Creating and Sharing Personalized Time-Based Annotations of Videos on the Web

Rodrigo Laiola Guimarães, Pablo Cesar and Dick C. A. Bulterman
CWI: Centrum Wiskunde & Informatica
Science Park 123
1098 XG Amsterdam, The Netherlands
+31 20 592 93 33

{rlaiola, p.s.cesar, dick.bulterman}@cwi.nl

ABSTRACT

This paper introduces a multimedia document model that can structure community comments about media. In particular, we describe a set of temporal transformations for multimedia documents that allow end-users to create and share personalized timed-text comments on third party videos. The benefit over current approaches lays in the usage of a rich captioning format that is not embedded into a specific video encoding format. Using as example a Web-based video annotation tool, this paper describes the possibility of merging video clips from different video providers into a logical unit to be captioned, and tailoring the annotations to specific friends or family members. In addition, the described transformations allow for selective viewing and navigation through temporal links, based on end-users' comments. We also report on a predictive timing model for synchronizing unstructured comments with specific events within a video(s). The contributions described in this paper bring significant implications to be considered in the analysis of rich media social networking sites and the design of next generation video annotation tools.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentations**]: Multimedia Information Systems - *Audio, Video.* I.7.2 [**Document and Text Processing**]: Document Preparation - *Format and notation, hypertext/hypermedia, Languages and Systems, Multi/mixed media.*

General Terms

Algorithms, Documentation, Design, Experimentation, Human Factors, Standardization, Languages.

Keywords

Timed end-user comments, Video annotation tools, Document transformations, Temporal hyperlinks, SmilText.

1. INTRODUCTION

Successful commercial video sharing systems have provided ample proof that video is a first-class Web object. In these sharing systems, video content serves both as a means of communicating a simple or complex story (using implicit or explicit cinematic rules) and as a catalyst for communication among third-party viewers of that content [4][7].

Recent developments by video service providers also have extended the means for third-party communication in ways that have never been possible with conventional broadcast or personal video systems. Consider the upper fragment of a YouTube[1] page shown in Figure 1. Here, we see the video title, the base video content, two overlay hyperlinks to external videos (to CCTV video #1 and #2), and a pop-up annotation directing the user to scroll two minutes ahead into the video. There is also a branding icon (in this case, to AIRBOYD.tv) and a set of labels that were 'burned' into the base video content. From an end-user's point of view, these are all producer-provided navigation and comments: they cannot be modified (beyond conditional display) without altering the source video content.

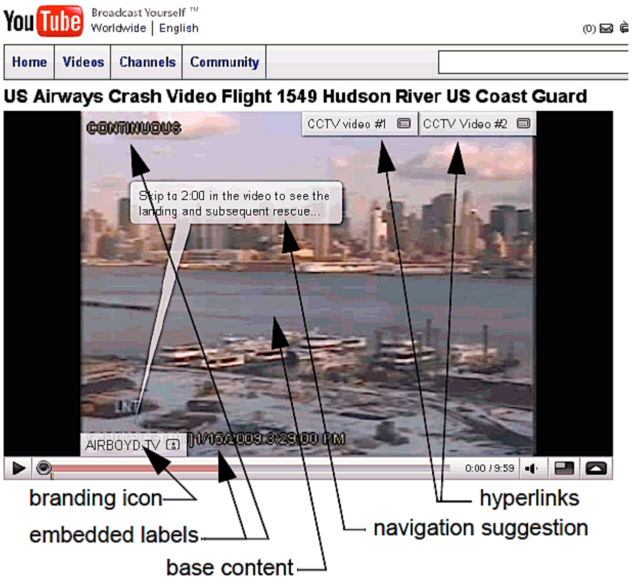
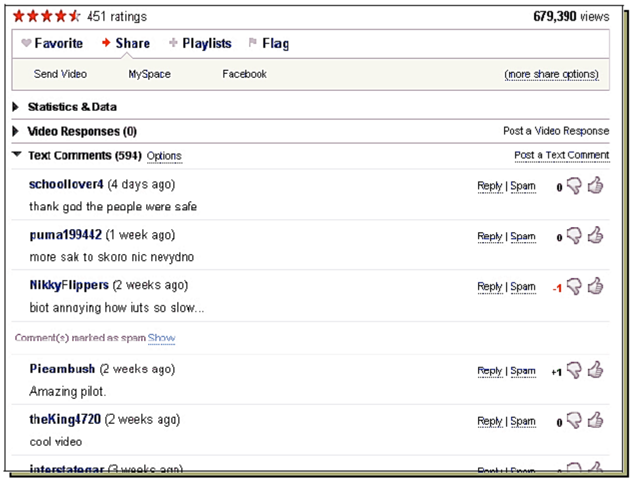


Figure 1. A typical YouTube annotated video.

[1] http://www.youtube.com



schoollover4 (4 days ago) 0

thank god the people were safe

puma199442 (1 week ago) 0

more sak to skoro nic nevydno

NikkyFlippers (2 weeks ago) -1

biot annoying how iuts so slow...

Pieambush (2 weeks ago) +1

Amazing pilot.

theKing4720 (2 weeks ago) 0

cool video

Figure 2. A typical YouTube set of end-user comments.

In addition to the base video, a typical YouTube page also provides space for end-user generated comments. An example of these is shown in Figure 2. End-user comments include implicit forms of commentary (such as anonymous ratings and number of views), and groups of explicit comments from interpreted viewers. In general, end-users can provide only a-temporal text comments, unless they have been given editing rights to the base video.

The primary contribution of this work is the identification and description of a set of transformations to multimedia documents that allow end-users to enrich third-party video content. Our solution, unlike current approaches, allows end-users to create styled and time-based text annotations on media. It also permits end-users to identify temporal navigation points by using hyperlinks within the comments, and to associate contextual timed metadata (e.g., who has made the comment and when) to the comments. In this work we focus on text, but a similar approach is valid to audio and image annotations as well. The benefit over current approaches lays in the usage of a rich captioning format that is not embedded into a specific video encoding format. As the comments are not stored within any of the videos but kept as separate structured documents, they respect the rights of content owners and they can be shared, modified and analyzed independently. The contributions of this paper are validated by the design and implementation of a next-generation Web-based video annotation tool, which also provides a predictive timing algorithm for temporal alignment of the comments with the base-content.

In particular, the requirements and constraints that motivated this work include:

i. *Retain base video integrity:* end-users should not be able to alter the base video content, either in terms of adding embedded captions/comments or providing visual overlays on the base content — this right is reserved for the content owner;

ii. *Allow multiple-video aggregation:* the captions and comments created by an application should be able to span multiple videos that are played as a continuous playlist;

iii. *Allow multiple-provider integration:* the end-user should not be locked into a single video service provider for candidate content, but should be able to populate the playlist from a diverse collection of content libraries;

iv. *Allow timed end-user captions and comments:* end-users must be able to add captions and comments that are visible only when particular fragments of the video are being shown, but which are not embedded in the source file of the video;

v. *Allow micro-personalized time-based annotations*: end-users should be able to create different sets of time-based captions/comments for individual users/communities, or share these as 'broadcast' comments (similar to existing approaches in YouTube and similar systems);

vi. *Allow selective end-user viewing*: end-users might be able to select and watch comments by specific individuals and/or user communities, by topic etc.; and

vii. *Allow timed end-user navigation*: end-user's comments should be able to include direct navigation support via timed anchors in the text content. This will allow others to navigate other interesting content in the same collection or to link to external media.

This paper is structured as follows. First, an overview of related work is provided in Section 2. Section 3 describes our contribution in terms of multimedia document transformations. Next, Section 4 validates the contribution based on the design and implementation of a Web-based video annotation tool that meet the requirements. Lastly, Section 5 discusses the results and reviews the contributions of our work.

2. CAPTIONS ON THE WEB

End-user a-temporal comments can provide valuable information for understanding media. There has been previous work that analyzes end-user comments and discussions on Web-based video interfaces. For example, investigations on the influence of usage patterns on the social conversational consequences [4]. Our work differs not only in focus, but also by the fact that we propose a solution for end-user time-based annotations.

Shamma et al. [14] investigated microblogging around live media events. Results indicated that the level of Twitter[2] activity during the event and conversational cues can reflect the topics of discussion in the live event. While twittering reactions to live broadcast media can yield significant insights into the semantic and temporal structure of media, the same methodology cannot be applied for obtaining temporal cues within a video hosted in a rich media social networking site such as YouTube.

Regarding applications, a number of research efforts have addressed end-user annotations and sharing of video content [6][13]. Most closely related to our work, Nathan et al. [11] presented a system that allows viewers to create text comments while watching a TV show. In an asynchronous scenario, previously generated comments (temporally-linked to the media stream) are shown to later viewers as they watch that program. Our work is related, but we focus on time-based text annotations of videos on the Web domain.

The rest of this section reviews representative document models in the context of our work. Based on our experience Table 1

[2] http://twitter.com

Table 1. Comparison of captioning approaches from the end-user perspective.

	Retain Integrity	Span Multiple Videos	Multiple Sources	Timed End-User Comments	Targeted Comments	Selective Viewing	Linking and Navigation
YouTube	++	--	--	+/-	--	-	+/-
HTML5	++	--	+	+/-	--	-	--
NCL	++	++	++	+	++	++	+
SMIL	++	++	++	++	++	++	++

Strong Support: ++; Basic Support: +; Weak Support: -; No Support: --.

provides a summary of how each approach supports the needs for generating flexible end-user comments, as defined in Section 1.

2.1 Supporting Captions on YouTube

As shown in Figure 2, YouTube only supports a-temporal text comments (e.g. not synchronized within the video material). Nevertheless, richer authoring capabilities have been integrated over the last couple of years. In terms of captioning, YouTube enables users to add closed captions to their own Flash videos [3] by uploading a caption file generated elsewhere[3,4] (but only the SubViewer[5] and SubRip[6] formats are supported).

YouTube Annotations — not to be confused with YouTube Captions - allows users to add notes and links to their owned videos. YouTube Annotations also gives users the ability to invite friends to help annotating their videos, though the original video owner will retain final control over what appears (the video owner may remove third-party annotations or even reset the annotation interface access link).

Our work differs not only in the underlying technology (YouTube uses its own annotation format), but also in the application of the technology. For example, comments created by YouTube cannot be exported (implementation restriction), other users than the owner cannot create captions unless invited, and the end-user cannot add metadata temporally associated with the comments. We believe that our solution indeed allows any end-user to create, micro-personalize and share time-based text annotations in any third-party video.

2.2 Supporting Captions in HTML5

HTML[7] (*HyperText Markup Language*) is the predominant markup language for Web pages. It provides the means to describe the structure of text-based information in a Web document — by denoting, for example, certain text as links and paragraphs — and to supplement that text with interactive forms, embedded images, and other objects. HTML can also describe, to certain degree, the appearance and semantics of a document, and can include embedded scripting language code (such as JavaScript[8]), which can affect the behavior of Web browsers and other HTML processors.

One of the innovative features of HTML5 is the introduction of the `<audio>` and `<video>` elements as first-class citizens of the HTML language. While the addition of these elements may at first glance appear as simple extensions of the existing media types offered by HTML (such as `<img>` and embedded text), each of these elements has grown to play several roles within a document.

The `<audio>` and `<video>` elements implicitly define a temporal scope for the referenced media object. However, HTML5 provides a very restricted scope of time that only applies to the video and the captions (if any). Currently, HTML5 does not support embedded captions, and the intention is to support only SRT. The `<video>` element also provides a layout restriction in that it (and not the captions) defines the space available for rendering caption overlays.

Note that, in HTML5, providing a new functionality (such as specifying the conditional rendering of one or more members of a set of captions) requires that this functionality be shoehorned into the existing limited syntax. This is unfortunate, and largely unnecessary because better declarative structuring alternatives are already widely available.

2.3 Supporting Captions in NCL

NCL (*Nested Context Language*) [18] is the standard XML-based application language for defining interactive multimedia presentations in the Brazilian Terrestrial Digital TV System (SBTVD-T). In NCL authors can take advantage of its high-level constructs to describe, in a declarative manner, the temporal behavior of a multimedia presentation. Authors can as well associate hyperlinks with media objects, define alternatives for presentation, and describe the layout of the presentation on multiple devices. Moreover, NCL provides support for imperative scripts in order to enhance its computational power [17].

Unlike HTML, NCL has a strict separation between the document's (or application's) content and structure, and it provides non-invasive control of presentation linking and layout. This means that NCL can be used to render videos in the context of a general presentation, and to control the timing and rendering properties of external caption content (HTML or SRT), while this is being displayed.

[3] Subtitle Horse. http://subtitle-horse.org

[4] dotSUB. http://dotsub.com

[5] Wikipedia, SubViewer. http://en.wikipedia.org/wiki/SubViewer

[6] Wikipedia, SubRip. http://en.wikipedia.org/wiki/SubRip

[7] http://www.w3.org/TR/html

[8] https://developer.mozilla.org/en/JavaScript

Regarding the requirements described in Section 1, NCL does not specify an encoding format for the captions themselves, making harder not only the support for basic timing, but also for embedded links in the text, styling and Meta information that can be used to structure information within the caption contents.

2.4 Supporting Captions in SMIL

SMIL – the Synchronized Multimedia Integration Language – is the main multimedia container format supported by W3C, the World Wide Web Consortium[9].

Like NCL, SMIL also is an integration format, and as such, it does not directly define media objects (with the exception of timed text content). Instead, SMIL acts as a container format in which spatial, temporal, linking and interactive activation primitives can be used to place, schedule and control a wide assortment of media objects.

SmilText is an embedded text format for use within SMIL 3.0. This format has several features. First of all, it is possible to take a smilText element and to transform its content directly to existing external formats [5]; this allows the captions to be processed separately by custom tools if necessary. SmilText also balances the need for text styling with the requirement for an efficient representation that can be easily parsed and scheduled at runtime.

While the SmilText format was developed as an embedded text structuring language, it is also possible to use it as an external container format. In this case, the SmilText file will contain intra-block formatting and timing control, with layout and general rendering control defined in SMIL. The primary advantage of using SmilText as an external format is that the text content can be bound to the presentation at document run-time, rather than at document authoring time. The text can be automatically generated based on information on the presentation user, or it can be dynamically updated from a streaming source.

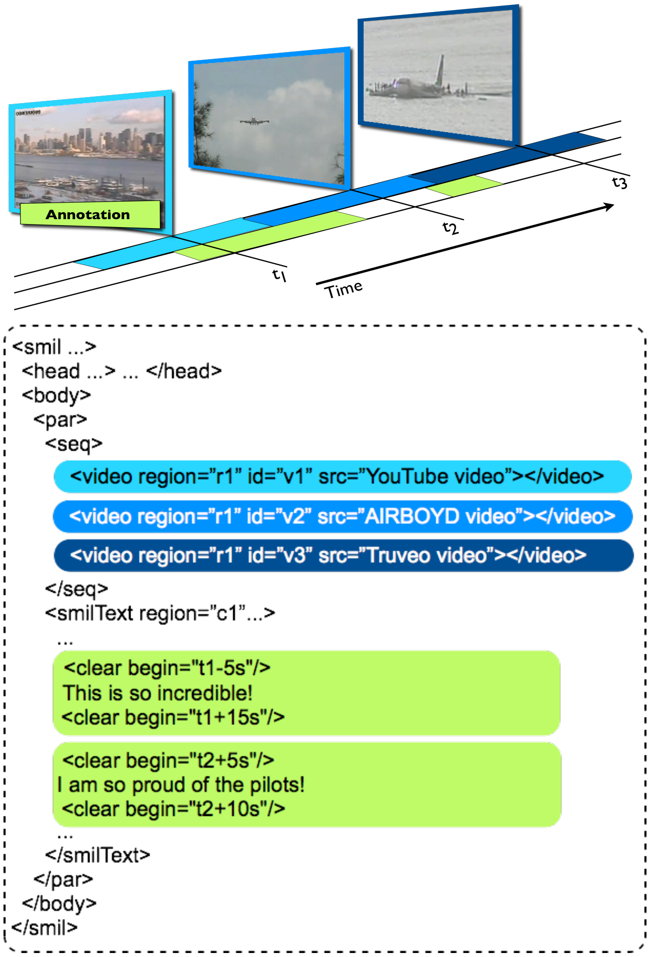
3. DOCUMENT TRANSFORMATIONS

The contribution of our work is the description of a set of document transformations that satisfy the requirements identified in Section 1. By document transformations we refer to the potential manipulations that can be applied to structured documents, in which one can add non-embedded, flexible temporal end-user comments. The transformations are possible because we create a structured multimedia document based on an input video. Our final objective is to provide enhanced video annotation tools that make use of the document transformations and thus leverage the authoring capabilities of end-users. Annotation of video is a topic that has been dealt with in many aspects, ranging from the usage of models that are not timed (e.g. HTML) or are unstructured (e.g. Flash) to standards such as MPEG-7 [8] and NCL. Based on our analysis of related work (Section 2), we rely on SMIL 3.0 as the basic framework that meets the requirements. The document model of SMIL 3.0 retains the base video integrity, and it allows multiple-video aggregation and multiple-provider integration. Timed-Text content and temporal hyperlinks allows end-users to add comments and to include timed end-user navigation, respectively. Contextual information of the annotations allows different levels of micro-personalized time-based annotations. Finally, the structured underlying model assures selective viewing.

3.1 Document Model

SMIL can integrate and compose a collection of audio, graphics, image, text, and video media items into a single presentation. Because Web media resources are by nature distributed – and might be very large in size - the SMIL language includes them by reference. SMIL defines a single generic media object (`<ref>`) element that allows the integration of external media objects into a SMIL presentation. However, it is also possible to use meaningful synonyms, when referencing external media objects (e.g. the `<video>` element is an alias for the generic SMIL media reference element). Note that as an implication of the usage of references, the integrity of the base media is preserved, meeting the requirement (i).

In addition, SMIL provides a powerful hierarchical composition model from which individual presentation timelines can be generated. The main temporal structuring elements are the parallel `<par>` and sequential `<seq>` containers, each of which provides a local time base for scheduling media objects (e.g. external videos) or child time containers. By using such time containers, it is possible to combine videos and comments in different temporal ways, as illustrated in Figure 3. In this example, three videos stored in different video servers are rendered as a continuous video, while the captions span across the videos. The structured time container behavior satisfies the requirements (ii) and (iii).



```
<smil ...>
 <head ...> ... </head>
 <body>
  <par>
   <seq>
     <video region="r1" id="v1" src="YouTube video"></video>
     <video region="r1" id="v2" src="AIRBOYD video"></video>
     <video region="r1" id="v3" src="Truveo video"></video>
   </seq>
   <smilText region="c1"...>
    ...
     <clear begin="t1-5s"/>
     This is so incredible!
     <clear begin="t1+15s"/>

     <clear begin="t2+5s"/>
    I am so proud of the pilots!
     <clear begin="t2+10s"/>
    ...
   </smilText>
  </par>
 </body>
</smil>
```

Figure 3. SMIL Document Model and Timed-Text Container.

[9] http://www.w3.org/AudioVideo

3.2 Timed-Text Content

Unlike most text formats [5], text content in SMIL is not only constrained by its style and layout capabilities, but also by the temporal context of the presentation. For instance, text must be rendered simultaneously with related objects, and it must be hidden when these are finished. Moreover, text content need to be synchronized with specific segments of the accompanying media object, such as when the text is used for captions.

The text content functionality in SMIL 3.0 allows authors to define small amounts of lightly formatted text containing embedded temporal markup within the context of a SMIL presentation. Such text may be used for labels within a presentation or for incidental captions or foreign-language subtitles. It is also possible to use large amounts of structured text (with or without temporal markup), but in this case it is recommended the use of SmilText as a text media object, or the use of objects encoded in formats such as XHTML or DFXP (*Distribution Format eXchange Profile*) [1].

The SmilText modules also define a set of additional elements and attributes to control timed text rendering. All SmilText content is processed in a manner consistent with other SMIL media. This means, among other aspects, that SmilText respects SMIL timing and layout behavior, including the semantics of the fit and fill attributes of SMIL Layout.

The SmilText profile also allows SmilText to be used as an external format. Moreover, since the smilText elements and attributes are defined in a series of modules, designers of other markup languages may reuse these modules when they wish to include a simple form of timed text functionality into their language.

SmilText as a text container with an explicit content model for defining timed text makes SMIL satisfy the requirement (iv).

3.3 Temporal Hyperlinks

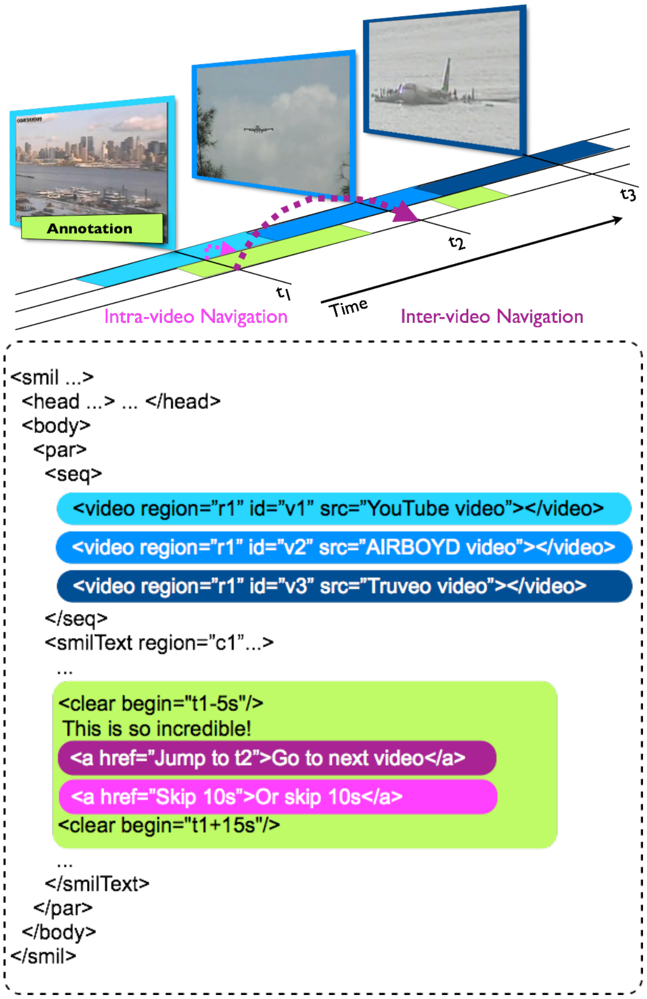
The SMIL 3.0 Linking Modules define the SMIL 3.0 document attributes and elements for navigational hyperlinking. These are navigations through the SMIL presentation that may be triggered by user interaction or other triggering events, such as temporal events. SMIL 3.0 provides only for in-line link elements. Links are limited to unidirectional single-headed links (i.e. all links have exactly one source and one destination resource).

As with styled time-based text annotations, adding temporal hyperlinks via text content can extend the ability of end-users to enrich the content viewing experience for them and for their social circle. This association makes SMIL meet the requirement (vii).

It is important to highlight that unlike the overlay navigation buttons in Figure 1, our document model allows links to be added to content without violating the legal rights on any party. This is possible because navigation points within the video are encoded as a series of content events in the SMIL document as shown in Figure 4 for the previous example.

Two classes of links can be provided (as illustrated in Figure 4):

- *Intra-video Navigation Link:* a text link that takes the viewer to another location within the same video content; and
- *Inter-video Navigation Link:* a text link that takes the viewer to another piece of content, outside of the active video.



```
<smil ...>
 <head ...> ... </head>
 <body>
  <par>
   <seq>
     <video region="r1" id="v1" src="YouTube video"></video>
     <video region="r1" id="v2" src="AIRBOYD video"></video>
     <video region="r1" id="v3" src="Truveo video"></video>
   </seq>
   <smilText region="c1"...>
   ...
     <clear begin="t1-5s"/>
     This is so incredible!
     <a href="Jump to t2">Go to next video</a>
     <a href="Skip 10s">Or skip 10s</a>
     <clear begin="t1+15s"/>
   ...
   </smilText>
  </par>
 </body>
</smil>
```

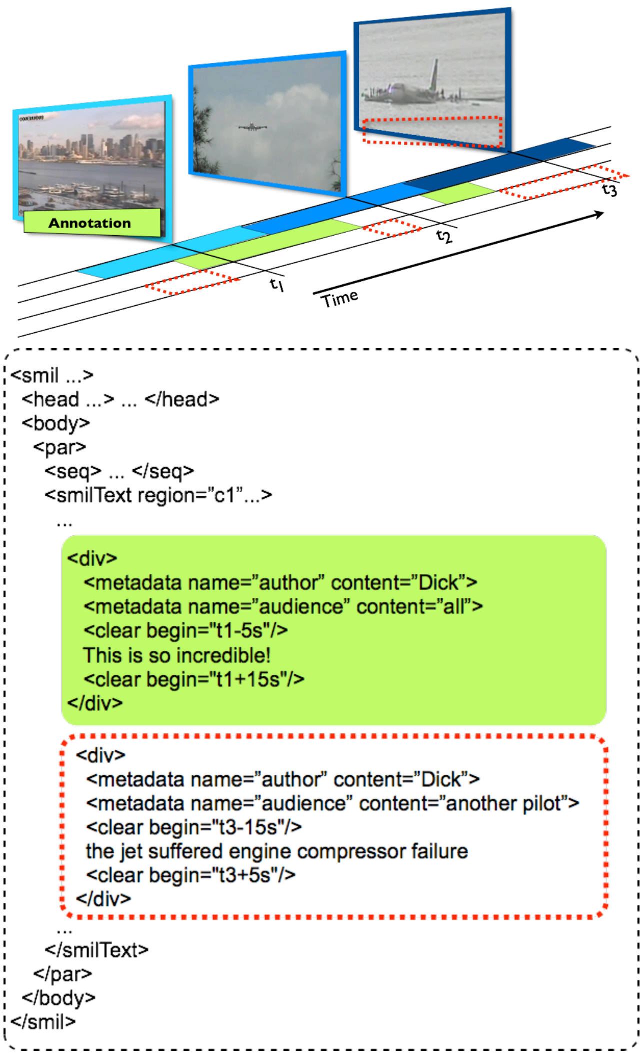
Figure 4. **Timed-Text Content and Temporal Hyperlinks.**

3.4 Contextual Information of the Annotations

Current video Web-based solutions provide limited support for including metadata related to the comments. For example, they do not allow end-users at authoring time to create different views on the annotations, depending on the target audience. One will not create the same annotations for his family and for his colleagues.

SMIL 3.0 allows associating meta-information to any element within the document body, including individual comments. This makes it possible to provide information on semantic intent within the presentation information, by binding relevant nodes with meta-information.

As mentioned before, SmilText allows text annotations to be described as single structured units that can be targeted to different audiences. Therefore, we can consider each comment entry as the smallest unit of annotation that can be tagged. In order to share a video with comments, essential metadata, such as who has created the annotations, when, why, how, and to whom, can be taken into account [9]. Support for targeted comments might increase the authoring overhead, but it provides a level of personalization that is lacking in common Web environments.



```
<smil ...>
 <head ...> ... </head>
 <body>
  <par>
   <seq> ... </seq>
   <smilText region="c1"...>
    ...
    <div>
      <metadata name="author" content="Dick">
      <metadata name="audience" content="all">
      <clear begin="t1-5s"/>
      This is so incredible!
      <clear begin="t1+15s"/>
    </div>
     <div>
      <metadata name="author" content="Dick">
      <metadata name="audience" content="another pilot">
      <clear begin="t3-15s"/>
      the jet suffered engine compressor failure
      <clear begin="t3+5s"/>
     </div>
    ...
   </smilText>
  </par>
 </body>
</smil>
```

Figure 5. Contextual Information and Selective Viewing.

SMIL can tackle the contextual problem (requirement v) by allowing text annotations to be tagged. Figure 5 illustrates this process. Here we see a master captions stream that has been composed by Dick specifically targeted for all viewers within his social circle.

3.5 Selective Viewing

One of the shortcomings of current captioning/annotation systems — whether closed captions or stream of comments on a Web page — is that every user is shown the same collection of annotation content during the video presentation. On the viewer point of view, public annotation posts can contain a lot of information not of all relevant to every end-user. For instance, it is doubtful that even the most interested reader will go through the set of nearly 10000 comments referenced in Figure 2 — but there is a much stronger incentive to view the 20 or so comments that are likely to be generated by family members or close personal friends.

In order to deal with such problem the structured nature of SMIL enables video annotation tools to apply different content selectivity alternatives (requirement vi). Video viewing tools can enable users to - besides the traditional turn on/off all annotations - select and watch the annotations created by a certain individual or community, the annotations about specific topics, or the annotations created on a certain day. Moreover, aggregated annotations and metadata can be used for generating diagrams of interest of videos. All of this is possible thanks to the document model - structured text annotations can be analyzed - and to the contextual information associated with the annotations. Figure 5 illustrates a scenario in which a viewer is interested in a certain category of comments.

4. A WEB-BASED VIDEO ANNOTATION TOOL

Ambulant Captioner[10], a Web-based video annotation tool, has been developed as a flexible testbed for justifying the contributions presented in this paper. It allows end-users to add structured temporal comments associated to any video by interfacing the document transformations identified in the previous section. Our tool, the Ambulant Captioner, makes use of the document transformations in a seamless manner, hiding the underlying complexities. The ultimate intention of the tool is to enable fluid communication with relatively dynamic social groups. Unlike many collaborative editing systems [16], however, the primary goal of content sharing is not the publishing of completed assets, nor the joint development of a collective common work, but instead to serve as a communication vehicle used by members of an extended community. Previous work includes the provision of integrated solutions for the full creation process [2], for remixing [15] and for repurposing tools [12].

The basic interface presented to an Ambulant Captioner user is illustrated in Figure 6. There is a primary video rendering space, a captions/comments rendering space and several sidebar controls. In most cases, relative passive end-users simply will want to watch a piece of content that was forwarded to them. If the content itself has embedded captions, these can be selectively turned on or off via the sidebar controls interface. The same is true for the native sound track.

During viewing, an end-user may also choose to insert new captions, either for general (broadcast) use, or for a specialized party. Figure 7 shows the extended input (relative to Figure 6) that is available when the 'add captions' button is switched on. The end-user may add new text that replaces or augments existing captions/comments. The timing can be (semi-) automatically determined, or directed in/out times can be added for each comment. In the future, we intend to provide styling support, in terms of letter type, size and color. Note that this is a major extension over the facilities provided by closed captions systems, where only minimal styling is typically supported [5].

4.1 Predictive Timing

As mentioned before, one key feature of this annotation tool is its ability to predict the timing and the temporal alignment of text annotations. The prediction can happen in two distinct ways as illustrated in Figure 8: viewing mode and direct access mode. The mode is automatically determined based on the state of the video player. For example, if the player is in playing state — the end-user is watching a video — the selected mode will be viewing mode. Otherwise, — the player is paused — the system will be in direct access mode.

[10] http://www.ambulantplayer.org/smilTextWebApp

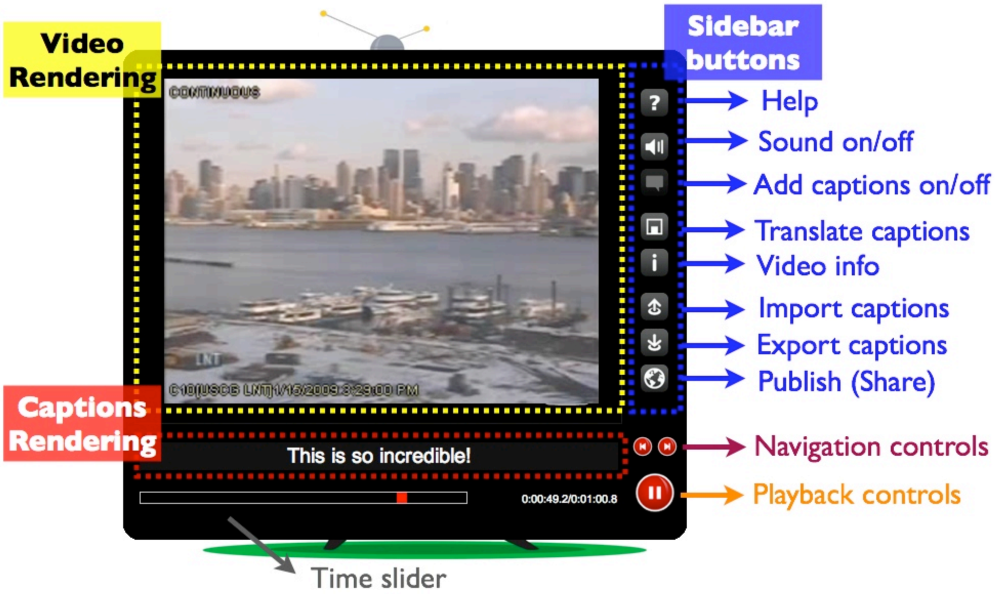


Figure 6. **Ambulant Captioner primary user interface.**

In viewing mode, we assume that comments might occur after the occurrence of an event in the video. Therefore, the end-user will react after the actual interesting moment has passed. Based on this premise, the end-user can use a keyboard shortcut to indicate that she wants to add a comment on the video. This action pauses the playback engine and focuses on the captions input area. Given this action was performed right after listening to or watching the event of interest, the current time moment (t_{now}) describes the end (t_{end}) of the comment entry (on the left side of Figure 8 $t_{end} = t_{now}$). As a preliminary guess, we consider the start time (t_{start}) equals the current time (t_{now}) minus a minimal duration (*MinDur*) that a short comment should stay in the screen for being effectively read ($t_{start} = t_{guess} = t_{now} - MinDur$). However, based on a duration model and parameters — the number of words in a comment entry (*N*), the average duration of a character/phoneme in a word depending on a specific language (α), and the average duration of pauses (β) — the value of t_{guess} can change, and t_{start} is now determined by the maximum value among t_{guess}, the end of the previous existing entry (t_{end}') and zero. Figure 8 shows scenarios in which t_{start} assumes different values. The video playback is resumed and the new annotation entry is saved when the activation key (keyboard shortcut) is pressed again.

In the direct access mode (player paused), the assumption is that when the end-user set the time slider to a specific point in the timeline, this point should be considered the start time of the annotation ($t_{start} = t_{now}$ on the right side of Figure 8). In this case the user is provided with an algorithm that takes into account the same parameters of the viewing mode to calculate t_{guess}, with the exception that depending on the text typed t_{end} goes onwards until the beginning of the next caption entry (t_{start}') or the duration of the video content (dur).

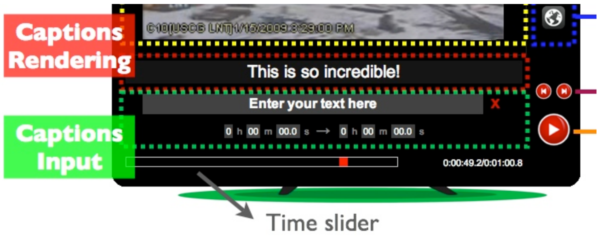


Figure 7. **Adding new captions.**

The use of such predictive timing will often provide only coarse temporal support; users may fine-tune the timing if desired. In our experience, such fine-tuning is not necessary unless tightly coupled subtitles are being created.

4.2 Implementation

In order to provide videos from different sources, Ambulant Captioner uses the Truveo Video Search[11] engine, which can be accessed through an AJAX (*Asynchronous JavaScript* + *XML*) API.

Since the search results do not provide the permanent URL (*Uniform Resource Locator*) of the videos files — but to a Web page that embeds a player for presenting the video —, a discovery module is necessary to solve the indirect URL provided by the search engine. In many cases, such indirection can only be solved at presentation time, when the video provider grants access to the video content for a certain period of time (e.g. YouTube videos). This module allows users to get videos directly from such sites and play them in the Ambulant Captioner.

For the video playback, we use the JW Player[12], which supports a vast range of media formats, and an embedded YouTube player, for YouTube videos. The YouTube and JW JavaScript API (*Application Programming Interface*) allow controlling an embedded video player via JavaScript. Calls can be made to play, pause, seek to a certain time in a video, set the volume, mute the player, and other useful functions. Most important, these APIs

[11] http://www.truveo.com

[12] http://www.longtailvideo.com

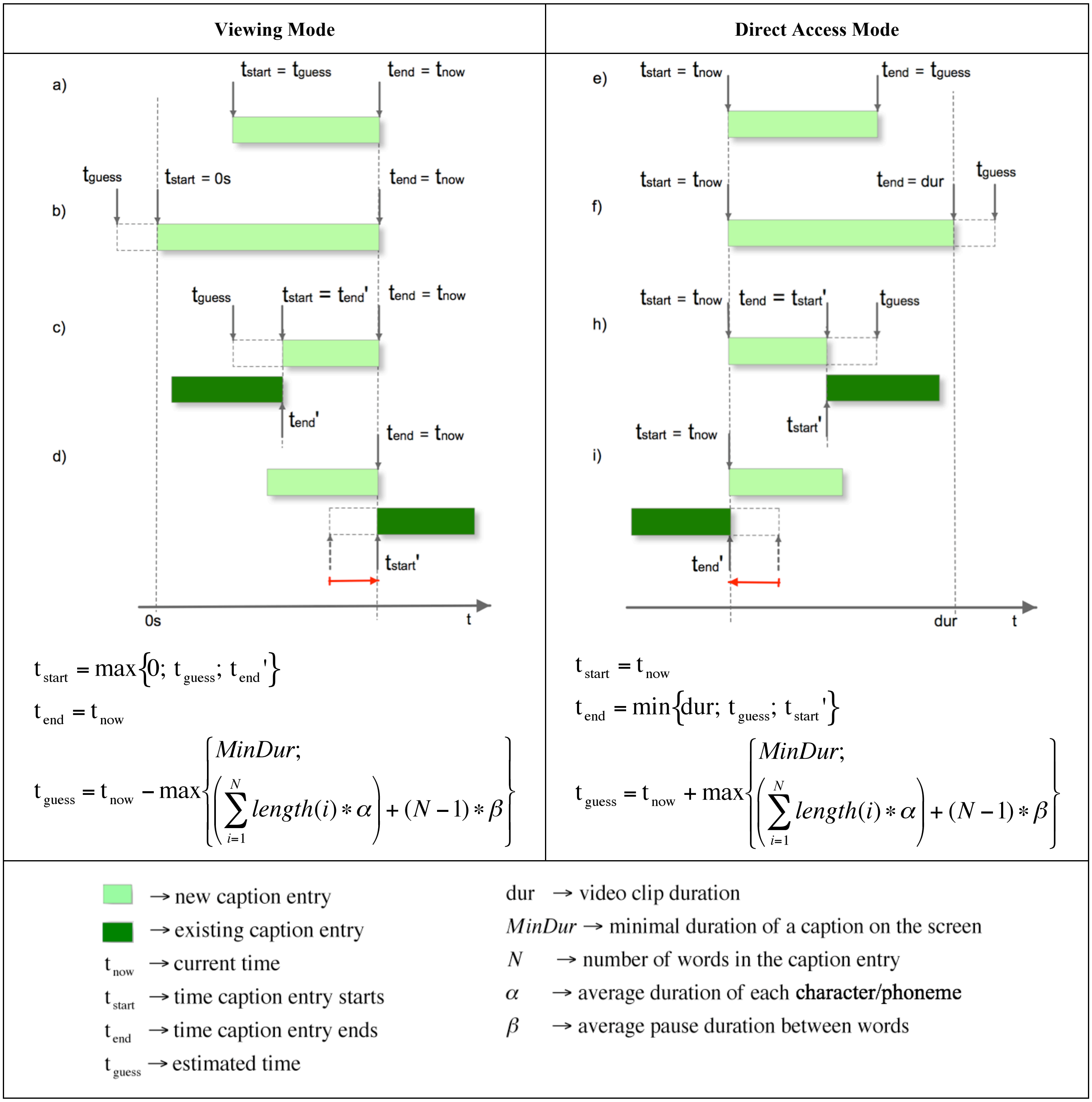
provide the video content temp
oral information necessary to synchronize the time-based text annotations.

As shown in Section 3, the requirement to store captions and comments separately from a base video implies the need for an encoding format for the captions themselves. The actual wrapper format used to encapsulate a video, plus a layered collection of captions/comments, is a JavaScript implementation of W3C's SMIL 3.0 that is tailored to our needs. Within this fragment, a video object is accompanied by appropriate metadata so that the player can negotiate with the relevant API to obtain the requested service.

Captions are defined using SmilText, the embedded text format for use within SMIL 3.0. The purpose of the SmilText JavaScript engine is to provide an implementation of SMIL 3.0 SmilText functionality within an HTML browser. The SmilText engine has reasonably complete coverage of the features defined in the SMIL 3.0 SmilText External Profile. The SmilText JavaScript engine allows source content to reside within the HTML markup (NB: not supported by all browsers), in a local file or on a server.

4.3 Usage of the Document Transformations

In order to use the document transformations identified before we



$$t_{start} = \max\{0;\ t_{guess};\ t_{end}'\}$$

$$t_{end} = t_{now}$$

$$t_{guess} = t_{now} - \max\left\{ \begin{array}{l} MinDur; \\ \left(\sum_{i=1}^{N} length(i) * \alpha\right) + (N-1) * \beta \end{array} \right\}$$

$$t_{start} = t_{now}$$

$$t_{end} = \min\{dur;\ t_{guess};\ t_{start}'\}$$

$$t_{guess} = t_{now} + \max\left\{ \begin{array}{l} MinDur; \\ \left(\sum_{i=1}^{N} length(i) * \alpha\right) + (N-1) * \beta \end{array} \right\}$$

🟩 → new caption entry

■ → existing caption entry

t_{now} → current time

t_{start} → time caption entry starts

t_{end} → time caption entry ends

t_{guess} → estimated time

dur → video clip duration

$MinDur$ → minimal duration of a caption on the screen

N → number of words in the caption entry

α → average duration of each character/phoneme

β → average pause duration between words

Figure 8. Automated temporal alignment of text annotation using predictive timing.

need a User Interface that hides all the complexity from the end-users, and just make use of the necessary functionalities. This is achieved with the Ambulant Captioner, which wraps the video content and all the styled timed-text annotations in a multimedia presentation.

The transformation process starts given one or more input URLs. Next, the Ambulant Captioner applies a document model transformation and a simple document presentation is generated as the output.

Timed-text content is applied as soon a user clicks on the 'add caption' button. This means that given a multimedia document, the Ambulant Captioner adds a parallel container that synchronizes the caption with a particular video or set of videos. Optionally, the Ambulant Captioner allows user to add navigation paths through temporal links.

Whenever a new comment entry is inserted implicit metadata is automatically added (e.g. who created the caption and when). Besides that, through the user interface, it is also possible to add additional contextual information, such as to whom the comments is targeted. Note that on the viewer side, depending on how the captions were specified the presentation engine might use SMIL state [10] for dynamically reason about exhibition of the captions.

5. CONCLUSIONS

Our work started with the goal to add non-invasive captions to YouTube content, but we rapidly discovered that the use of a rich document model to structure group-based comments had significant potential to be considered in the analysis of rich media social networking sites and the design of next generation video annotation tools.

From a document model perspective, all the requirements presented in this paper are feasible by using SMIL. The authoring system reported in this paper fulfills all the authoring requirements identified in Section 1. In the near future, we intend to implement a viewer that takes into consideration requirements (vi) and (vii).

The most relevant results of this work can be summarized as follows:

- *On protecting the integrity of video content:* One of the initial comments that we typically receive is: why go to enormous lengths to not place comments and links inside videos — everyone does it, so can you! We fundamentally disagree. Our interest is not so much in protecting Disney's copyrights (they will do this better themselves...), but to foster a sharing environment where 'clean' assets can be easily shared. While we sympathize with the technical expediency taken by YouTube (and others), as illustrated in Figure 1, we view this form of content overlay as being non-sustainable once real sharing starts to take place;
- *On providing timed comments/captions:* After many years of experience working on captioning systems and formats, we realized that for most people, adding detailed subtitles to content is simply too much effort. The payback is limited, unless the work is being done for a targeted community. What we have found in early trials with the Ambulant Captioner is that the system is used more for incidental labeling content and for inserting personal comments for directed friends of family members. This does seem to be worth the efforts of the user community. Moreover, the caption/label creation burden can be minimized by the use of predictive timing; and
- *On providing temporal linking within text captions*: Several technologies (principally SMIL), have allowed the non-embedded insertion of hyperlinks over video content for many years, but the technology has never gained widespread use. We suspect that one reason is not the authoring effort in creating links, but the visual effort in triggering them. By inserting custom navigation information in external text captions, a number of both technical and practical questions appear to be solved. Users are familiar with the link model, the timing associated with the link is natural (because it is associated with a caption) and the semantic labeling of the link is available 'for free' with the embedded text.

The contribution of this paper is not restricted to SMIL. In the last months there has been several initiatives for providing an adequate captions support for HTML5. In our opinion the current proposals – support of SRT – is far too restrictive and does not meet the requirements as outlined in Section 1. Therefore, we have been actively participating in the W3C's HTML5 Accessible Media sub-group. We expect that many of the contributions included in this article will be incorporated in the final HTML5 standard.

As future work we intend to explore ownership, version control and multiple-provider integration issues of the various versions of a document as it is annotated.

6. ACKNOWLEDGMENTS

This work was supported by EU FP7-ICT project TA2.

7. REFERENCES

[1] Adams, G. 2006. Timed Text (TT) Authoring Format 1.0 – Distribution Format Exchange Profile (DFXP), W3C.

[2] Adams, B., Venkatesh, S. and Jain, R. 2005. IMCE: Integrated media creation environment. In ACM TOMCCAP, 1(3), pp. 211-247. DOI= http://doi.acm.org/10.1145/1083314.1083315

[3] Adobe Systems Incorporated. 2008. Video File Format Specification, Version 10. http://www.adobe.com/devnet/flv/pdf/video_file_format_spec_v10.pdf

[4] Benevenuto, F., Rodrigues, T., Almeida, V., Almeida, J. and Ross, K. 2009. Video interactions in online video social networks. In ACM TOMCCAP, 5(4): n. 30, 2009. DOI= http://doi.acm.org/10.1145/1596990.1596994

[5] Bulterman, D. C. A., Jansen, A. J., Cesar, P. and Cruz-Lara, S. 2007. An Efficient, Streamable Text Format for Multimedia Captions and Subtitles. In Proceedings of the ACM symposium on Document Engineering, pp. 101-110, 2007. DOI= http://doi.acm.org/10.1145/1284420.1284451

[6] Cesar, P., Bulterman, D. C. A., Geerts, D., Jansen, J., Knoche, H. and Seager, W. 2008. Enhancing social sharing of videos: fragment, annotate, enrich, and share. In Proceeding of the 16th ACM International Conference on Multimedia, pp. 11–20, 2008. DOI= http://doi.acm.org/10.1145/1459359.1459362

[7] Choudhury, M. D., Sundaram, H., John, A. and Seligmann, D. D. 2009. What makes conversations interesting? Themes,

Participants and Consequences of Conversations in Online Social Media. In Proceedings of the International WWW Conference, pp. 331-340. DOI= http://doi.acm.org/10.1145/1526709.1526754

[8] IEEE MultiMedia, MPEG-7: The Generic Multimedia Content Description Standard, Part 1, IEEE MultiMedia, v.9 n.2, pp. 78-87, April 2002. DOI Bookmark= 10.1109/93.998074

[9] Fagá Jr, R., Furtado, B. C., Maximino, F., Cattelan, R. G. and Pimentel, M. G. C. 2009. Context information exchange and sharing in a peer-to-peer community: a video annotation scenario. In ACM Special Interest Group for Design of Communication, pp. 265-272. DOI= http://doi.acm.org/10.1145/1621995.1622048

[10] Jansen, J. and Bulterman, D. C. A. 2008. Enabling adaptive time-based web applications with SMIL state. In Proceedings of the ACM symposium on Document Engineering, pp. 18-27. DOI= http://doi.acm.org/10.1145/1410140.1410146

[11] Nathan, M., Harrison, C., Yarosh, S., Terveen, L., Stead, L. and Amento, B. 2008. CollaboraTV: Making Television Viewing Social Again. In Proceedings of UXTV, pp. 85-94. DOI= http://doi.acm.org/10.1145/1453805.1453824

[12] Pea, R., Mills, M., Rosen, J., Dauber, K., Effelsberg, W. and Hoffert, E. 2004. The DIVER project: interactive digital video repurposing, In IEEE Multimedia, 11(1), pp. 54-61. DOI Bookmark= 10.1109/MMUL.2004.1261108

[13] Pimentel, M. G. C., Cattelan, R. G., Melo, E. L., Prado, A. F. and Teixeira, C. A. C. 2010. End-user live editing of iTV programmes. In International Journal of Advanced Media and Communication, 4(1), pp.78-103. DOI Bookmark= 10.1504/IJAMC.2010.030007

[14] Shamma, D. A., Kennedy, L. and Churchill, E. F. 2009. Tweet the Debates: Understanding Community Annotation of Uncollected Sources. In Proceedings of the first SIGMM Workshop on Social Media, pp. 3-10. DOI= http://doi.acm.org/10.1145/1631144.1631148

[15] Shaw, R. and Schmitz, P. 2006. Community annotation and remix: a research platform and pilot deployment. In Proceedings of 1st ACM International Workshop on Human-Centered MM 2006 (HCM '06), pp. 89-98. DOI= http://doi.acm.org/10.1145/1178745.1178761

[16] Sgouros, N. M. and Margaritis, A. 2007. Towards open source authoring and presentation of multimedia content. In Proceedings of the ACM Workshop on Human-Centered Multimedia, pp. 41-46. DOI= http://doi.acm.org/10.1145/1290128.1290136

[17] Soares, L. F. G., Moreno, M. F. and Sant'Anna, F. 2009. Relating Declarative Hypermedia Objects and Imperative Objects through the NCL Glue Language. In Proceedings of the ACM symposium on Document Engineering, pp. 222-230. DOI= http://doi.acm.org/10.1145/1600193.1600243

[18] Soares, L. F. G., Rodrigues, R. F. 2006. Nested Context Language 3.0 Part 8 – NCL Digital TV Profiles. Technical Report. Departamento de Informática da PUC-Rio, MCC 35/06. http://www.ncl.org.br/documentos/NCL3.0-DTV.pdf