic Overlays* and wider *Multimedia Overlays* can be placed over the video object, like captions, animations and different graphical enrichments. These overlays are instances of the appropriate viewers placed over the video player interface with appropriate temporal and spatial attributes. These components are convenience elements and are not explicitly defined, since any visual component can be overlaid by a multimedia content through a proper layout definition.

4.2.3. *CHM High Level Components*

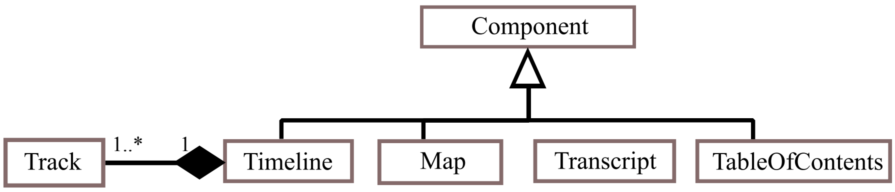

Figure 4. CHM High Level Components

For hypervideo document design, we propose a set of high level components shown in figure 4, built upon the plain ones. We have isolated this first set of high-level components by reviewing a number of hypervideo systems or frameworks, in order to identify common recurring patterns (see section 2.4 and table I). We plan to further refine these and define additional components. This extensible set of useful built-in components eases the coding task. When a needed component does not exist, the author can still create it from the existing lower-level and possibly other high level ones.

Transcript: The `Transcript` component displays an interactive textual transcription of the audiovisual document. It allows to navigate in the video by activating a portion of the transcription. It also highlights the displayed text in a synchronized way with the video being played.

Maps and ToCs: Maps and tables of contents give the reader means for exploring the hypervideo document in a guided fashion and for selecting a particular narrative path through appropriate entries which are specific contexts (time intervals, layout, etc) of the document presentation. A document map gives a general sketch of the presented content and offers a branching opportunity to navigate to a particular perspective. Many maps can be defined within the document to illustrate different features. The `Map` component can be made of textual or graphical entries like screenshots. The `TableOfContents` (ToC) component defines a hypervideo table of contents with navigation capabilities and reveals the structure of the video regarding a selected feature or annotation type. Many tables of contents may be instantiated, presented as a plan or a hierarchical tree.

Timelines and Tracks: A `Timeline` component is a visual interactive representation of the hypervideo time, to spatially place particular features over time in a graphical and chronological representation. The timeline is supplied with a slider to indicate the current position and standards buttons to control the presentation playback. Timelines place media elements, meaningful annotations and links along a timed axis, on different tracks. A `Track` component is the atomic temporal media representation, showing the active period of the corresponding annotation. The time axis is represented in a relative way because the effective tracks begin, end and duration may not be known before runtime (for instance when they are event-based).

4.3. Annotation-based Model

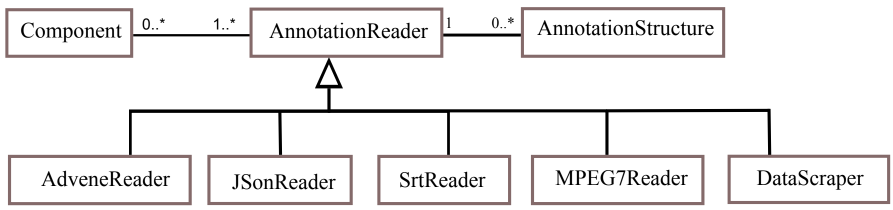

Figure 5. CHM Data Access Components

Matching the annotation-based hypervideo framework and the Advene/Cinelab data model, a CHM annotation is defined as any information associated with a logical spatio-temporal video *fragment*, that defines its scope (begin and end timecodes) in video time and its coordinates relatively to the video frames. For instance, a spatio-temporal anchor is defined by an annotation that addresses a fragment covering its presentation interval. Attributes of an annotation include its *type*, *media reference*, *begin/end timecodes* and *content*. Depending on user requirements, the proposed annotation attributes can easily be extended and adapted to specific needs. Annotations add structure and content to audiovisual documents. CHM relies on them in many ways to generate hypervideos and to supply them with interaction and visualization artifacts like anchors/hostpots and links, overlays, subtitles, comments and tables of contents.

Data access components presented in figure 5 are middleware components with functional interfaces that offer unified access to the data structure (annotations and resources). Multiple readers can provide annotation data, depending on storage format and structure schemas. The `AnnotationReader` element describes the generic data access interface and can be used for defining more specialized readers to access data stored with specific formats. We provide dedicated data readers for some of the most common formats: (1) `JSonReader` allows reading from a *JSon* file, (2) `SrtReader` allows using *SRT* files, (3) `MPEG7Reader` gets data from MPEG-7 files, (4) `DataScraper` generates annotations by scraping DOM subtrees (from the current document or from external ones) and (5) `AdveneReader` is the data access component for annotations produced with the Advene annotation tool.

4.4. CHM Spatial Model

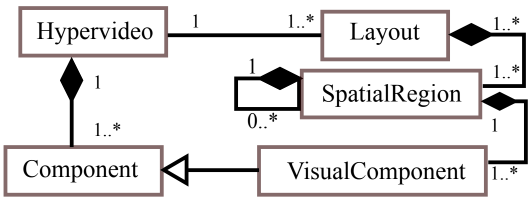

Figure 6. CHM Spatial Model

The spatial model presented in figure 6 describes the spatial arrangement of the document visual content on the presentation interface. A `VisualComponent` is a component with explicit visual manifestations. A non visual component is usually an audible-only component or a hidden one. The visual components are placed within spatial regions (`SpatialRegion` element) which may embed other spatial regions. Placements and dimensions can be expressed explicitly or implicitly, absolutely or relatively. The `Layout` root element determines the placement of these regions on a virtual rendering surface.

The layout and spatial regions have attributes for presentation specification like sound level, default fonts and dimensions. Along with attributes of the contained high and lower level components, this model encapsulates the AHM notion of *Channel* used for the same purposes.

4.5. CHM Temporal Model

The document temporal specification is achieved through a timeline-based model. The timeline metaphor is often used within continuous media editors (Bulterman and Hardman, 2005) such as LimSee3 (Deltour and Roisin, 2006). Such models use an explicit time scale as a common temporal reference for all media objects.

4.5.1. *TimeLine Reference Paradigm*

A *Timeline Reference* (TLR) is a virtual time reference attached to a video playback component or to the global document, in order to schedule related document components. Time-based components are activated/deactivated when reaching specific timecodes provided or computed by reference to the TLR. The non time-dependent objects are associated with the global document clock, the top level reference of the entire user-declared TLRs.

The access and control of a TLR is performed thanks to the "position", "state" and "duration" attributes. *Position* indicates the playback point of the TLR, while *state* indicates whether the TLR time is in progress, paused or stopped. *Duration* is a read-only attribute holding the length of the TLR. Any update of the TLR position or state affects the related TComponents playback.

4.5.2. *Multimedia Synchronization*

The time scope of a component can be provided through absolute timestamps, supplied by the author or retrieved from the related annotations timecodes, or can be event-based which defines item-relative relations, as will be explained in 4.7.

Media synchronization can be *hard* or *soft*. A hard synchronization forces the media to maintain a high synchronization with its TLR. A soft synchronization allows the media to slip in time with respect to the TLR; synchronization is then available for some meaningful instants: start and end of the media playback. During the rendering interval, the synchronization is not actively maintained, introducing great presentation flexibility. In both cases, pausing or stopping the TLR implies pausing or stopping all the related media components.

Document items "de-synchronization" is an issue in many scenarios, due for example to distant resource access through an unstable or unpredictable network. In such cases, the document presentation has to remain synchronized with the main video addressed by the TimedMedia via the TLR. Hence, if the main video pauses for buffering, all related timed components will be paused. However, if such a timed component became de-synchronized, the presentation would not pause or stop; instead, the TLR would try to restore the content synchronization by resolving its temporal playback position at each time update.

4.6. CHM Link Model

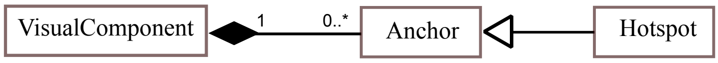

Figure 7. CHM Link Model

Differently from common hypermedia models, CHM hypervideo links are defined in space and time allowing them to appear, move (change location) and disappear relatively to the timeline reference. They are unidirectional, defined as attributes of the source anchors and represented by the `Anchor` and `Hotspot` elements in figure 7. There is no separate link (as in AHM) within the model; SMIL and HTML also do not use separate link components.

A classic hypertext `Anchor` can be defined on a specific region of a textual or graphical component. When placed on a region of a continuous media, with spatial and temporal constraints, it is represented by a `HotSpot` element. A hotspot is a TComponent positioned over the video player interface which triggers events when activated. Hotspots can be defined through a structure to describe a moving region whose location changes over time.

A link may be internal or external. An internal link leads to a particular video hotspot, an instant of the timeline reference or any other point in the hypermedia space. Activating such a link causes a temporal shift of the presentation by an update of the TLR position. A return behavior as already introduced in (Girgensohn et al., 2004) can be specified to express whether the presentation will pause, stop, return to the source anchor point or continue from the target one. An external link leads to a foreign anchor expressed by an URL. The target anchor can be displayed in a new window or replace the current content. When such a link is activated, the current presentation can pause, continue or stop playing.

CHM does not rely only on a link-based model to navigate across independent story fragments as in AHM, but it uses also an event model, presented in 4.7, to trigger navigation actions. Moreover, through event specification, different behaviors can be added for anchor and link activation. For instance, a hotspot can be also used for other visualization needs, for example for displaying a pop-up window on mouse traversal.

4.7. CHM Event Model

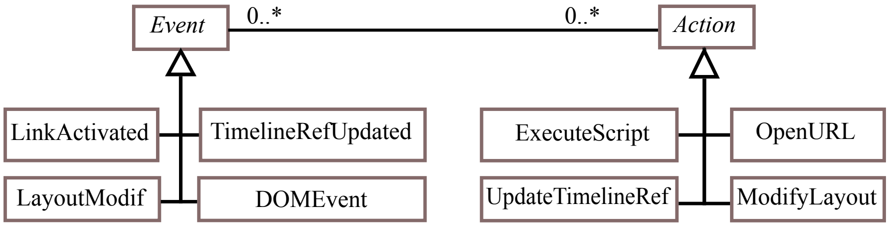

Figure 8. Hypervideo Events and Actions

The dynamic behavior of a CHM hypervideo is handled by an event-based mechanism, expressed by the `Event` and `Action` elements shown in figure 8. Defined as an instantaneous interest occurrence, an event is triggered by a particular environment change at a specific point in space and time. Event listeners are responsible for detecting events occurrence and invoking appropriate procedures called *actions*. An action can be an atomic instruction that acts on the document or a set of operations which may trigger other events and cause further actions.

We have identified some maningful and specific events related to hypervideos: (1) `TimelineRefUpdated` represents a position or status change event in the TLR, for instance from playing to pause, (2) `LayoutModif` occurs when the rendered content is spatially updated, for example when a component reaches its end timecode and disappears or when a component is newly displayed, (3) `LinkActivated` represents the fact that an anchor is activated and (4) `DOMEvent` represents generic components events inside the hypervideo document DOM tree representation. For instance, this may be used to capture the event related to a specific component mouse traversal or focus loss.

Many actions can be associated to an event: (1) `OpenURL` allows the display of a target URL, (2) `ModifyLayout` allows the modification of the displayed content, (3) `UpdateTimelineRef` specifies a state or position change of the timeline reference and (4) `ExecuteScript` allows the execution of a user specified set of operations.

5. WebCHM: a Web-based Implementation of CHM

5.1. Principles

5.1.1. *A declarative and Web-standards compliant approach*

Different programming languages and various architectures can be used to implement CHM. With the prevalence of online video and in order to demonstrate practical uses of the model, we have developed a publicly accessible Web-based implementation that combines video annotations with common Web technologies. Compliance and integration with the existing and widely supported Web technologies being a fundamental part of any successful Web application (Geurts, 2010), we chose to implement WebCHM through a declarative syntax and a set of JavaScript libraries in a Web-standards compliant manner[4].

The rationale behind the declarative approach, as opposed to an imperative one, is to permit authors to create high-level descriptions that explain what should happen rather than how the effects are to be obtained. Such an approach ensures documents readability, re-usability, maintenance and accessibility and is promoted by the W3C within various language standards for complex document manipulations. To extend the browser abilities to support the WebCHM specifications, a set of client-side JavaScript libraries have been developed in order to ensure a seamless support by modern browsers with no need for additional software. The rendering of the audiovisual content is mainly based on the use of the HTML5 `<video>` element and its corresponding DOM API specification[5]. The CHM spatial model is supported by the HTML layout model via cascading stylesheets (CSS) and container elements like `div`.

5.1.2. *Timing and Synchronization*

As with any support for temporal behavior in structured documents, the development of Web-based hypervideos tools brings forth the need for temporal semantics (Jansen et al., 2010).

Many attempts to define online synchronization mechanisms have been proposed, motivated by the time-independent nature of the current Web which hampers the wider use of timed contents. HTML+TIME (Schmitz et al., 1998), SMIL (Bulterman, 2008) and XHTML+SMIL (Schmitz et al., 2002) are examples of such efforts to add timing capabilities to the Web. HTML+TIME was the attempt from Microsoft, Compaq/DEC and Macromedia to integrate SMIL semantics with HTML and CSS. The proposed specification was modified by W3C working groups and emerged as the W3C XHTML+SMIL (Schmitz et al., 2002). SMIL 3 (Bulterman, 2008) allows great complexity and sophistication in time-based scheduling of multimedia components and documents. These different proposed solutions have not been followed by concrete implementations at a large scale. Although HTML5 multimedia

[4] The first version of the proposed language and tools —with code and examples— is available at `http://www.advene.org/chm/`

[5] A Flash fall-back for video management has been implemented for cases when the browser does not support the `<video>` element.

elements implicitly define a temporal scope for media objects, the specification provides a very restricted notion of time that only applies to media and captions (Laiola Guimarães et al., 2010) and not to the document. As there is still no established standard way to add temporal behavior to Web documents, alternatives ways to handle time in the rendered content are required.

To implement the CHM timing specification, we use the W3C SMIL Timesheets 1.0 specification that makes SMIL 3.0 element and attribute timing control available to a wide range of other XML languages, including a-temporal ones such as HTML. Presented as the temporal counterpart of CSS, a Web implementation of the SMIL Timesheets through a JavaScript engine is proposed in the *Timesheet Scheduler* project (Cazenave et al., 2011), whose goal is to rely on SMIL Timing and SMIL Timesheets to synchronize HTML content in a declarative and standards-based way. This specification implementation offers interesting features and is designed to support and take advantage of the HTML5 media elements.

5.1.3. *Annotations*

Annotations can be defined by hand or using dedicated tools and may be provided in a variety of formats. Data readers retrieve the content, parse it and may apply filters to provide appropriate content to components in a unified way. They ensure the presence of the model attributes, especially timecodes for fragment definition. When displayed by a component, the annotation content may be temporalized and thus, scheduled to begin and end with respect to its timecodes in relation with the timeline reference of the container component.

Each annotation implicitly defines a fragment. A fragment is addressed with a syntax that contains the target stream along with begin or begin/end instants. We are using the W3C MediaFragment[6] proposal to address temporal and spatial media fragments. For instance,
http://www.exemple.com/video.mp4#t=.120,1.230
addresses the fragment of the video identified by its begin and end timecodes. When the end instant is not defined, it is supposed to be equal to the duration of the medium.

5.2. WebCHM Syntax

WebCHM introduces a syntax for authoring hypervideos designed as an extension above the HTML language. Unlike XHTML, the HTML specification does not fully support namespace declarations. As XHTML is not as widespread as HTML and since our intention is to allow a larger

[6] http://www.w3.org/2008/WebVideo/Fragments/

use of the language, CHM namespaced attributes are associated to standard HTML elements, and extend them with CHM specific behaviors. Such behaviors are handled by a set of generic and extensible JavaScript libraries that perform content transformation to dynamically generate standard HTML code. Complex hypervideos can therefore be authored as standard Web documents, styled with CSS, extended with SVG and controlled by scripts. Common Web content is written in standard HTML while the hypervideo components are expressed through the CHM attribute-based syntax. The whole syntax and detailed description of the components and their attributes is presented on the project website.

A component declaration requires associating the `chm:component` attribute to a standard HTML container element like `div`. The value of this attribute defines the component type. The available values cover data readers (*annotationreader*, *datareader* and *jsonreader*) and rendering components: simple ones (*textviewer*, *richtextviewer*, *imageviewer*, *audioplayer* and *videoplayer*) and higher level components (*map*, *toc*, *hotspot*, *transcript*, *timeline* and *track*). We also provide a caption component to directly display synchronized content on the player as subtitles. For instance, an image viewer can be declared as follows:

```
<div chm:component="imageviewer".../>
```

Each component is parameterized through a set of specific attributes defined by prefixing their name by `chm:`. Usual HTML attributes like `id`, `title` and `style` can still be used. For instance, the following code snippet declares a text reader with the following attributes: *discours* as a unique identifier, the video player *tm* as a time reference, the *data* annotation reader as data provider, and the *speech* annotation type as filter for content retrieval.

```
<div id='discours'   chm:component='textviewer'
     chm:src='data'  chm:filter="type=='speech'"
     chm:timelineref='tm' />
```

The filter is used for conditional insertion of content. When the reader gets data from the specified source, the filter-provided expression is evaluated against it. The expression maybe a JavaScript assertion or function and data is transmitted to the component only if it satisfies the expression.

5.3. Illustrative examples

To illustrate hypervideo editing with the proposed syntax, we present two examples in the following paragraphs. Both of them make reference to the *Tim Berners-Lee talk at TED 2009* video, annotated using the

Advene application[7], for which six annotation types have been defined (see table II). The annotations have been exported in a *JSon* file called `Tim.json` (see listing 1).

Table II.: Annotations defined for the talk analysis

Type	**Annotation Count**	**Type of content**
Shot	165	Detected shots (Advene video analysis)
Transcript	165	Speech transcription
TranscriptFR	165	French translation of the transcription
Summary	42	Subset of transcript annotations selected for speech summary
Parts	12	Main parts of the speech
Links	7	Cited external resources
Concepts	22	Keywords and concepts of the talk

```
  [...]
{
 "id": "a1072",
 "type": "Transcript",
 "media": "TimBerners-Lee_2009.mp4",
 "begin": 36223,
 "end": 44701,
 "content": "So going back to 1989, I
      wrote a memo suggesting global
      hypertext system."
},
{
  "id": "a1242",
  "type": "Links",
  "media": "TimBerners-Lee_2009.mp4",
  "begin": 36223,
  "end": 44701,
  "content": "text=Tim Berners-Lee's
       1989 memo\n
  url=http://info.cern.ch/Proposal.html
       "
},
{
 "id": "a1341",
 "type": "Summary",
 "media": "TimBerners-Lee_2009.mp4",
 "begin": 0,
 "end": 44701,
 "content": "So going back to 1989, I
      wrote a memo suggesting global
      hypertext
system."
},
{
 "id": "a9",
 "type": "Shot",
 "media": "TimBerners-Lee_2009.mp4",
 "begin": 37708,
 "end": 46291,
 "content": "num=8",
},
{
 "id": "a1073",
 "type": "Transcript",
 "media": "TimBerners-Lee_2009.mp4",
 "begin": 44701,
 "end": 47822,
 "content": "Nobody really did
      anything with it very much."
}
  [...]
```

Listing 1: An excerpt of an annotation structure exported into *JSon* format.

A Basic example: Assume we want to present the video enriched with two common components: a *textual transcription* of the talk speech

[7] See `http://www.advene.org/examples/tbl_linked_data/making_of.html`

displayed over the video player in a synchronous way and a *table of contents* presenting an index of the talk to offer direct access to certain points of the video. Then the CHM logical structure of the hypervideo would be represented as described in figure 9.

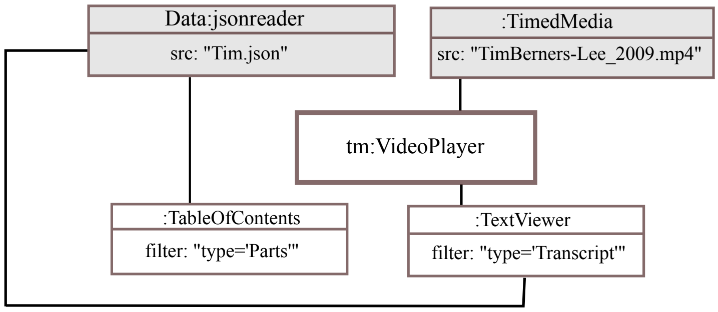

Figure 9. Logical representation of the basic example

Implementing the document with WebCHM syntax involves several stages and declarations:

1. A JSon data reader is required to query the annotation structure described within the `Tim.json` file:

```
<div chm:component="jsonreader" id="data" chm:src="Tim.json"/>
```

2. The video is displayed by the CHM media player parameterized with the URI of the video file:

```
<div chm:component="videoplayer" id="tm"
     chm:src="TimBerners-Lee_2009.mp4"/>
```

3. A caption component is declared by the following code:

```
<div chm:component="textviewer" chm:src="data"
     chm:filter="type=='Transcript'"
     chm:content="$content" chm:timelineref="tm"/>
```

4. The table of contents component is declared as follows:

```
<div chm:component="toc" chm:src="data"
     chm:filter="type=='Parts'" chm:content="$content"
     chm:timelineref="tm" title="Tim Story Parts"/>
```

The HTML document that contains these CHM declarations is associated to a CSS file to describe how it will be rendered. The result is shown in figure 10.

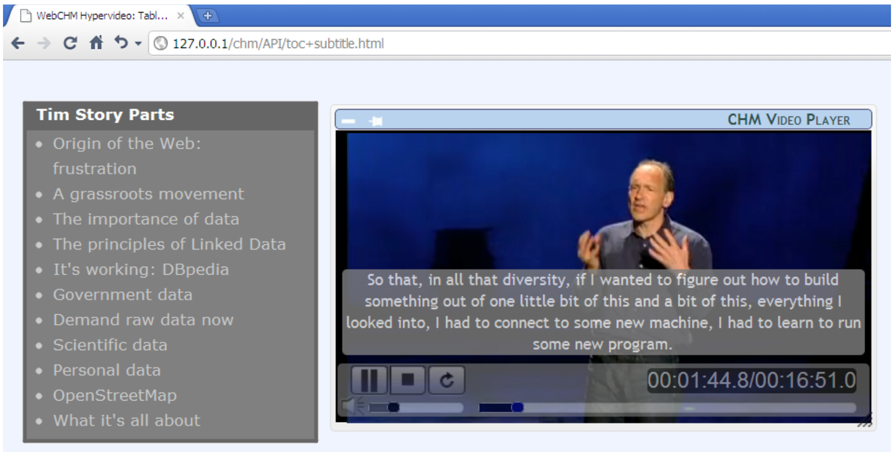

Figure 10. Rendering of the basic example. Captions laid over the *videoplayer* using a *textviewer*. A *table of contents* is shown on the left side.

A more complex hypervideo example: A more complex hypervideo can be created so as to provide more insights into the content of the talk through an augmented and interactive video-based presentation. A formal CHM description of such a hypervideo is described by figure 11. It illustrates the use of many CHM display components: enrichment content viewers, a video player, hotspots, a timeline and a graphical map, to design the rich views of the video.

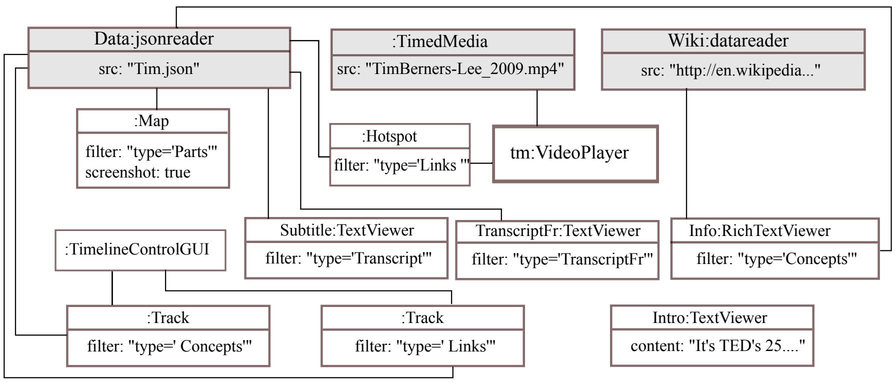

Figure 11. Modeling of the *Talk* hypervideo

The hypervideo uses data provided by the *JSon* package and also from the *Wikipedia.org* encyclopedia, in a video-synchronized way. To retrieve the desired data, two readers are defined, parameterized by a query string: a *jsonreader* —to query movie annotations— along with a generic *annotationreader* to fetch Wikipedia documents. For instance, the display component that synchronizes the display of the definition

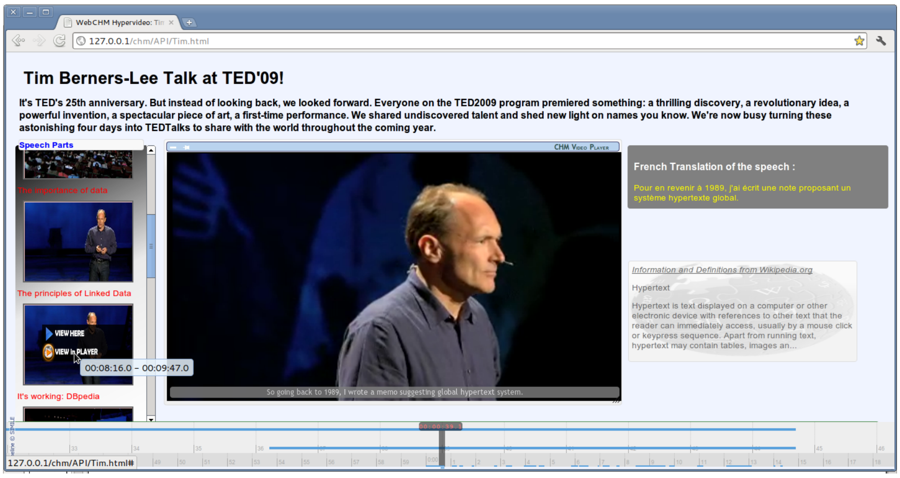

Figure 12. Rendering of a more complex hypervideo. A HTML content is placed on top of the page. Captions, rendered by a *textviewer*, are laid over the *videoplayer*. A graphical *map* is displayed on the left side of the player. Another *textviewer*, shown on the right side, displays the translated content. Below it, data retrieved from the wikipedia is displayed by a *richtextviewer*, and a *timeline* with *tracks* and a slider are displayed. Data is mainly retrieved using a *jsonreader*.

of new terms uses both readers. First, the term and the timecodes corresponding to the interval of introduction and definition by the speaker are retrieved from the *JSon* data structure and the data is used to pull information from *Wikipedia*. The final data is displayed by the *richtextviewer* component according to its specifications in style and time. The authored hypervideo can be seen in figure 12.

In addition to the above-mentioned rich text viewer, and except for the static HTML text used as introduction of the presentation context, the remainder of the content is retrieved exclusively using the *json-reader* component. As shown by both the presentation description and its rendering, a graphical *map* of the scenes is permanently presented on the left side of the page and contains video screenshots. The screen-shots are anchors to play the associated video fragment, either in the autonomous videoplayer or as embedded video replacing the screenshot. The discourse is captioned as subtitle by a *textviewer* at the bottom of the videoplayer, and a french translation of the discourse is rendered at the left side by another *textviewer*. When a web link is provided, a *hotspot* appears over the speaker face to offer an opportunity to display additional information in a separate window. All meaningful events corresponding to new concepts and link possibilities are represented as *tracks* over a *timeline* visible at the bottom of the page.

6. Preliminary Usage Study

6.1. Experimental Setup

Methodology. We have conducted an exploratory user study in order to get a first qualitative evaluation of the relevance of the proposed model and its implementation. We used an *inquiry method* that relies on user-based opinions related to subjective ratings of their impressions and comments (Daniels et al., 2007), through post-test interviews and questionnaire. The main goal of this evaluation is to gather participant comments, concerns and suggestions about the proposed model and the developed authoring and reading features in order to assess the relevance of our proposal and to identify directions for improvements.

Participants. The study was conducted with two user groups (*G1* and *G2*). Since evaluating a theoretical model requires a specialized background and knowledge, the first group *G1* is composed of three multimedia researchers[8]. Since a proven experience in authoring Web documents is needed to evaluate the proposed syntax, we invited for our second group *G2* four web developers who are familiar with rich media integration on websites. For the aspect of reading hypervideo documents designed through WebCHM, we invited both groups to give us their feedback.

Two sessions compose the usage study: the first one addresses the model, while the second one evaluates the prototype. At the end of each session, participants are interviewed and invited to answer corresponding questionnaire. Using five-point Likert rating scales (Likert, 1932), they can rate their appreciation of a set of features that we introduce below.

6.2. Theoretical Session

6.2.1. *Protocol*

While one proof of the quality of a model is the direct implementation of a system through the use of the proposed model (Olivier, 2009), we conducted this study in order to further assess the relevance of the proposed concepts and components. We would also like to determine how the existing components could be extended; looking forward, what new components may enrich the existing set?

While this session is attended by both groups, it mainly targets users of the first group *G1* for the evaluation procedure. The session

[8] from the *Multimedia Systems and Structured Documents* team of CERIST, Algeria

begins with a survey of the hypervideo domain including the main concepts and relevant previous work (models and tools). This allows us to familiarize the participants with hypervideo paradigms and to explain our research motivations.

A presentation of CHM follows. It details all available features. Some modeling examples are presented, including our own samples and some popular hypervideos. After a short discussion, participants are invited to give a sketch of a CHM-based example randomly selected from the presented ones. Then, *G1* users answer the first part of a questionnaire to evaluate the model definition against five main criteria commonly used in model evaluation, identified and discussed in (Olivier, 2009):1/*Simplicity*: are the essential aspects of the model easily understood? 2/*Comprehensiveness*: can the model address all aspects of hypervideo modeling? 3/*Generality*: can the model represent common hypervideo variations, from the plain to the more complex ones? 4/*Exactness*: is CHM accurate in describing all existing and possible aspects of hypervideos? 5/*Clarity*: does the model allow users to clearly understand all relevant model paradigms?

6.2.2. *Results*

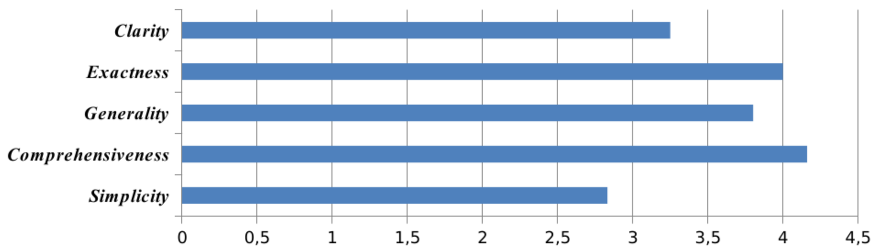

Figure 13. User ratings of the model

Model evaluation requires a wide field and state of the art knowledge, and a good understanding of the theoretical foundations. Hence, for discussing CHM, opinion from *G1* researchers is put forward. Figure 13 gives the average results of the *G1* user ratings of the model.

The field survey performed at the beginning of the evaluation was a great help for participants in expressing their opinion and rating. Once CHM is presented, all of them agree that, while its *comprehensiveness* is appreciated since it apprehends different facets of hypervideos, it has a drawback: it is not very *simple* to understand since many concepts originating from various disciplines are used, ranging from component-based modeling to annotation description, hypermedia navigation and multimedia representation. *Clarity* is another aspect that is underlined:

it is not trivial to correctly capture all the paradigms which have to be well explained. Indeed, although many modeling examples were presented, the participants still needed some assistance when trying to give a formal representation for examples of their choices.

The proposed model is based on the study of common and recognized hypervideo tools and examples. This, combined with the model extensibility, underlines the *generality* of CHM and the participants were pleased to see how some existing hypervideo cases, even the more complicated, could be actually analyzed through CHM. Moreover, they appreciated the modeling *exactness*, expressed in terms of accuracy in hypervideo description. However, the researchers group has noticed some weaknesses in the model, for instance the absolute approach for expressing temporal and spatial positions (via timecodes and coordinates). They suggested to extend it and to add the possibility to express spatial or temporal relations between components. Another suggestion for improving the link model is to take into account the possibility to define non-visual anchors.

6.3. Practical Session

6.3.1. *Protocol*

This part aims to get feedback from users of WebCHM, first as readers and then as creators of hypervideos. Hence, users from the second group *G2* are the main target of hypervideo authoring evaluation while all users (*G1* + *G2*) are concerned when evaluating hypervideo reading. After a short tutorial about the implementation of various CHM components and the proposed syntax, participants practice some illustrative examples and are asked to give their impressions. Then they are invited to read the source code of the examples and asked to create hypervideos of their own, in order to fill the second part of the questionnaire.

Reading User Experience Study. We asked participants to evaluate their user experience when navigating WebCHM hypervideos and to discuss it according to the following aspects: 1/*Utility*: what is the user's feeling about the usefulness of integrating CHM into Web documents? 2/*Aesthetics*: to which extent do WebCHM hypervideos attract users in terms of interfaces and interaction with video-based documents, including the quality of design and the smoothness of the interaction? 3/*Interaction*: what are the benefits, the novelty and the freedom made possible by content structuring and navigation possibilities?

Authoring Usability Study. For our language and framework authoring usability study, we asked the participants to report their impressions according to the main criteria defined by ISO (ISO, 1998) in addition to language accessibility aspects, as suggested in (Barišić et al., 2011) for evaluating usability of domain-specific languages (DSLs): 1/*Effectiveness*: ability and level of completeness to author hypervideo documents using WebCHM; 2/*Efficiency*: level of effectiveness achieved at the expense of various resources, such as mental and physical efforts, measured typically as the amount of time spent to author a hypervideo document; 3/*Satisfaction*: capacity of the language to offer the expected features in hypervideo design; 4/*Accessibility*: learnability and memorability of the syntax and libraries to author hypervideos.

6.3.2. *Results*

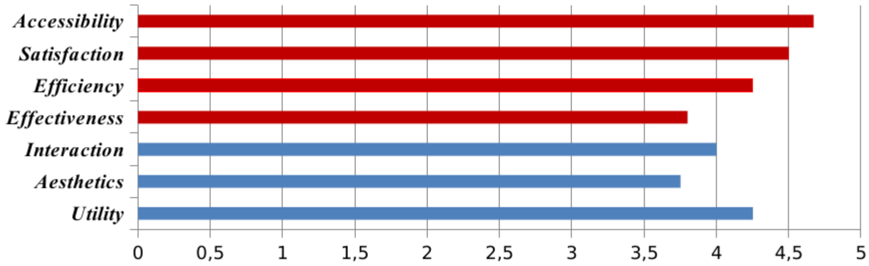

Figure 14. User ratings of User Experience (in blue) and Usability (in red) of WebCHM

Reader Point of View. All our participants (from both groups *G1* and *G2*) were invited to give their opinion about the user experience of reading hypervideos authored with WebCHM. Online hypervideo documents like the ones we designed using WebCHM were a new experience for most of the participants. The feedback was generally positive during the guided exploration phase. Figure 14 illustrates the average results of the user ratings. The *utility* of this integration is highly acknowledged and participants did agree that hypervideo is an effective means to share knowledge associated to video. They were interested in experimenting with new ways of reading and navigating videos, and appreciated WebCHM features in this aspect. They found some possibilities very *useful and attractive*, e.g. describing time-based links, a feature they were not aware of. Indeed, most of their remarks and perplexity were rather about behavior and design of the example hypervideos they experienced than about the principles or the components of CHM. While the *aesthetic* choices were explained, they were differently assessed and participants have expressed some rating hesitations, mainly due to cognitive reasons. Examples of questionable designer choices were the fact that too much data could sometimes be presented simultaneously with possible links and hotspots; the fact that the player automatically stopped after having played a fragment in response to a link activation; or that the timeline did not always present meaningful track labels and figures. It was also suggested that timelines could be customized by the reader.

Developer Point of View. The group of Web developers *G2* is at the core of this evaluation since authoring WebCHM documents still supposes a minimum knowledge of Web technologies, mainly HTML language. However, as illustrated by figure 14, the *accessibility* of the language is the aspect that was the most highly rated by all participants, once the language foundations were explained. Indeed, when looking at the source code of an hypervideo after a transformation, they were positively surprised to understand that, using WebCHM syntax, less than ten lines of code was equivalent to thousands of pure HTML. Moreover, all participants were able to quickly understand, learn and memorize the syntax and the associated requirements after a very short time and no huge coding experience was needed. Globally, the *effectiveness* of CHM-based authoring hypervideos is acknowledged and its capacity to take over the discussed features is a source of *satisfaction*. For instance, participants did appreciate the possibility to add and link supplementary information to video in a time synchronized manner, using a declarative language with no need for additional plugins.

While the *efficiency* of the language is highly rated, since authoring hypervideos becomes as easy as editing any HTML document, the effort associated to create the annotations was underlined. Indeed, while they appreciated the possibility to separate document data from its rendering, the production of annotations using a third party application (in our case, Advene) was found to be a time consuming task.

All of the participants were able to edit and adapt the source code for WebCHM examples. Moreover, they had no difficulty in creating new examples and personalizing them with their own CSS/JavaScript code. However, it appeared that participants had some difficulties to design “good” hypervideos after such a short period, especially regarding temporal issues. For instance they often did not check the annotation timecodes, causing metadata and links to be displayed for a too short time; or they sometimes instantiated too many components at the same time causing cognitive overload for the reader. This led the group to suggest 1/ the integration of the framework in a system that may help them to control such aspects; 2/ the development of a dedicated GUI or at least a plugin that would add basic syntax and library support to existing Web development systems (like Adobe Dreamweaver). When asked to suggest new components to complete the existing set of high level ones, the participants did point out the need for communication and collaboration tools; for instance, offering means to rate and comment a video and to share resulting data as annotations; from a higher-level perspective, offering to end-users means to annotate/edit videos were mentioned as interesting features.

6.4. Summary

This preliminary study validated some aspects of our model and implementation. The importance and usefulness of hypervideo for rich internet application was recognized. The component-based approach to hypervideo was clearly understood, as well as the fact that providing high-level components can ease the design and implementation of hypervideos. On the implementation side (WebCHM), while the syntax was comprehensible, participants suggested that convenience tools would be appreciated to ease coding and check some usability aspects of the resulting hypervideos.

This evaluation targeted early users with well-targeted profiles and who may embrace the concepts and the WebCHM methodology. The next step of this evaluation could consist in presenting our language and framework to a more general audience.

7. Discussion

Comparison with other models. Hypervideo being a specialization of hypermedia, general conceptual models like Dexter, AHM and NCM can be used for abstract representation. These models, mainly AHM and Dexter, were designed to cover all relevant theoretical concepts of hypertext and hypermedia fields. However, some authoring and reading concerns associated to hypervideos are different compared to conventional hypermedia ones. Being a time based multimedia phenomenon in its own right, video poses a significant challenge for hypermedia design, not yet fully addressed (Chambel and Guimaraes, 2002); for instance and differently from classic hypermedia systems, hypervideo links are not always static since they may appear and disappear with time as the video streams play out (Mujacic et al., 2012). Actually, the continuous nature of video results in an information distribution over space and time, making data seizing, access and processing a complex task. A hypermedia application with the inclusion of dynamic media, like hypervideo, presents new challenges and requires new production models (Morales, 2001). Consequently, as stated in (Hammoud, 2006), discussing hypervideos only from hypermedia/multimedia perspectives results in inefficiency and weakness to express their logical patterns since the underlying concepts are too general to grasp and characterize hypervideo details.

Regarding model evaluation parameters proposed by (Olivier, 2009), *simplicity* and *clarity* when describing hypervideos are the main issues of hypermedia models. Moreover, the *comprehensiveness* and the *exactness* of these models in describing general hypermedia have the drawback that they do not provide explicit concepts to address some features of hypervideos, from a semantic structuring, linking and reading point of view. Since the suitable integration of interactive video requires more specialized hypermedia models (Memon and Khoja, 2003), hypervideo logical architecture and specific features need dedicated and robust document models capable to encompass the limitations associated to general hypermedia/multimedia modeling.

CHM addresses these issues by featuring a high-level, hypervideo-dedicated, component-based approach. It can of course be expressed using general hypermedia reference models like Dexter and AHM. For instance, CHM components which consist in content, presentation specification, attributes and anchors can be represented by atomic or composite AHM components, with temporal or atemporal composition. CHM also encapsulates the AHM notion of channel used for the same purposes, along with attributes of the rendering components. Moreover, as proposed in AHM (and SMIL), CHM explicitly separates the spatial layout from the temporal specifications.

Key points of CHM. CHM strength resides in its high level of abstraction, aiming at expressiveness. Indeed, with the CHM high level concepts, hypervideo definition and implementation are quite straightforward. The proposed modeling choices try to introduce intuitive concepts, based on the analysis of a number of existing hypervideo systems. As exemplified by the Web implementation and the illustrative examples, complex hypervideo documents can be conceptualized and coded rather easily and efficiently, alleviating the need for low-level constructs.

The annotation-driven approach defines various access granularities to video, by allowing fragment definition with no length or coverage restriction. Fragments can be arbitrarily overlapped or nested, associating various data and multiple views to the same audiovisual information unit. By separating data from content presentation, managing, maintaining and exchanging the document structure is eased, independently from the audiovisual stream. The annotation structure definition in CHM is system-independent although a minimal schema is provided. Its realization may be achieved by any annotation standard such as MPEG-7 or a user-defined structure. In our implementation, we have used the Advene annotation tool and schema, exposing annotation data as a *JSon* structure.

Implementation. So as to practically evaluate our claims, CHM has been implemented by grafting hypervideo specific concepts and attributes in a declarative manner to the HTML language. This makes authoring such documents more convenient compared to existing systems, thanks to the Web standards-compliant syntax: writing a hypervideo document is as easy as editing any conventional Web page.

The identification of common very-high level components such as the transcript or the timeline particularly improves the conciseness of the model. Moreover, the declarative syntax allows describing what is expected rather then coding the desired behavior, which is a tedious task. However, as shown by our preliminary user study, the components we propose can still be further refined, and automated checking would be useful when it comes to time-based rendering issues.

SMIL and NCL, based respectively on AHM and NCM, offer robust and comprehensive languages and tools for multimedia/hypermedia documents composition. While NCL still exposes elementary hypermedia components and does not provide high level hypervideo ones, SMIL is a complex language and the specification itself contains no formal model definition (Jansen et al., 2010). Moreover, as stated in (Deltour and Roisin, 2006), direct manipulation of such languages is too complex for most users, requiring a deep understanding of the languages semantics. As a consequence, authoring hypervideo documents with the very general concepts and their huge language specification is an arduous task, on the contrary to the CHM/WebCHM proposal.

The *Popcorn.js* approach (that uses a technology driven strategy) is close to ours in its goals but it diverges on some key points: 1/ it is JavaScript-based, thus imperative, while we adopted a declarative approach (that can be customized through JavaScript if needed) which has more convenient properties for edition and introspection; 2/ metadata is expressed as JavaScript objects, and strongly tied to the targeted visualizations – we promote a stronger decoupling between metadata and its visualizations; 3/ its development process is based on quick iterative cycles, starting from a very small core and principles, while we propose a more targeted process; 4/ like most online video projects, *popcorn.js* uses exclusively the time inferred from the video stream to schedule the presentation. This weakness still exists in the latest HTML specification. In our prototype, the timing mechanisms allow hypervideo to be fully Web-based multimedia documents, thanks to the temporal model implemented through the SMIL Timesheets JavaScript library.

8. Conclusion and Future Work

Motivations and Contributions. Hypervideo —hypermedia with a focus on video— constitutes an opportunity to experiment with innovative modalities in editing and presenting audiovisual material with more elaborated features and interactivity. Common hypermedia models do not precisely address hypervideo specificities, while their genericity makes the modeling of hypervideo documents cumbersome. This motivated our model and implementation proposals. They aim at defining a a general framework to try out original experimentations with more advanced usage forms and interaction modalities for video-based knowledge communication. Our main contributions are:

1. CHM, a high level operational model suitable for the design of annotation-based hypervideos. CHM promotes a clear separation of data from the specification of content through the use of annotations and high-level rendering components. By emphasizing the importance of the annotation structure and the variety of renderings, CHM offers a new point of view on video-centered hypermedia systems. The high level components we identified are commonly needed in hypervideos, and their presence in the model allows concise modeling of hypervideos;

2. a Web implementation of the proposed model that relies on a declarative syntax within HTML elements, associated to a javascript library for hypervideo rendering. The WebCHM framework illustrates practically how the higher-level instrumentation of Web technologies can reduce the accessibility barrier of hypervideo design on the Web. Our research paves the way to the development of standard components and tools for building annotation-based hypervideos.

Perspectives. Current work focuses on implementing the remainder of the CHM model in WebCHM and putting WebCHM in use on more use cases, allowing us to refine the specification of the high-level components of CHM and to carry a more ambitious user-study. We expect more use cases to enhance the offered tools and to help to investigate more advanced interaction and visualization approaches and techniques for better Web oriented video consumption. Future works include the development of a graphical user interface (GUI) environment and its integration in the Advene application. We also want to rely on cognitive theories to propose more meaningful rendering components and libraries and other interaction possibilities like supporting interactive annotation and hypervideo generation.

Acknowledgments

This work has been partially funded by the French FUI (Fonds Unique Interministériel) CineCast project. It is supported by the Algerian Research Center on Scientific and Technical Information (CERIST).

References

Alisi, T., M. Bertini, G. D'Amico, A. Del Bimbo, A. Ferracani, F. Pernici, and G. Serra: 2009, 'Arneb: a rich internet application for ground truth annotation of videos'. In: *Proceedings of the 17th ACM international conference on Multimedia*. New York, NY, USA, pp. 965–966, ACM.

America, M. E.: 2000, 'Authoring with hyperlinked video'. Technical report, Mitsubishi Electric America.

Aubert, O. and Y. Prié: 2005, 'Advene: active reading through hypervideo'. In: *Proceedings of the 16th ACM conference on Hypertext and hypermedia*. New York, NY, USA, pp. 235–244, ACM.

Barišić, A., V. Amaral, M. Goulão, and B. Barroca: 2011, 'Quality in use of domain-specific languages: a case study'. In: *Proceedings of the 3rd ACM SIGPLAN workshop on Evaluation and usability of programming languages and tools*. New York, NY, USA, pp. 65–72, ACM.

Bove, M., J. Dakss, S. Agamanolis, and E. Chalom: 2000, 'Hyperlinked television research at the MIT Media Laboratory'. *IBM Syst. J.* **39**, 470–478.

Bulterman, D.: 2008, 'Synchronized Multimedia Integration Language (SMIL 3.0)'. W3C recommendation, W3C. http://www.w3.org/TR/SMIL3/.

Bulterman, D. C. A. and L. Hardman: 2005, 'Structured multimedia authoring'. *ACM Trans. Multimedia Comput. Commun. Appl.* **1**, 89–109.

Casanova, M. A., L. Tucherman, M. J. D. Lima, J. L. Rangel Netto, N. Rodriquez, and L. F. G. Soares: 1991, 'The nested context model for hyperdocuments'. In: *Proceedings of the third annual ACM conference on Hypertext*. New York, NY, USA, pp. 193–201, ACM.

Cazenave, F., V. Quint, and C. Roisin: 2011, 'Timesheets.js: When SMIL Meets HTML5 and CSS3'. In: *Proceedings of the 11th ACM symposium on Document engineering*. New York, NY, USA, pp. 43–52, ACM.

Chambel, T., M. N. Chhaganlal, and L. A. R. Neng: 2011, 'Towards immersive interactive video through 360° hypervideo'. In: *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*. New York, NY, USA, pp. 78:1–78:2, ACM.

Chambel, T. and N. Guimaraes: 2002, 'Context perception in video-based hypermedia spaces'. In: *Proceedings of the thirteenth conference on Hypertext and hypermedia*. College Park, Maryland, USA, pp. 85–94.

Chambel, T., C. Zahn, and M. Finke: 2004, 'Hypervideo Design and Support for Contextualized Learning'. In: *IEEE International Conference on Advanced Learning Technologies (ICALT'04)*. Joensuu, Finland, pp. 345–349.

Daniels, J., S. Fels, A. Kushniruk, J. Lim, and J. Ansermino: 2007, 'A framework for evaluating usability of clinical monitoring technology'. *Journal of Clinical Monitoring and Computing* **21**, 323–330. 10.1007/s10877-007-9091-y.

Davis, M., S. King, N. Good, and R. Sarvas: 2004, 'From context to content: leveraging context to infer media metadata'. In: *Proceedings of the 12th annual*

ACM international conference on Multimedia. New York, NY, USA, pp. 188–195, ACM.

Deltour, R. and C. Roisin: 2006, 'The LimSee3 multimedia authoring model'. In: *Proceedings of the 2006 ACM symposium on Document engineering*. pp. 173–175.

Fagá, R., V. G. Motti, R. G. Cattelan, C. A. C. Teixeira, and M. da Graça Campos Pimentel: 2010, 'A social approach to authoring media annotations'. In: *10th ACM symposium on Document engineering*. pp. 17–26.

Finke, M.: 2004, 'Mobile interactive video'. Technical report, IST-2001-37365, MUMMY Project.

Geurts, J. P. T. M.: 2010, 'A Document Engineering Model And Processing Framework For Multimedia Documents'. Ph.D. thesis, Technische Universiteit Eindhoven.

Girgensohn, A., L. Wilcox, F. Shipman, and S. Bly: 2004, 'Designing Affordances for the Navigation of Detail-on-Demand Hypervideo'. In: *Proceedings of ACM Advanced Visual Interfaces*. pp. 290–297.

Halasz, F. and M. Schwartz: 1994, 'The Dexter Hypertext Reference Model'. *Communications of the ACM* **37**(2), 30–39.

Hammoud, R. I.: 2006, 'Introduction to Interactive Video'. In: *Interactive Video*. Springer-Verlag, pp. 3–24.

Hardman, L., D. C. A. Bulterman, and G. van Rossum: 1994, 'The Amsterdam Hypermedia Model: adding time and context to the Dexter model'. *Commun. ACM* **37**, 50–62.

Hickson, I.: 2011, 'HTML5'. http://www.w3.org/TR/html5/.

Hoffmann, P., T. Kochems, and M. Herczeg: 2008, 'HyLive: Hypervideo-Authoring for Live Television'. In: *Proceedings of the 6th European conference on Changing Television Environments*. Berlin, Heidelberg, pp. 51–60, Springer-Verlag.

InnoTeamS GmbH: 2012, 'ADIVI - Hypervideo für Lernen und Support'. Website `http://www.adivi.net/` accessed 2012/01/30.

ISO: 1998, 'Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 11: Guidance on Usability'. ISO Standard.

Jansen, J. and D. C. Bulterman: 2009, 'SMIL State: an architecture and implementation for adaptive time-based web applications'. *Multimedia Tools Appl.* **43**, 203–224.

Jansen, J., P. Cesar, and D. C. Bulterman: 2010, 'A model for editing operations on active temporal multimedia documents'. In: *10th ACM symposium on Document engineering*. pp. 87–96.

Kipp, M.: 2001, 'ANVIL - A Generic Annotation Tool for Multimodal Dialogue'. In: *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*. Aalborg, pp. 1367–1370.

Kokkoras, F., H. Jiang, I. Vlahavas, A. K. Elmagarmid, E. N. Houstis, and W. G. Aref: 2002, 'Smart videotext: a video data model based on conceptual graphs'. *Multimedia Syst.* **8**, 328–338.

Laiola Guimarães, R., P. Cesar, and D. C. Bulterman: 2010, 'Creating and sharing personalized time-based annotations of videos on the web'. In: *10th ACM symposium on Document engineering*. pp. 27–36.

Li, J., Y. Ding, Y. Shi, and J. Zhang: 2010, 'Building a Large Annotation Ontology for Movie Video Retrieval'. *JDCTA* **4**(5), 74–81.

Likert, R.: 1932, 'A technique for the measurement of attitudes'. *Archives of Psychology* **22**(140), 5–55.

Lippman, A.: 1980, 'Movie-maps: An application of the optical videodisc to computer graphics'. In: *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. New York, NY, USA, pp. 32–42, ACM.

Meixner, B., B. Siegel, G. Hölbling, F. Lehner, and H. Kosch: 2010, 'SIVA suite: authoring system and player for interactive non-linear videos'. In: *Proceedings of the international conference on Multimedia*. New York, NY, USA, pp. 1563–1566, ACM.

Memon, Q. A. and S. A. Khoja: 2003, 'Object-Database Approach for a Hypervideo Platform'. *Malaysian Journal of Computer Science* **16**(2), 73–80.

Miller, G., S. Fels, A. Al Hajri, M. Ilich, Z. Foley-Fisher, M. Fernandez, and D. Jang: 2011, 'MediaDiver: viewing and annotating multi-view video'. In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*. New York, NY, USA, pp. 1141–1146, ACM.

Morales, M.: 2001, 'Hypervideo as a Tool for Communicating Mathematics'. Master's thesis, European Master in Multimedia and Audiovisual Business Administration, Haute Ecole Groupe ICHEC ISC ST-Saint-Louis-ISFSC.

Mozilla: 2012a, 'Popcorn.js The HTML5 Media Framework'. Website `http://www.popcornjs.org/` accessed 2012/01/30.

Mozilla: 2012b, 'Web Made Movies'. Retrieved from `https://www.drumbeat.org/en-US/projects/webmademovies/` on 2012/01/30.

Mujacic, S., M. Debevc, P. Kosec, M. Bloice, and A. Holzinger: 2012, 'Modeling, design, development and evaluation of a hypervideo presentation for digital systems teaching and learning'. *Multimedia Tools Appl.* **58**(2), 435–452.

Navarrete, T. and J. Blat: 2002, 'VideoGIS: Segmenting and indexing video based on geographic information'. In: *5th AGILE Conference on Geographic Information Science*. pp. 1–9.

Nelson, T. H.: 1965, 'Complex information processing: a file structure for the complex, the changing and the indeterminate'. In: *Proceedings of the 1965 20th national conference*. New York, NY, USA, pp. 84–100, ACM.

Nelson, T. H.: 1987, *Computer Lib/Dream Machines*. Redmond, WA, USA: Microsoft Press.

Olivier, M. S.: 2009, *Information Technology Research — A Practical Guide for Computer Science and Informatics*. Pretoria, South Africa: Van Schaik, 2nd edition.

Pollone, M., M. Rusconi, and R. Tua: 2002, 'From hyper-film to hyper-web: The challenging continuation of a european project'. In: *Electronic Imaging and the Visual Arts Conference*. Florence, Italy.

Roisin, C.: 1998, 'Authoring Structured Multimedia Documents'. In: *Proceedings of the 25th Conference on Current Trends in Theory and Practice of Informatics: Theory and Practice of Informatics*. London, UK, UK, pp. 222–239, Springer-Verlag.

Sadallah, M., O. Aubert, and Y. Prié: 2011, 'Component-based Hypervideo Model: high-level operational specification of hypervideos'. In: ACM (ed.): *Document Engineering 2011 (DocEng 2011)*. pp. 53–56.

Sawhney, N. N., D. Balcom, and I. E. Smith: 1996, 'HyperCafe: Narrative and Aesthetic Properties of Hypervideo'. In: *UK Conference on Hypertext*. Bethesda, Maryland, United States, pp. 1–10.

Schmitz, P., A. Patterson, and D. Newman: 2002, 'XHTML+SMIL Profile'. W3C note, W3C. `http://www.w3.org/TR/XHTMLplusSMIL/` accessed 2012/01/30.

Schmitz, P., J. Yu, and P. Santangeli: 1998, 'Timed Interactive Multimedia Extensions for HTML'. W3C note, W3C. http://www.w3.org/TR/NOTE-HTMLplusTIME accessed 2012/01/30.

Shipman, F., A. Girgensohn, and L. Wilcox: 2003a, 'Combining spatial and navigational structure in the hyper-hitchcock hypervideo editor'. In: *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia*. pp. 124–125.

Shipman, F., A. Girgensohn, and L. Wilcox: 2003b, 'Generation of interactive multi-level video summaries'. In: *Proceedings of the eleventh ACM international conference on Multimedia*. New York, NY, USA, pp. 392–401, ACM.

Silva, H. V. O., R. F. Rodrigues, L. F. G. Soares, and D. C. Muchaluat Saade: 2004, 'NCL 2.0: integrating new concepts to XML modular languages'. In: *Proceedings of the 2004 ACM symposium on Document engineering*. New York, NY, USA, pp. 188–197, ACM.

Soares, L. F. G. and R. F. Rodrigues: 2005, 'Nested Context Model 3.0 Part 1 - NCM Core'. Technical report, Pontifícia Universidade Católica do Rio de Janeiro.

Tiellet, C. A., A. G. Pereira, E. B. Reategui, J. V. Lima, and T. Chambel: 2010, 'Design and evaluation of a hypervideo environment to support veterinary surgery learning'. In: *21st ACM conference on Hypertext and hypermedia*. pp. 213–222, ACM.

Vendrig, J. and M. Worring: 2003, 'Interactive Adaptive Movie Annotation'. *IEEE MultiMedia* **10**, 30–37.

VideoClix.tv: 2012, 'Clickable video. Interactive video advertising.'. Website http://www.videoclix.net/ accessed 2012/01/30.

YouTube: 2009, 'Getting started: Creating or editing annotations'. Website http://www.google.com/support/youtube/bin/answer.py?hl=en&answer=92710 accessed 2012/01/30.

Zahn, C., B. Barquero, and S. Schwan: 2004, 'Learning with hyperlinked videos - design criteria and efficient strategies for using audiovisual hypermedia'. *Learning and Instruction* **14**, 275–291.

Zoller, P.: 2001, 'HMT: Modeling Interactive and Adaptive Hypermedia Applications'. In: *Information Modeling in the New Millennium*. Springer-Verlag, pp. 383–405.