EUROGRAPHICS Workshop on Multimedia (2004)
N. Correia, J. Jorge, T. Chambel, Z. Pan (Editors)

Creating Peer-Level Video Annotations for Web-Based Multimedia

Dick C.A. Bulterman

CWI: Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands

Abstract

The TabletPC is an example of a new generation of user interface device where pen-based manipulation of information is integrated directly into a user's workflow. Using the TabletPC's existing pen and electronic ink systems, a wide range of static documents can be created or annotated. While the facilities of the TabletPC are useful for creating virtual images containing ink that can be overlaid on text or picture context, there is little support for creating annotations of time-based content such as video.

This article describes an annotation authoring model and interface for creating peer-level annotations to video media. Peer-level annotations allow existing content to be enriched with additional content annotations that can be co-presented with the original media. A system for creating a SMIL language document containing SVG-based annotations that exist along-side the visual content is described, along with a discussion of the needs and limitations of supporting video markup in a web context. An example using peer-level annotations in a medical context is provided.

Categories and Subject Descriptors (according to ACM CCS): 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.

1. Introduction

One of the basic properties of most multimedia presentations was a fundamental division between the activities of creating and consuming information. Unlike text, which could be easily edited and enriched, audio, video and image data have long been relatively 'closed' media types. Recent changes in user interface technology have begun to blur this distinction. One example of such a device is the TabletPC.

The TabletPC is a small-format laptop that provides an integrated drawing tablet and pen as its primary user interface device. The TabletPC can been seen as a step in interface evolution that tries to free the user-machine interface from a keyboard based model for standard tasks such as note-taking, data entry and information searching.

In order to better understand the facilities provided the Tablet PC for document mark-up, consider the sample document in Figure 1. Here we see a text document that marked-up using the Windows Journal [Mic02] and which contains the following types of annotations:

- highlighted text using semi-transparent virtual ink;
- text annotations based on handwriting recognition;
- free-hand (non-interpreted) text mark-up;
- free-hand (non-interpreted) vector drawings;
- audio commentary; and
- a flag, indicating something important.

The Windows Journal has other features, but these are not relevant to our discussion.

The workflow used by the Windows Journal consists of the following steps:

1. The document to be edited is converted into a internal format, in which it is treated as a virtual image.
2. The annotations are made on a layer above the document using the facilities of the Journal interface.

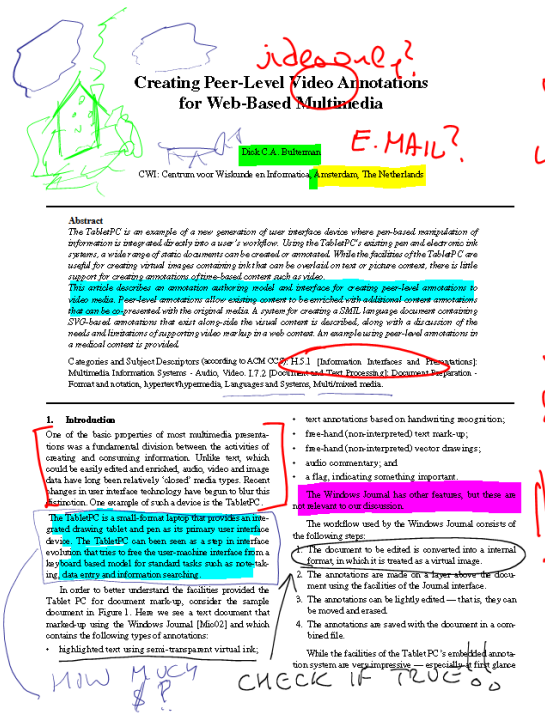

Figure 1. *Annotation using the Windows Journal [Mic02].*

3. The annotations can be lightly edited — that is, they can be moved and erased.
4. The annotations are saved with the document in a combined file.

While the facilities of the Tablet PC's embedded annotation system are very impressive — especially at first glance — the underlying annotation model has a number of restrictions. A principal problem is that annotations can only be attached to static documents, and then only to a single projection of that document as an un-interpreted image.

The TabletPC's pen-and-ink model is one of various approaches to annotating media objects. Understanding the differences in scope of these approaches is important in constructing an annotation system. A classical definition of annotation reads:

> *Meta-information associated with a document providing an enrichment of the document* (Rigamonti, 1998)

The unfortunate aspect of this definition is that is ties all annotation to metadata. While metadata-based annotation is useful for archival purposes, it is rarely suited as the basis for augmented presentations of the type described below in Section 2. In order to better differentiate the two major uses of document annotation, we prefer the following definitions:

- *Hierarchical annotations*: document markup (including metadata) that provides an abstract classification of media content for a given use and ontology; and
- *Peer-level annotations*: document markup that provides companion information and which results in augmented media content.

(We also can define other classes, such as syntax annotations for edit lists, but these are beyond the scope of this paper.)

The ultimate goal of our work is to be able to support 'open' peer-level annotations: hierarchical annotations, in which various generations of documents can be created and maintained. As a first step toward this goal, this paper looks at user-related requirements and implementation issues for supporting first-degree peer-level annotation of video media objects. We start with a brief example of the types of annotations we address with our work. We then describe the requirements for a video annotation system and compare these requirements to other examples of annotation systems. We next describe our interface for creating and maintaining peer-level annotations, including the real-time creation of a SMIL document framework with embedded SVG annotation paths. We close the paper with a description of the *Ambulant Annotator* [Amb04]: an implementation of the annotation environment for creating and viewing annotations based on the concepts presented in the paper. Note that while the TabletPC is used to motivate initial interest in video annotation, the same facilities can be supported on conventional PC's with a digitizing tablet or even a mouse-only interface.

2. Peer-level annotation of a video object

Figure 2 shows three frames of a video of a horse. There are many possible uses for this video: it could be used to showcase the jockey, to describe one of the potential uses of a dirt roadway or as part of a sales brochure for the horse and/or wagon. Describing the video for each of these uses would require the specification of multiple collections of classification keywords, each taken from specialized domain vocabularies.

Suppose that we wanted to use the video as a diagnostic aid in describing symptoms related to problems in the back right leg of the horse. These descriptions could consist of:

- a text document that describes background information on the history of the horse;
- a piece of audio commentary that explains a particular symptom; and
- a set of pen annotations that are placed "on top" of the video, highlighting particular problems at the moment in the video that these are most relevant.

Figure 3 illustrates the video object after such annotations are applied:

Figure 2. *Three fragments from a video object. The horse and rider are shown at 50, 60 and 75 seconds into the video image.*

- The frame labelled X_{50} contains a small text icon near the upper right corner. Selecting this icon would pause the video and would bring up a page of information in an associated browser.
- The frame labelled X_{60} contains an audio icon at top right. Selecting this icon would start an audio commentary that was synchronized with the video. The frame also contains an ink object over the horse's back leg. This ink could be used to highlight a particular problem.
- The frame labelled X_{75} shows the result of having the ink that appeared in X_{60} be animated and tracked across the image. The object has also changed size and shape during the animation.

A key concern in creating the annotations described above is that all of them must not corrupt the original video: any annotations must consist of separate objects that are presented in parallel with the video object. As will be shown, we use SMIL [BR04] as an encapsulating language to describe both the video and the annotations.

3. Annotation characteristics and user interface

Annotation of audio and video is a by-product of the digital era [DSP91],[MD89]. One of the earliest applications of media annotation was for defining edit lists: collections of media object excerpts that could be used to redefine the order in which media content was rendered [DWC01]. The facilities available for media editing led to research in the area of hierarchical media annotation with various classes of metainformation [FQA88],[Mar97]. The goal of hierarchical markup is either to assist in content classification (for use in indexing or retrieval applications) or in providing an abstract semantic model of the media object's content for (semi-)automatic processing on the media object [HvOR01]. Peer-level annotation techniques for continuous media have been less studies, chiefly because of the limitations of platform and interface technology in supporting this form.

3.1 Hierarchical annotation characteristics

Hierarchical annotations consist of metadata information that are based on an ontology about a particular subject domain. Figure 4 illustrates a video annotation system that can be used to produce hierarchical annotations on animals.

Hierarchical annotations are created as a post-production activity that provides a content layer that is at a higher level of abstraction than the base document. This can be a useful model when using annotation for searching or analysis, but it is limiting when using annotation for providing augmented content.

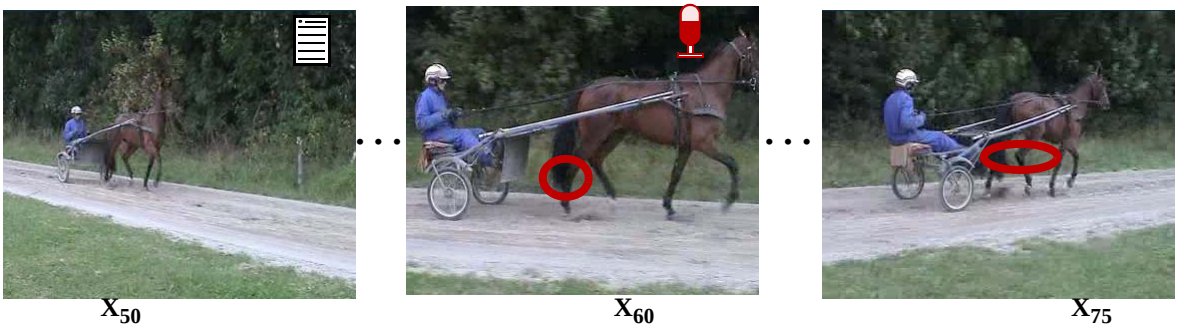

Figure 3. *Three annotated fragments from a video object. The icon at top right of X_{50} indicates the presences of a text annotation. The circle at the center of X_{60} is an ink markup of the video. This markup travels with the video as an animation through X_{75}.*

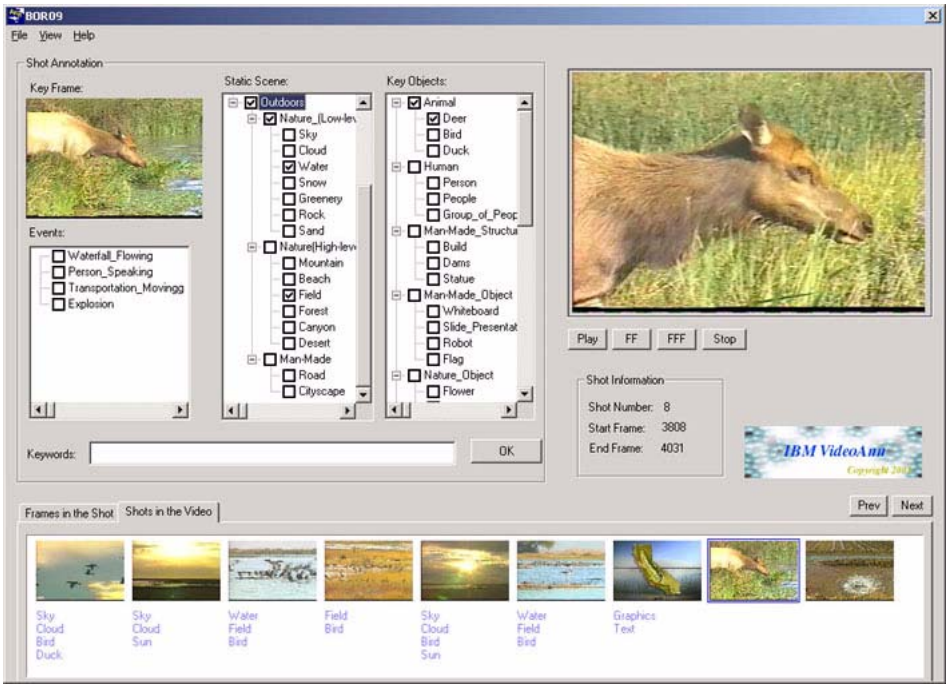

Figure 4. *Example of a video annotation system [IBM02]. This example illustrates various facilities for annotating a single media object. Note that all of the characteristics relate to abstract aspects of the media: that is, the image's contents are considered outside any particular context of use. If we were interested in describing the stance of the ears, the clarity of the eyes or the animal's gait, a totally different keyword ontology would be required.*

In terms of our example in Section 2, the use of hierarchical annotations is limited. There must be a specific ontology that is rich enough to describe the nuances of a particular symptom set but flexible enough to describe the symptoms in the content of the given application. As we can see from Figure 4, the process of working with a complete ontology is often tedious: only three classes of animal are shown (of which deer is a pretty obvious choice, given the alternatives!). As this example unintentionally illustrates, the definition of a complete and comprehensive ontology is not an easy task, and the definition of a usable interface to tag objects based on that ontology is even more difficult.

While hierarchical annotations may themselves consist of either continuous or discrete media objects, in all practical applications they consist of a collection of text strings.

3.2 Peer-Level Annotations

Our work looks at annotation as a means for providing dynamic, conditional content. Rather than being used to locate a video object, our annotations are used to clarify and augment object content. Our annotators are consumes of the document's contents, rather than intermediaries that model the content for the use of others.

The creation of peer-level annotations is not so much a post-production activity as an iterative process in which various types of annotations can be attached to a base media object. The annotations themselves may be both discrete (text and images) objects and continuous (audio) objects.

The display of peer-level annotations can be user and use dependent. They may also be conditional. For example, consider the video fragment represented by Figure 5. Here we see three views of a single video. In Figure 5(a), we see the base video. All annotations are hidden, either because the user doesn't want to be influenced by them or because the user isn't authorized to see them. In Figure 5(b), the base video and the annotations are shown. In Figure 5(c), the annotations are augmented by a yellow warning object at lower right. Selecting this object, which may only be available for a restricted class of viewers, would pause the base video and show supplemental content of a conditional nature.

3.3 Combining Hierarchical and Peer-Level Annotations

While our work has been focused on the specification and creation of peer-level annotations, there is no reason to fundamentally separate hierarchical and peer-level

Figure 5. *Three Use Cases for Peer-Level Annotations of a Leg Joint Operation. (a) Base video object. (b) Base video with annotations enabled. (c) Based video with annotations and restricted-used content flag.*

annotations. A single media object could contain one or more sets of hierarchical annotations (such as the name of the patient, the type of procedure, the name of the attending physician, etc.) as well as peer level annotations on the procedure found in the video itself. The key distinctions about the two types would be that hierarchical annotations would typically provide information about the video object, while peer-level annotations would provide information about the *use* of the video object.

4. Creating and Editing Peer-Level Annotations

This section considers the facilities available for creating and managing peer-level annotations. We begin with a discussion of the container format used to specify the base media and the annotation set.

4.1 Using SMIL as a Container Format

The fundamental requirement that must be met for the generalized peer-level annotation of video (and nearly all other media types) is that the annotations must not corrupt the associated base media object. This requirement not only is a result of copyright or ownership restrictions, but also because a single media object may need to be used many times, in many contexts.

In order to satisfy this requirement, we use the SMIL 2.0 language [BR04] as the basic container format for encapsulating the video object and its annotations. SMIL has a number of advantages as a container format: it contains a rich set of media timing and activation primitives; it provides a simple and flexible layout architecture; it provides both system- and user-test attributes for content control; and it provides a rich hypermedia architecture for support temporal links to associated content. SMIL is also widely deployed: it is already available on over 700,000,000 browsers and players, ranging from telephones to supercomputers.

A discussion on the specifics of using SMIL as a container language for peer-level annotations is given in [Bul03].

4.2 Workflow for Annotation Creation

An annotated video is created as an iterative process that consist of previewing a base video object and then extending it with various types of annotation objects.

Figure 6 illustrates the base workflow for creating and editing annotation objects:

- A base video object is selected for annotation. The video may be in any standard video encoding format and it may be located either locally or across a network. (Note that since the video is not being altered, it does not need to be stored locally. The user performing the annotations does not need write access to the object, nor is the object's copyright violated during the annotation process.)
- The base video is sent to a media previewing engine appropriate for its encoding. At the same time, a copy of the video is sent to an annotation editor. The annotation editor is able to control the previewer and is able to recover time codes and individual media frames.
- Depending on the annotation needs, various types of annotations can be attached to the video, under a variety of constraint situations, including:
  - Various types of media can be attached to the video object. These include continuous media or discrete media (text or images). Hybrid media (animated vector graphics) may also be attached.
  - Several types of timing constraints can be applied to the annotation insertion: objects can be inserted as preempting media, the activation of which will cause the base video to pause, or the annotations can be inserted as companion media, the activation of which runs in parallel with the media object presentation.

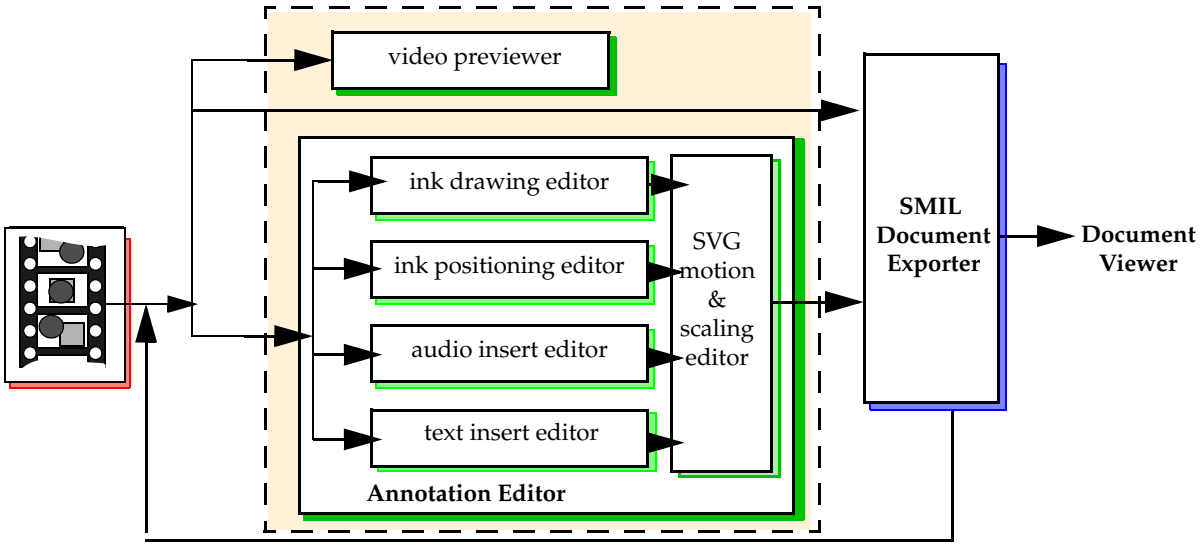

Figure 6. *Workflow Diagram for Capturing Annotations.*

  - The display of the annotations can be conditional: an indication of the presence of the media object is given (but activation depends on user interaction), or annotation objects are automatically displayed.
  - Individual annotation objects can be restricted to certain classes of users or to specific combinations of system facilities (such as on a type of device or when used inside a local environment).
- After annotations are specified, they can be animated from with the annotation editor. Animations consist of SVG motion paths and temporal scaling operations that are applied to individual annotations.
- The result of the annotation and the video presentation is exported to a SMIL 2.0 document. This document can also be used as the basis for downstream editing of the annotated document.
- The document can be used in any conventional SMIL 2.0 player. (Note that some SMIL players, such as IE-6's HTML+Time do not provide full support for SMIL linking, but all other annotations work as specified.)

Many of the steps in the workflow are independent of the annotation creation device. That is, a pen-based system is not needed for most operations. Special use is used of the pen and digital ink facilities available on the TabletPC for capturing ink annotations. At present, these are converted to SVG objects, which are saved and then transparently imported into the SMIL document structure. Early prototypes of the system used essentially similar functionality by allowing a number of predefined drawing objects to be included as annotations rather than relying on custom inked shapes.

5. The Ambulant Annotator

The workflow described in Section 4.2 has been implemented as part of the AMBULANT ANNOTATOR project [Amb04]. The AMBULANT ANNOTATOR takes base ink functionality provided on the TabletPC and integrates it into a SMIL annotation editor. This section illustrates how individual peer-level annotation tasks can be completed using the ANNOTATOR's editing interfaces.

The general editing interface provided by the AMBULANT ANNOTATOR is shown in Figure 7.

5.1 Inserting Text Annotations

Text annotations can be supplied by dragging the text icon onto the display space. Depending on the temporal moment, a text link will be added to the presentation. The target of the link, the duration of the link and the impact of the link on the presentation can be adjusted by entering a set of properties into the link's property tabs. (See Figure 8.)

5.2 Inserting Audio Annotations

Audio annotations are inserted by dragging the image icon to the image area. (As with text annotations, the icons are anchored in standard positions at top right.) When an audio annotation is inserted, the user is given the option of having the audio annotation be rendered in parallel with the base media (starting at the time of the insertion) or as a preemptive object. Preemptive objects are modelled in SMIL using the *excl* element and as a *priority class* of a higher precedence than the video.

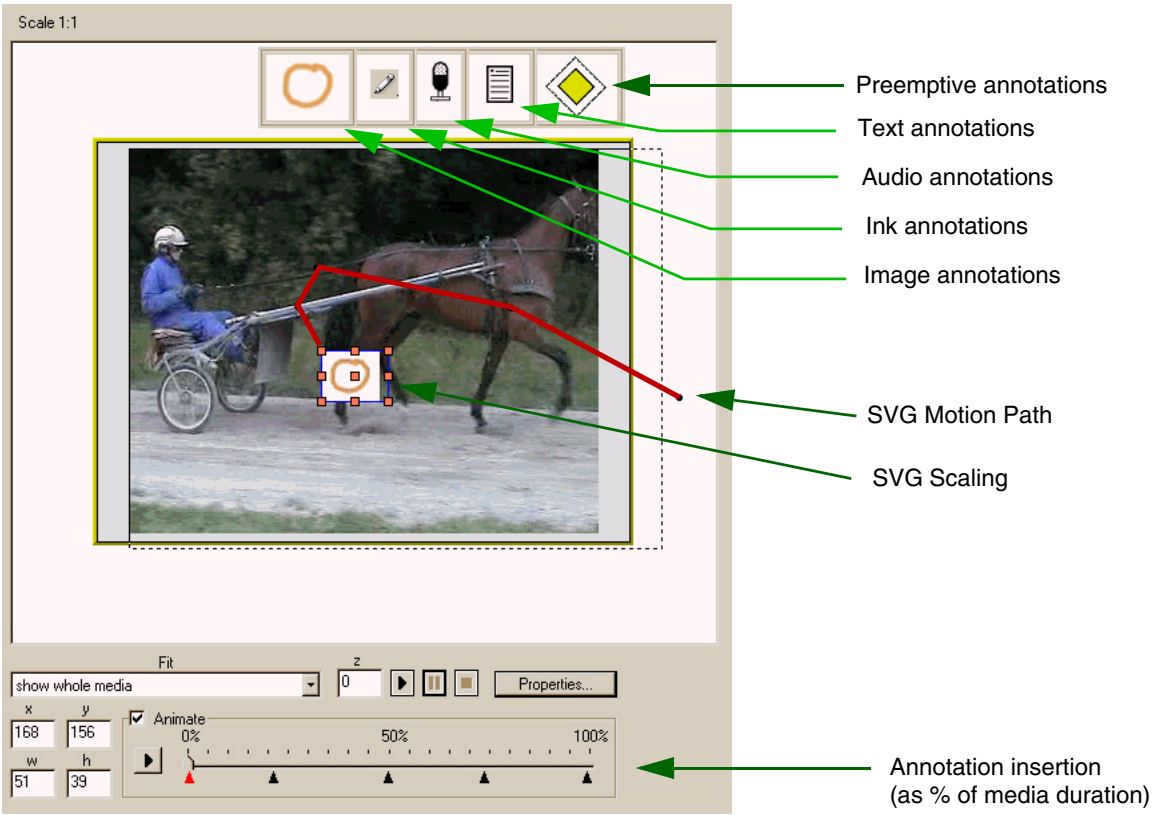

Figure 7. *General Interface for Specifying Annotations.*

5.3 Inserting Image Overlay Annotations

An image overlay can be inserted onto the video object in an overlay plane. It is added to the document at the temporal moment of the insertion (the moment at which the video is paused). The image can then be scaled and moved along with the video using the Annotator's animation facilities (see Section 5.5).

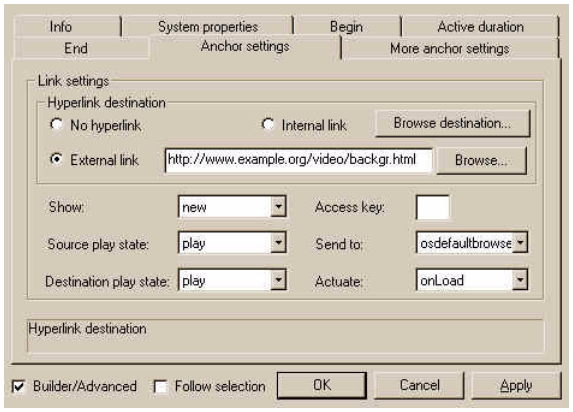

Figure 8. *Options for Text Annotations.*

5.4 Inserting Ink Annotations

Pen annotations can be captured by the drawing device (if available). The presentation is paused and the ink icon is selected. Once an set of strokes is drawn, they are converted to SVG paths and saved as an SVG object. The object is then included in the presentation as if it were an overlay image.

5.5 Animating Motion and Size

Once a set of animations has been defined, the placement and size of the annotations can be adjusted using an animation editor integrated in the environment. The resulting animations are encoded as SVG motion paths and SVG scaling operators. The animation editor allows the annotation to be selected and then positioned/resized based on the total time of the video object. This works reasonably well for short video objects; for longer objects, a direct time code or higher-order video partitioning scheme may be more appropriate.

6. An Example SMIL Encoding

Figure 9 illustrates a SMIL encoding for a video object that is combined with a series of annotations. The main timing portion of the document contains an exclusive element containing a high priority overlay and a parallel component that contains both the base video and an overlay image. Since the element containing the video starts at time '0', this is the starting point of the presentation. If, during the video, the user selects the Audio Icon, then the higher-priority content is rendered. Otherwise, the video and the overlay are presented. Note the motion path for the overlay.

7. Status and Future Work

The primary goal of our work has been to construct a testbed video annotation tool that can be used to annotate arbitrary videos with various types of annotation content. The first phase of this work is to consider single videos, but in later work we expect to annotate arbitrary SMIL 2.0 base documents.

Our current environment makes use of manual editing and placement of annotations. While may useful results can be obtained manually, there are several sets of convenience features that would enhance the annotation process:

- *Integrated editors for audio*: at present, audio objects can be inserted into a document, but these objects are produced outside of the annotation system. A light-weight system for creating and editing audio annotations would be useful.
- *Automatic motion path tracking*: we currently provide facilities for moving and scaling annotations (either ink, SVG objects or overlay images) based on user selected frames in a presentation. An object recognition system with auto-tracking could enhance the production process.
- *Automatic slideshow reduction*: a fairly simple extension would be to generate a slideshow based on the base video object and its annotations. We do not support this functionality at this time.

```
[1] <?xml version="1.0"?>
[2] <smil xmlns="http://www.w3.org/2001/SMIL20/Language">
[3]   <head>
[4]     <layout>
[5]       <topLayout id="MainWindow" backgroundColor="white" width="400" height="300">
[6]         <region id="audio"/>
[7]         <region id="Video" left="24" width="352" top="9" height="288" z-index="1"/>
[8]         <region id="PenInk" left="12" width="376" top="0" height="300" z-index="2"/>
[9]         <region id="TextIcon" left="320" width="20" top="5" height="30" z-index="3"/>
[10]        <region id="AudioIcon" left="290" width="20" top="5" height="30" z-index="3"/>
[11]      </topLayout>
[12]    </layout>
[13]  </head>
[14]  <body>
[15]    <excl id="Videos" dur="indefinite" fillDefault="freeze">
[16]      <priorityClass id="ExtraInfo" peers="defer">
[17]        <audio src="important.mp3" region="audio" begin="AudioObj.activateEvent"/>
[18]      </priorityClass>
[19]      <priorityClass>
[20]        <par begin="0">
[21]          <video src="baseVideo.mpg" region="Video" />
[22]          <img id="AudioObj" region="AudioIcon" src="AudioIcon" begin="2"
                    dur="9s" fill="remove" />
[23]          <img id="Overlay1" region="PenInk" begin="1.55" dur="8.7s" fill="remove"
                   src="Images/SVG-O1.gif" customTest="Annotations"
                   left="64" width="26" top="146" height="21" fit="fill">
[24]            <animateMotion values="(64 146);(75 123);(146 140);(160 120);
                  (134 115);(133 115)"
                  keyTimes="0;0.4013;0.5705;0.74;0.93;1" fill="freeze" dur="8.7s"/>
[25]            <animate attributeName="width" values="26;70;66;89;60;47"
                  keyTimes="0;0.4013;0.5705;0.74;0.93;1" fill="freeze" dur="8.7s"/>
[26]            <animate attributeName="height" values="21;66;77;51;51;51"
                  keyTimes="0;0.4013;0.5705;0.74;0.93;1" fill="freeze" dur="8.7s"/>
[27]          </img>
[28]        </par>
[29]      </priorityClass>
[30]    </excl>
[31]  </body>
[32]</smil>
```

Figure 9. *SMIL 2.0 Example Document with Annotations.*

In addition to the collection of single annotations, we will also integrate these annotations with domain specific knowledge and a set of domain-specific user roles. It is clear that for medical multimedia examples, some of the information may be protected, it may be restricted in terms of location of use and it may be context sensitive. We feel that all of these aspects can be accounted for by using — or extending — the declarative functionality embedded within the SMIL language. (Using a declarative base, while not always convenient, does provide the best guarantee for information reuse within or outside the project.)

The integration of domain-specific interface techniques is also of interest. In particular, we are interested in capturing descriptive information based on gestures or non-text input. We feel that this is important, since continuous media objects do not lend themselves to text annotation.

One of the practical issues with using SMIL as an annotation base is that there were no public-domain open source SMIL players available. This means that constructing an annotator requires not only the logic to intercept user commands and to generate the appropriate SMIL documents, but it also requires building a full SMIL player engine to support document previewing and presentation.

Our group at CWI is addressing this issue by building an open-source, public domain SMIL 2.0 player: the Ambulant Player [DWC01] project started in early 2003, and is expected to produce a fully compliant SMIL 2.0 player by early 2004.

We expect public binary releases of the initial versions to be available by early 2004. The most recent status of both of these project is available at the Ambulant web site (www.ambulantPlayer.org).

8. Acknowledgements

Our research into temporal annotation interfaces is supported under the TOPIA-II project. Implementations of the annotation system are based on work by A. Uginet, K. Kleanthous, J. Jansen and K.S. Mullender. The development of the Ambulant SMIL player is made possible by a generous grant from the NLnet foundation. The medical requirements specified in this paper have been the result of discussions with our partners at the Erasmus Medical Center in Rotterdam and the Dierenkliniek, Emmeloord, The Netherlands.

References

[DWC01] BULTERMAN, D.C.A, JANSEN, A. J., KLEANTHOUS, K., BLOM, K. and BENDEN, D.: The Ambulant SMIL 2.0 Open Source Player, *Proc. ACM Multimedia 2004*, October 2004, New York.

[Amb04] The Ambulant SMIL 2.0 Annotator, http://www.cwi.nl/projects/Ambulant/ .

[Bul03] BULTERMAN, D.C.A.: Using SMIL 2.0 to Encode Interactive, Peer-Level Annotations, *Proc. Doc Engineering 2003*, Grenoble, November 2003.

[BR04] BULTERMAN, D.C.A. and RULTEDGE, L.: *SMIL 2.0: Interactive Multimedia for the Web and Mobile Devices*, Springer Verlang, Heidelberg and New York, 2004.

[DSP91] DAVENPORT, G., SMITH, T.A. and PINCEVER, N.: Cinematic Primitives for Multimedia, *IEEE Computer Graphics and Applications*, July 1991.

[DWC01] DECLERCK, T., WITTENBURG, P. and CUNNINGHAM, H.: The Automatic Generation of Formal Annotations in a Multimedia Indexing and Searching Environment, in *Proceedings of the ACL/ EACL Workshop on Human Language Technology and Knowledge Management*, pp 129-136, 2001.

[FQA88] FURUTA, R., QUINT, V. and ANDRE, J.: Interactively Editing Structured Documents, *Electronic Publishing* 1(1), 1988.

[HvOR01] HARDMAN, H.L, V. OSSENBRUGGEN, J. and RUTLEDGE, L.: Multimedia Meets the Semantic Web, CWI Report, June 2001

[MD89] MACKAY, W.E. and DAVENPORT, G.: Virtual Video Editing in Interactive Multimedia Applications, *Comm. ACM*, 32 (7), July 1989, pp. 802-810.

[IBM02] IBM CORP.: IBM Video Ann, Annotator Demonstration Application, 2002.

[Mar97] MARSHALL, C. C.: Annotation: From Paper Books to the Digital Library, *Proc. 2nd ACM Conf. on Digital Libraries*, July 1997.

[Mic02] MICROSOFT, CORP., The Windows Journal, Commercial Software, http://www.microsoft.com/windowsxp/ tabletpc/evaluation/overviews/ pctools.asp#journal, 2002.