Video on the Web: Experiences from SMIL and from the Ambulant Annotator

Dick C.A. Bulterman, A.J. Jansen, and Pablo Cesar
CWI: Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam
The Netherlands
{Dick.Bulterman, Jack.Jansen, P.S.Cesar}@cwi.nl

ABSTRACT

Since the arrival of YouTube on the desktop, video has entered its second lifetime on the Web. The main difference between this incarnation of video and its predecessors is at the source: where first generation video was about repurposing content, the YouTube generation is all about user-generated content and few-to-few (rather than one-to-many) sharing. The fact that video is not new to the Web is a great advantage. It means that much of the work from the past can be reused and updated to meet current needs. This paper provides an overview of how video (and audio) have been processed on the Web using SMIL. It also provides a discussion of some extensions to SMIL functionality that show how video is processed as a first-class object in a video interaction framework within the Ambulant Annotator.

1. INTRODUCTION

Video content presents an opportunity and challenge to the Web community: massive objects of time-constrained opaque content need to be defined, found, transferred and rendered across an infrastructure that is inherently time-insensitive, using browsers, interfaces and tools that have been optimized for non-temporal character-based data. Many parties have a lot to gain from the current video-on-the-Web focus, from hardware and networking vendors, to language and interface designers, to the millions of end-users who carry video devices with them every minute of every day. The challenge for the Web community is to find a useful niche in which new tools and technologies can be developed that make video more useful in an open framework.

The Distributed Multimedia Languages and Infrastructures group at CWI in Amsterdam has been studying various tools and technologies for higher-order (post-codec) video sharing. Our central focus has been on transforming a video object from a closed container to a more open, layered container in which generations of viewers can customize access to, and share annotations across, bounded fragments of video — but in a manner that still respects and protects the integrity of the underlying video object [3],[4]. One aspect of this work has been the development of a *micro-recommendations* architecture, in which users can interactively build personal recommendation messages *while watching* the video. At the same time, we have worked within the W3C community to make all forms of timed-media (video, audio, animated graphics, hyperlinks) first class web citizens.

This paper provides a summary of existing Web support for video as provided by Synchronized Multimedia Integration Language (SMIL). The way that SMIL manipulates video content — and, more importantly, the way that SMIL integrates video in the larger context of a coherent media-centered presentation — can provide useful insights to designers of a new generation of Web video tools. This paper also provides a snapshot overview of one of our approaches to enriching video without altering the component's encoding: a framework for data-model-based sharing of video context in a peer-user social network, in which context information outside the codec is used to serve related information as an adjunct to the main video data.

2. PROCESSING VIDEO WITH SMIL

This section provides an overview of the basic timing concepts that are used within a SMIL presentation.

At first glance, SMIL provides a feature-rich and somewhat overwhelming collection of control parameters. An urge may exist to choose a simpler path: just define a top-level video tag that activates a video object. While this approach is appealing in its directness, we are convinced that it presents a dead-end path. The most important part of video integration in future Web contexts is that video is a peer content element: it needs to exist and inter-operate with other elements in a presentation. Unfortunately, this brings with it the need to provide inter-object control. In the temporal world of video, this control is often synchronous in nature.

The complexity of SMIL results in large measure from the need to control dozens of types of synchronization relationships between a video object and its environment. Of course, given unlimited amounts of special-purpose Javascript code, a declarative synchronization framework may not be necessary — but this approach does not provide the transformable, reusable framework that made the Web famous. Although it may be possible to choose a simpler subset than that supported in a full SMIL dialect, the fact remains that the host level environment cannot escape synchronization issues if rich video manipulation is desired.

Note that the use/substitution of SMIL concepts is not a video codec issue: the more control that is migrated to the codec, the less the flexibility to adapt a single video content container for different environments and purposes. Making such manipulation possible is a key part of SMIL.

2.1 SMIL Timing Basics

A SMIL presentation distinguishes two types of timing control: control that determines how much of a particular media object is rendered (and looped) during a single instance of its presentation, and control that determines when a particular media object (or sub-structure of objects) gets activated and terminated relative to other objects in the presentation.

2.1.1 SMIL Time Containers

A SMIL presentation does not consist of one fixed timeline, but a nested collection of timelines -- some with pre-defined scheduled durations, some within interactive durations. This allows a single video to be accompanied by multiple set of subtitles, or to be able to have differentiated background music. The hierarchy of timelines is specified using three time containers: the parallel container (<par>), the sequential container (<seq>) and the exclusive container (<excl>). Of these, the <par> container is the most general: it defines a generic timeline on which its children can be scheduled. The <seq> provides a convenience container in which each of the children are scheduled by default to start at the conclusion of their lexical predecessor. The <excl> container allows a number of peer-level candidates to be specified, of which only one will be active at any given point -- starting one of the other peers typically replaces the currently-active peer.

Each continuous media object also defines a *pseudo* time container: it defines a time base that can be used to bring various pieces of supplemental information (such as link anchors) into scope.

The set of SMIL time containers provides a basis for inter-media synchronization. The following fragment illustrates this:

```
...
<par>
  <video  ... />
  <audio ... />
</par>
...
```

SMIL can also be used to structure the logical parts of a video:

```
...
  <seq>
    <video  id="scene1" clipBegin="0s" clipEnd="12s" src="..."/ >
    <video  id="scene2" clipBegin="12s" clipEnd="22s" src="..."/ >
    <video  id="scene3" clipBegin="22s" clipEnd="34s" src="..."/ >
    <video  id="scene4" clipBegin="34s" clipEnd="46s" src="..."/ >
  </seq>
...
```

In this example, the whole video is shown as a continuous object, but the separate segments allows content-based (rather than only timeline-based) navigation: a hyperlink can take a viewer directly to *scene3*.

2.1.2 Relevant SMIL Duration Concepts

There a several duration-related concepts in SMIL that are important for understanding how a video is used. These are:

- *inherent duration*:The inherent duration is the 'natural' duration of a media object.
- *simple duration*: The simple duration is the inherent duration of a media object, possibly modified by *clipBegin* and *clipEnd* attributes. The simple duration can be overridden using the *dur* attribute.
- *active duration*: The active duration is the simple duration of a media object, possibly extended by specifying a loop count or a loop duration.

These time definitions are important because they represent a hierarchy of temporal contexts in which a video can be processed: the context of the media object, the context of one instance of the media object and the context of the instance within the greater whole of the enclosing presentation.

Note that a video's inherent duration is not always easy to determine. Some formats include duration information as part of the video header, but this is not always the case. Live video feeds (which are globally continuous and have no set beginning or end) have no defined inherent duration. Also, many video encodings do not define the inherent duration of their object; in these cases, the only way to determine the inherent duration is to scan the entire video file.

2.2 Controlling a Video Instance

This section reviews the attributes used to define a single instance of a video object's activation. The attributes are discussed in Table 1.

SMIL provides a rich time value syntax, ranging from simple time in seconds to full SMPTE support. Interested readers should consult the SMIL *Timing and Synchronization module* for details.

Table 1: Video Instance Control Attributes

Attribute	Arg Type	Definition	Example
clipBegin	time value	Defines the temporal offset that serves as the start of a clip. Defaults to '0s'.	<video src="... " clipBegin="3s" ... />
clipEnd	time value	Defines the temporal offset that serves as the end of a clip. Defaults to the end.	<video src="... " clipEnd="12s" ... />
begin	time value or event definition	Defines the begin time of the media object either as a time value or an event name.	<video src="..." begin="3s" ... />
end	time value or event definition	Defines the explicit end of the active duration, either as a time value or an event name.	<video src="..." end="12s" ... />
dur	time value	Defines the simple duration of a media object or time container.	<video src="..." dur="10s" ... />

In a SMIL presentation, the start and end of a particular video object does not occur in a vacuum. It may be synchronized with other objects that are also defined in the presentation (or, in HTML terms) on the page. SMIL distinguishes two types of activation behavior: scheduled activation/termination and event-based activation/termination. It is possible to mix both scheduled and event-based activation/termination: these forms may be mixed (the first timing control value that gets resolved will control the object).

2.3 SMIL Temporal Linking Concepts, Elements and Attributes

One of the most powerful interaction features of SMIL is the ability to specify time-variant anchors that allow temporal navigation across a presentation. The key to this feature is that SMIL does *not* place anchors in the video content, but defines anchors as peer-level content that is activated along with the video object. Each of the anchors has a visual component (defined by an area attribute), a scheduled component (defined by *begin/end/dur* attributes) and a link target component (defined by an *href* attribute):

```
...
<video src="video" title="Interview" >
   <area begin="3s" dur="10s" title="first question"
        href="#question"/>
   <area begin="20s" dur="20s" title="first answer"
        href="http://www.example.org/answer"
       shape="rect" coords="10.2, 14.5, 48.3, 62.7"
        sourcePlaystate="pause"/>
</video>
...
```

In this example, the video object has two anchors defined: one begins 3 seconds into the video and is active for 10s, while the second begins 20 seconds into the video and remains active for 20s. In this example, the first anchor covers the entire visual area of the video object, while a specific shape and placement relative to the object (using the *shape* and *coords* attributes) is defined for the second anchor.

Anchors can also be used to segment video content, or to attach temporal metadata to individual objects.

2.4 Integrating SMIL Timing and Synchronization

The SMIL language is highly modularized, which allows language designers to pick-and-choose the parts that they need to support video timing and synchronization. There are also existing technologies for integrating all (or part) of SMIL into an HTML framework: *XHTML+SMIL* and the new *SMIL 1.0 Timesheets* proposal.[1]

3. A FRAMEWORK FOR VIDEO INTERACTION VIA THE DATA MODEL

SMIL provides a host of packing structures to allow a video to be a central part of an embedded media presentation. With the first generation of video on the Web, the video content either was presented without a contextual wrapper, or it was locked inside a wrapper presentation.

One of the opportunities with second-generation video is the ability to separate core video data from one or more sets of secondary data streams. These stream can then be selectively included in some higher-order presentation.

Consider the application shown in Figure 1. Here, a SMIL object is shown that contains fairly conventional video+text data. The language used for subtitles is Dutch: this setting could have been inherited from a parent non-SMIL presentation, such as (X)HTML. The HTML page also presents a host

1. All of these specifications are available from the SMIL website: http://w3.org/AudioVideo/ .

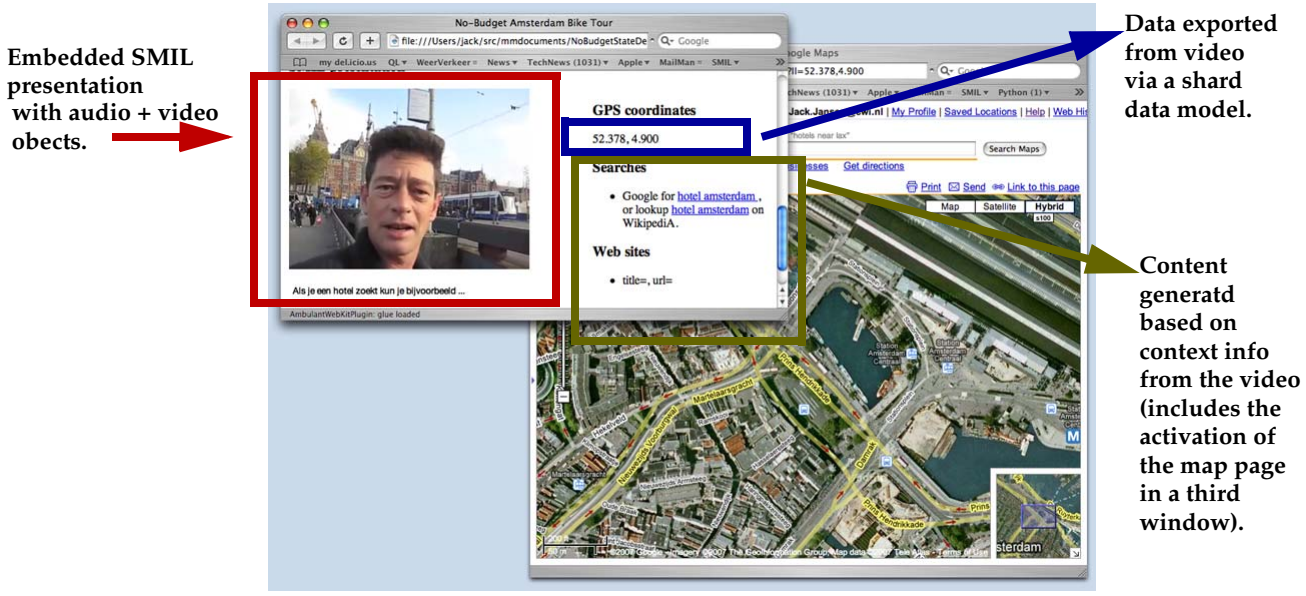

Figure 1. An example of an embedded multimedia object (coded in SMIL) that steers contextual information in the embedding Web page.

of (optional) additional information, based on data exported to the HTML page from within the video. Note that the video designer need not know (or care) how this information is used -- this is the role of the integrating page.

Based on the experiences from European projects such as Passepartout (ITEA) and SPICE (IST IP), we can conclude that there is a need for a standardized mechanism to provide rich interaction for media content. This section surveys a framework that permits web developers to share the data-model between temporal-based documents and the web browser. The framework adds a controlled temporal dimension to existing a-temporal web browser.

3.1 Background

In order to inter-operate from a services-oriented perspective with video, an interaction model needs to be defined that transcends the traditional control set of *start* / *stop* / *pause*. The content within the video element will need to be triggered from external, peer-level content and the video, in turn, will need to trigger related content within the context of a higher-level embedding. We feel that the key to extended content-related (not content-based!) interaction lies in wrapping the video with an external data model. The data model – rather than the video encoding – should become the focal point for sharing, mashing and reusing individual objects.

3.2 Communication and interaction via the data model

There is a clear need for richer temporal semantics when integrating a conventional (X)HTML browser interface with multimedia documents described using SMIL, SVG, or HTML+Time. To this end, we have developed a framework that includes a language-independent data model, support for evaluating expressions and manipulating state variables, and mechanisms for state variables[2] storage/retrieval.

Firstly, the data model is defined as a small XML document that can be expressed and addressed using XPath. This data-model is language-independent and can be shared between different XML-based documents such as (X)HTML and SMIL. The data model is defined by a <*state*> container element. Figure 2 shows an example, in which the data model,

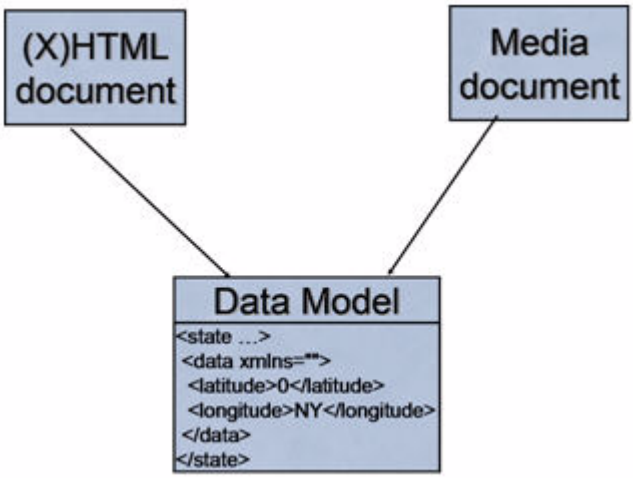

Figure 2. Shared Data Model.

in this case a geographical location, is shared between a media document and an (X)HTML document.

In addition, the framework provides support for defining and manipulating the value of associated variables. For example, we can modify the value of the *latitude* variable within the media document, at a given time of the presentation, by using the element <*setvalue*>. In our example, this would be achieved by the following line of code:

<*setvalue ref="latitude" value=" 52.429222">*.

(The actual value for the variable might be obtained at runtime from some other Web component.) Moreover, the framework provides the mechanism to evaluate variables and returns a Boolean value. Finally, the framework allows saving the state variables value for the next time the media document is played.

By exporting the data model to the outside world, it becomes possible for the media document to affect other contexts (e.g., the (X)HTML presentation); at the same time, external engines can affect the media presentation. Figure 3 shows two examples on how the state variables can be shared. On the left, the video exports location coordinates, depending on the current played scene. These coordinates can be used, for example, by an external utility to show the location on a map. On the right, user interaction affects and customizes the actual media presentation to be played at runtime.

2. The full specification can be found in the SmilState module on the SMIL 3.0 specification [http://www.w3.org/TR/SMIL3/smil-state.html]

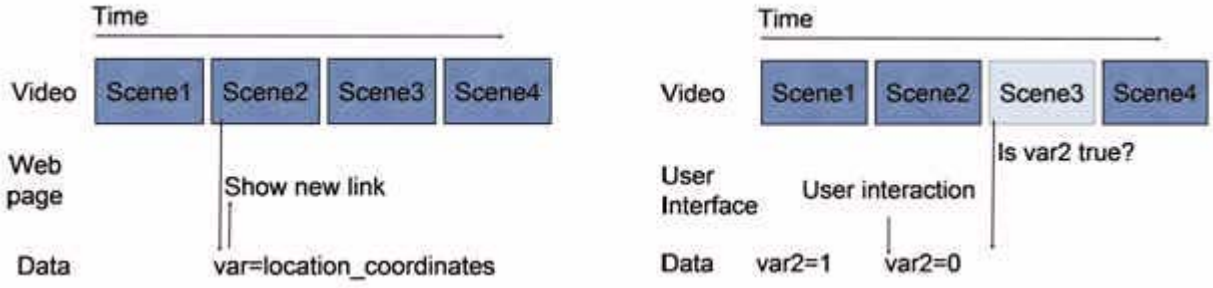

Figure 3. Example on the interaction between the media engine and external engines

Some motivating scenarios of our work include:

- *E-tourism*: an online guide of a city that includes videos of the place of interest. The presentation can dynamically export the coordinates of the locations presented in the videos, which can be used for representing the guide tour in an external engine in a mobile device, which could present on-demand information to a user depending on location.
- *E-learning*: an e-learning portal that includes synchronized videos and slides. An interactive test controlled by an external calculation engine can provide results to the media player, so the learning material can be adapted to the knowledge of the student.
- *E-commercials*: media-based commercial can be customized to a specific user by, for example, displaying the name of the user and adapting the media presentation based on user preferences.
- *Persistent segmentation*: for example, by allowing the user to explicitly pause a presentation and then restart it at some later point – possibly days or weeks later.

Note that unlike almost independent embedded video players, in our examples the video plays an active role in the webpage. And, thus, the data model of the multimedia presentations is shared with the browser.

3.3 Current Status

At the writing this article, the framework has been implemented in the Ambulant [1] open-source SMIL player and is available for use. The framework is completed for the MAC OSX and Win32/CE versions of the player, while the support for Linux-based platforms is under development. The work sketched in this article has been submitted to the W3C's SYMM working group under the name of *smilState*. This effort is based on our participation in W3C Backplane activity [5]. It is expected to be integrated in the SMIL 3.0 release in early 2008.

4. VIDEO ON THE WEB: DIRECTIONS

The W3C workshop on video has set out a broad agenda of topics. Based on our experiences with video and timed-media manipulation, here is a summary of our perspectives:

- *Strategic thinking*: making video a first-class citizen means making video a peer-level citizen with other first class objects. For this to happen, a common temporal basis needs to be defined at the highest levels.
- *User experience*: Many of the efforts on interactive television have failed because of the baseline assumption of centralized control over data distribution and viewing. While legitimate rights of the publisher need to be respected, video sharing is not simply a matter of selection but of *interacting* with video content.
- *Video production*: The main challenge for video on the web is probably not the development of a replacement for MPEG-7's metadata standard. Instead, the problem is more of packaging: how can end-users associate their own meaning and content to a shared piece of video content.
- *Web architecture*: There is a lot that can be learned from SMIL and SVG. The main contribution may be that it is 'time' that needs to be a first class citizen, not any particular media encoding.

Video is an important class of web asset, but its importance lies not only in what the video 'says', but how the video is 'used' in a broader context. Making that context available, sharable and customizable presents a wonderful opportunity for enriching Web content.

5. ACKNOWLEDGMENTS

This work has been funded by the NWO BRICKS PDC3 project, by the ITEA Project Passepartout and by the FP6 IST project SPICE. Development of the open source Ambulant Player and CWI's participation in the SMIL standardization effort have been funded by the NLnet foundation.

REFERENCES

[1] Bulterman, D. C. A., Jansen, J., Kleanthous, K., Blom, K., and Benden, D. 2004. Ambulant: a fast, multi-platform open source SMIL player. In Proceedings of the 12th Annual ACM international Conference on Multimedia MULTIMEDIA '04. ACM, New York, NY, 492-495.

[2] Bulterman, D.C.A., Is It Time for a Moratorium on Metadata?, IEEE Multimedia, 11(4) , Pp. 10-17, 2004.

[3] P. Cesar, D.C.A. Bulterman, Z. Obrenovic, J. Ducret, and S. Cruz-Lara, "An Architecture for Non-Intrusive User Interfaces for Interactive Digital Television," Proceedings of the 5th European Interactive TV Conference, Amsterdam, The Netherlands, May 24-25, 2007, pp. 11-20.

[4] P. Cesar, D.C.A. Bulterman, and A.J. Jansen, "Social Sharing of Television Content: An Architecture," Proceedings of the IEEE Symposium on Multimedia, TaiChung, Taiwan, December 10-12, 2007.

[5] W3C Backplane: http://www.w3.org/MarkUp/Forms/2006/backplane/