Using SMIL to Encode Interactive, Peer-Level Multimedia Annotations

Dick C.A. Bulterman
CWI: Centrum voor Wiskunde en Informatica
Kruislaan 413
1098 SJ Amsterdam, The Netherlands
+31 20 592 43 00
Dick.Bulterman@cwi.nl

ABSTRACT

This paper discusses applying facilities in SMIL 2.0 to the problem of annotating multimedia presentations. Rather than viewing annotations as collections of (abstract) meta-information for use in indexing, retrieval or semantic processing, we view annotations as a set of peer-level content with temporal and spatial relationships that are important in presenting a coherent story to a user. The composite nature of the collection of media is essential to the nature of peer-level annotations: you would typically annotate a single media item much differently than that same media item in the context of a total presentation.

This paper focuses on the document engineering aspects of the annotation system. We do not consider any particular user interface for creating the annotations or any back-end storage architecture to save/search the annotations. Instead, we focus on how annotations can be represented within a common document architecture and we consider means of providing document facilities that meet the requirements of our user model. We present our work in the context of a medical patient dossier example.

Categories and Subject Descriptors

H.4.3 [**Information Systems Applications**]: Communications Applications - *Information browsers*. 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

Design, Documentation, Experimentation, Languages.

Keywords

Annotation, SMIL, Medical Systems, Horses.

1. INTRODUCTION

The annotation of text and media content is a well studied area. The reasons for such annotation include classification of information for indexing and retrieval and the capture of secondary information for use in (future) information content localization. In these cases, annotations become a layer of meta-content that abstracts aspects of the base documents contents.

Text annotation (both formal and informal) has been around as long as the printed word has been available as a mass-distribution device. Annotation of audio and video is a by-product of the digital era, with the work of Lippman and Davenport et al [7],[12] at MIT setting the initial standard. 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 [8]. The facilities available for media editing led to research in the area of media annotation with various classes of metainformation [9],[14]. The goal of metainformation markup was 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 [11].

The research described in our paper takes a different view of annotations: that of conditional content that can be used to selectively enrich or expand the contents of the base document. Our use of annotation is not to provide an abstract model of the content, but to expand the content itself. The content in our work consists of a coordinated collection of media items that form a composite presentation. It is the composite presentation that is being annotated, rather than any of the individual media objects. This has the result than an annotation encoding structure needs to be developed in which annotations are attached to the composite document. In order to distinguish our use of annotation from indexing/retrieval/semantic uses, we have adopted the term *interactive, peer-level* annotations.

This paper looks at document-related requirements for supporting the annotation of composite collections of continuous and discrete media objects. Using a medical patient dossier as the basis for our research, we consider the requirements for composite-content annotation in an interactive viewing setting. We then explain how the event-based features and layout facilities of the SMIL 2.0 language can be used to define a general, XML framework for non-invasive annotations. While examples are provided from the medical domain, we argue that the use of SMIL has broad application to many domains. We close the paper with a description of the Ambulant Annotator: an interactive annotation environment for creating and viewing annotations based on the concepts presented in the paper.

2. INTERACTIVE, PEER-LEVEL ANNOTATION: AN EXAMPLE

In order to place our work in a concrete context, we begin with a short example. (Note: this section concentrates on the document modelling aspects of peer-level annotations, not on a particular user interface or delivery system.)

2.1 Application Domain

Our work uses the general area of medical information systems as an application domain. In particular, we are interested in studying the requirements of compiling, maintaining and enriching medical patient dossiers. Figure 1 gives an example from such a dossier.

While the use of annotated multimedia patient dossiers has wide application, we have focused our study on veterinary applications. As is clear for the image in Figure 1, we have concentrated ourselves to studies involving horses.

Why horses? There are several reasons, including:

- *Horses can't talk*: nearly all diagnostic activities involving horses (and most other animals) are based on observing the behavior of the animal, and on the content of images and other medical data. Since the patient can't describe its symptoms, a process of investigation and consultation is often required to determine a diagnosis.
- *Horse diagnosis usually involves consultation*: since a doctor can't discuss his findings with the horse, other experts (often not co-located) are frequently used to confirm a suspected diagnosis.
- *Horses are not very portable*: sending a horse to a consulting specialist is usually more complex than referring a human patient to another doctor. As a result, the effort required to construct, maintain and share a (horse's) patient dossier — which is expensive and time-consuming — is often less expensive and time-consuming than sending the animal itself.
- *Horses usually are not insured*: in Holland, governments, hospital regulations and insurance companies highly influence the application of new technology for medical patients. As a result, "people" doctors need to invest their own time and resources on co-developing electronic patient dossiers. Since veterinary activities are less controlled and more driven by the potential of short-term results, they can afford to work on experimental systems that yield even partial cost and effort savings.

Note that while all of these reasons may seem to border to the trivial, they form an important framework for our work: we tried to involve specialists from local hospitals (for human patients) in our experimentation, but operational constraints in the hospitals and the workload of individual specialists limited their initial interests. To our surprise, veterinarians proved to have greater initial interest in discussing patient requirements.

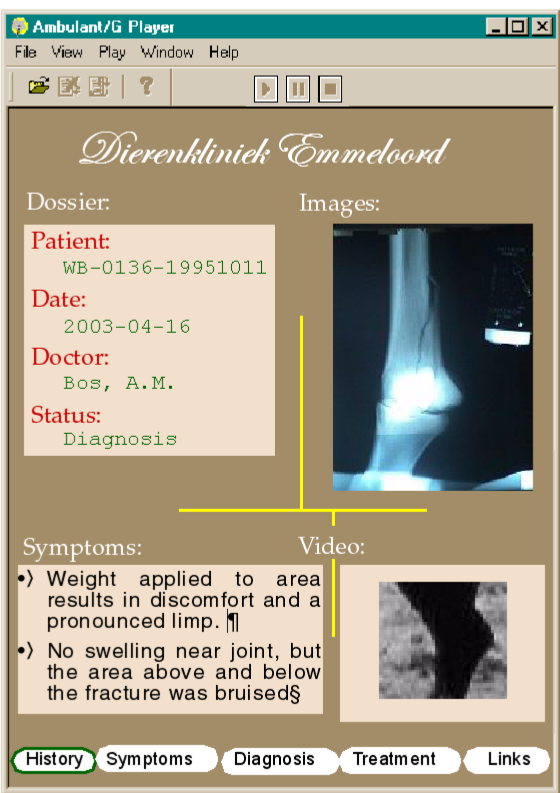

Figure 1. Example of a patient dossier.

2.2 Dossier Workflow

As illustrated in Figure 2, the following steps are typically used to build and annotate an electronic patient dossier:

- *Administrative entry*: as with people, a basic amount of record-keeping is required to enter a patient the system. The goal of our total environment is to build an information server that can be used to access a patient dossier from local and remote locations.
- *Raw media capture*: during initial and extended diagnosis, use can be made of text descriptions of symptoms. In addition, depending on the problem, use can also me made of (collections of) image, video and audio media. This media is typically captured with digital cameras and uploaded to a PC.
- *Initial media processing*: the collected media is typically edited to remove non-essential material. The raw material is typically not saved.
- *Report creation:* a basic report is created, using a template-based multimedia editor. We use the GRiNS editor [17] extended with appropriate presentation templates.

At this point, the basic presentation is available for use in annotation. During annotation, the base document may be extended with audio, text, animation, image and video objects. Links may also be added to the content for either internal navigation or external reference.

2.3 Information Retrieval and Sharing

Information sharing is not a key aspect of our experimental work. Instead, we focus on providing facilities to create the

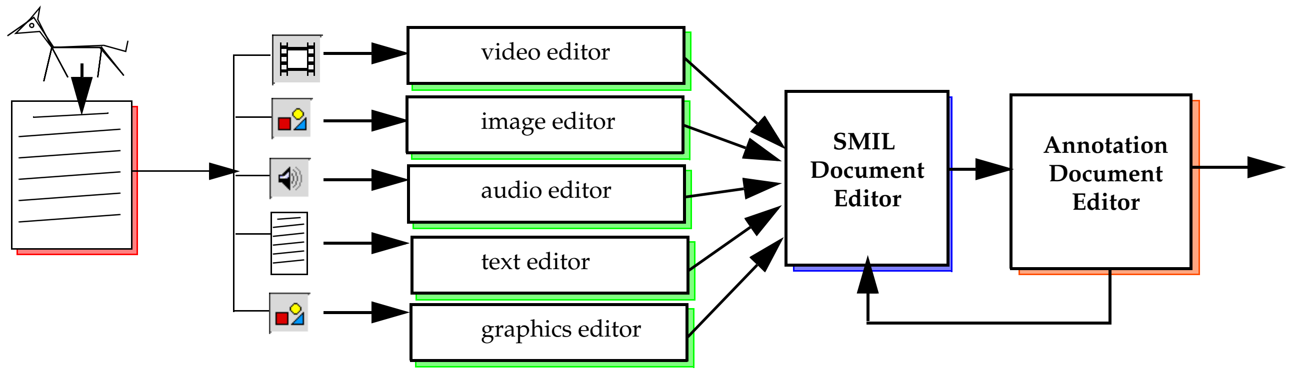

Figure 2. Workflow Diagram for Capturing Annotations.

base document and to extend it with interactive annotations. The feedback we have received from our users is that the detailed annotation of content for use outside the context of a single dossier is typically more effort than is justified. Approaches used in single-media editors, such as the video editing system illustrated in Figure 3, have little practical relevance in building medical dossiers (except for use in archival studies for teaching and education).

3. INTERACTIVE PEER-LEVEL ANNOTATIONS

Interactive, peer-level annotations are collections of conditional content that can be used to expand on information inside a base document. In this section, we consider the document engineering aspects of such annotations. We start with a review of the general functional characteristics of interactive, peer-level annotations and then discuss the requirements (and non-requirements) of documents containing these annotations.

3.1 Annotation Characteristics

These are annotations on top of existing content. The content may be continuous or discrete, the annotations may also be continuous or discrete. (Both types can be cross-applied.)

A classical definition of annotation reads:

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

Annotation of this type is 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.

Our work looks at annotation as a means for providing dynamic, conditional content. Rather than being used to locate

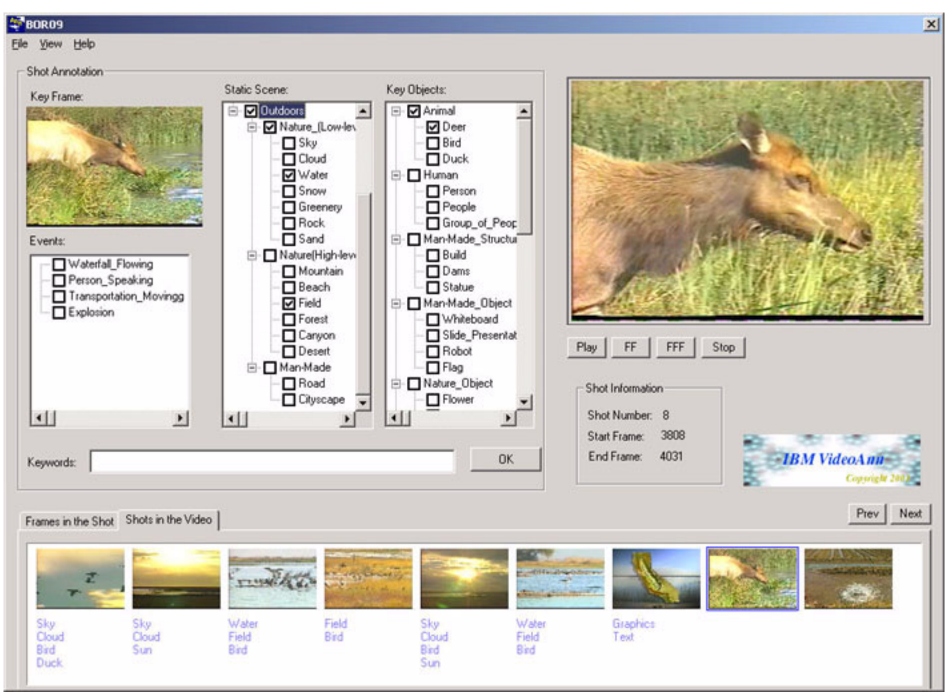

Figure 3. Example of a single-media annotation system [13].

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.

a document, our annotations are used to clarify document content for a human consumer. Our annotators are consumes of the document's contents, rather than intermediaries that model the content for the use of others. Perhaps the key distinction of our annotation is that we annotate collections of media objects with the context of its composite presentation, rather than a single media.[1]

We assume that the base document contains a structured collection of information and contains media that is both discrete (text and images) and continuous. The annotations themselves may also be either discrete or continuous, and each type may be used to annotate any other types. That is, continuous media can be associated with either discrete or continuous base objects: such associations can be associated with a discrete moment in a document, or the annotations can be played in parallel with the base document's contents.

3.2 Document Requirements

Several document requirements need to be met in order to support our peer-level annotations. There are also a set of requirements that are not essential, but which can prove useful in the content of a full annotation application environment. This section considers both sets of requirements.

3.2.1 Primary document properties

The following list gives a set of document requirements that are needed to support interactive, peer-level annotations. This list is based on user requirements for our medical dossiers, but it has broader applicability to other annotated documents. The (summary) list is:

- The document should be digital (this aids searching, sharing, editing);
- The document structure must be exposed (that is, the logical associations among content types must be visible to attach annotations within the correct context);
- The document must support some user role differentiation, both for access control and annotation segmentation (this supports multi-person creation, multi-role viewing and/or partitioning in which a person is allowed to view or allowed to view and annotate);
- The document structure must support annotations on the base document and annotations on the annotations (that is, is must support encapsulation of annotations);
- The document must support interactive annotation while in viewing mode (users don't want to enter a separate application to do annotation, they want to have the annotations added within the primary viewing application);
- focused capture of annotation: don't capture everything, capture everything relevant.
- The document must associate the annotation to the content 'in context' (that is, the annotation will probably have little meaning outside the context)
- The annotations within the document must be inherently editable (with access control)
- The document must allow continuous-media annotations to either pause the associated source media or to occur in parallel with that source content;
- The document should model the structure of the information rather than the structure of a presentation device (that is, no shared whiteboard model is used);
- The document must support conditional viewing of annotations (and it must support selectivity across different sets of annotations).

Note that in this discussion, the storage model of the document (and the encoding syntax) is not of primary importance, as long as the functional requirements are met.

3.2.2 Optional or secondary document properties

There are a number of optional requirements associated with the document architecture that fall under the category 'nice to have, but not essential'. In general, these relate to properties for a particular use of the annotations in their own right, rather than in the context of the dossier:

- The document does not need to logically differentiate annotation content from other content (this follows from our not using the annotations as meta-information but as conditional information).
- The document model does not need to support integration with an annotation management system in which the annotations are stored in a different format;
- The document does not need to support an embedded rights management system (exposing user differentiation and role differentiation is more important than enforcing such distinctions).

A document model that supports these features unobtrusively can have added value, but our experience is that most annotation systems focus on the annotations rather than on the annotated content.

3.3 Environmental Requirements

Annotations typically need to be supported by a runtime environment. In our work, that environment has the following requirements:

- The environment must be focused on the user-centered creation of annotations rather than on any automatic annotation creation. (This is a consequence of the need for experts demanding to be in control of the diagnostic process.)
- In addition to user-centered creation, the annotations must support user-centered analysis (rather than automatic analysis) of annotations. (Automatic analysis of annotations is potentially useful, but presents liability issues beyond the scope of our work.)
- The annotation environment should be focused on sequential, individual analysis of annotations rather than on a cooperative work model of simultaneous annotators modifying the same work. (A 'cooperative' annotation system does not provide new annotation requirements, only general conflict management requirements; it is not a user requirement.)

In general, for our medical applications, the simpler and more local the annotation activity, the more confidence it inspired among practitioners.

[1] It is possible to annotate a single item, but this is not the usual case.

4. USING SMIL TO SUPPORT INTERACTIVE, PEER-LEVEL ANNOTATION

There are several options available to users wishing to encode the document shown in Figure 1: they could use HTML, Powerpoint, Flash, Quicktime or a number of other document formats. When it comes to supporting annotations (rather than simple presentations), none of these formats meet the requirements specified in “Document Requirements” on page 4. One format that does support these requirement is the W3C’s SMIL 2.0 format [5]. In this section, we discuss the relevant characteristics of SMIL that make it a useful basis for annotation, we discuss the advantages of using SMIL 2.0 and we discuss the disadvantages of using SMIL. While our discussion relies heavily on code fragments drawn from SMIL, a complete description of the language is beyond the scope of this contribution. Interested readers should consult [4].

4.1 Relevant Characteristics of SMIL 2.0

SMIL 2.0 provides an attractive language to support annotation requirements because of several key features:

- *SMIL is an XML language*: this is not a benefit in-and-of itself (XML documents tend to be much larger than their binary counterparts), but it does mean that the result annotated document can be post-processed by a host of tools.
- *SMIL is extensible:* SMIL’s namespace architecture allows authoring and player environment to extend the language in a clean and predictable manner. While it is tempting to think that the use of namespaces will allow for some sort of dynamic DNA-like growth of tools to support all possible combinations of languages, this is not true: an implementation is still needed that understands the extensions. If nothing else, however, namespace-based extensions will at least guarantee that the base document should play on all compliant SMIL players.
- *SMIL supports non-invasive integration of annotations on top of existing document objects*: SMIL does not store media information, it stores information *about* media objects. The SMIL document can be annotated without changing any of the underlying information bits. This is essential for medical (and most other) applications.
- *SMIL is a declarative language*: all of the timing and layout constructs in SMIL are defined in terms of temporal and spatial specifications rather than embedded scripts. As a result the presentation can be constructed and extended ‘on the fly’ by any SMIL savvy editor. (Not having to understand underlying scripts is a key — if under appreciated — aspect of SMIL functionality.)
- *SMIL support extensive timing, layout, linking and content control options:* these facilities allow all of the requirement of “Document Requirements” on page 4 to be supported. (More on this below.)

SMIL has more characteristics that are generally attractive to Web-based applications, but these are not specifically relevant to annotations.

4.2 An Example SMIL Encoding

Figure 4 sketches a SMIL encoding for the application fragment presented in Figure 1. The description contains a layout section defining rendering positions for 8 objects, and a body section containing both temporal and structural references to the contents of the document. In the examples below, we refer this as the *Base Document*.

4.3 Advantages of SMIL 2.0 as an Annotation Base Format

In “Relevant Characteristics of SMIL 2.0” on page 5, several motivations for using SMIL were presented. (Some of these relate to other XML languages, but only SMIL supports the complete set.) In this section, we look at apply specific SMIL functionality to solving particular annotation requirements. Where appropriate (and as space permits), we will illustrate this functionality by showing SMIL code excepts.

SMIL’s functionality is divided across several functional areas (each containing one or more modules). We will discuss SMIL’s applicability to interactive, peer-level annotations by functional area. Please note again that this section considers only document content issues, not authoring or editing issues.

4.3.1 Timing support

Of all of the issues associated with the specification of interactive, peer-level annotations, timing is the most critical. Annotations form an extension to an existing ‘story’, both in terms of the presentation of new content and the specification of the moment at which that content gets introduced.

In text/image annotations (such as that provided using pen-like scribbles on top of an existing document [15]), the temporal moment of the appearance of the annotation is typically not critical. (Spatial relevance is usually more critical.) This is a by-product of the timeless modelling of text and images used in annotation systems.[2] Most medical personnel are trained to evaluate image data in a certain order (left to right, top to bottom) when scanning an x-ray image. Only after a full scan is attention drawn to specific areas. (If there are multiple problems, these also seem to be check in a regular — thus quasi-temporal — manner.) The temporal constraints of defining the moment for annotating continuous media objects is more easily defined, although the behavior at the moment of annotation can vary: the annotation can pause the base document while an annotation is presented, or the annotation can be presented in parallel with the information in the base document.

The general process that can be used to attach an annotation to a media object (or collection of objects) is to wrap the objects from the base document in a SMIL <par> container, and then to wrap this combination of elements in another <par> element along with the annotation.

[2] The timeless nature of text is, of course, an illusion. Text is consumed in terms of ordered collections of letters, words, sentences, paragraphs and sections. This ordering is not random, and as a result it is possible to identify a temporal moment at which annotations associated with text content should appear. This moment is right before the associated content is about to be consumed by the reader.

SMIL supports the timing requirements for annotation using several different constructs. We discuss these in terms of four annotation activities:

- discrete media annotations to discrete content
- discrete media annotations to continuous content
- continuous media annotations to discrete content
- continuous media annotations to continuous content

(This list does not cover all of the cases of temporal relationships for annotations, but these cases are representative.)

Discrete media annotations to discrete media contents

A discrete media object (such as an image or text) has a duration that is either explicitly specified in the document or which is derived from context of the presentation. In a similar way, a discrete annotation can have a specified or an implied duration.

Figure 5 illustrates how a simple annotation of the X-ray object *x2* on line 23 in Figure 4 can be accomplished. (Note: layout and other aspects are considered later.) The original statement can be extended with (in this case) an image annotation; this annotation defaults to inheriting the duration of its peer, but the annotation can also be qualified to be a sub-duration of the peer. The outer timing container will ultimately control the display duration of both the base document's image and annotation.

```
[23]    <img id="x2" src="xray136-2.png"
             region=Images" />
```

(a) The original statement.

```
[23]  <par>
[24]    <img id="x2" src="xray136-2.png"
             region=Images" />
[25]    <img id="x2A1" src="xray136-2.png"
             region=ImagesA" />
[26]  </par>
```

(b) After attaching the annotation.

```
[23]  <par>
[24]    <img id="x2" src="xray136-2.png"
          region=Images" />
[25]    <img id="x2A1" src="xray136-2.png"
          region=ImagesA" begin="3s" dur="7s" />
[26]  </par>
```

(c) After qualifying the relative presentation timing.

Figure 5. Attaching a simple annotation.

Discrete media annotations to continuous content

A discrete media object can be added to a continuous media object in a manner similar to the process just described for discrete base document content. The principal difference is that the continuous

```
[1] <?xml version="1.0"?>
[2] <smil xmlns="http://www.w3.org/2001/SMIL20/Language">
[3]   <head>
[4]     <meta name="title" content="Medical Dossier"/>
[5]     <layout>
[6]       <topLayout id="MainWindow" backgroundColor="coffee" width="400" height="500">
[7]         <region id="audio"/>
[8]         <region id="Background" left="0" width="400" top="0" height="500"/>
[9]         <region id="Title" left="0" width="400" top="0" height="50" z-index="2"/>
[10]        <region id="Dossier" left="10" width="185" top="85" height="170" z-index="3"/>
[11]        <region id="Images" left="330" width="60" top="85" height="190" z-index="1" fit="meet"/>
[12]        <region id="Symptoms" left="6" width="225" top="335" height="110" z-index="4"/>
[13]        <region id="Video" left="335" width="160" top="245" height="120" z-index="5"/>
[14]        <region id="Links" left="6" width="388" top="465" height="35" z-index="6"/>
[15]      </topLayout>
[16]    </layout>
[17]  </head>
[18]  <body>
[19]    <par >
[20]      <img src="b_image.jpg" region="Background" />
[21]      <par>
[22]        <ref id="Patient" src="Patient136.html" region="Dossier"/>
[23]        <img id="x2"      src="xray136-2.png"region=Images" />
[24]        <ref id="S-136"   src="Symptom136.html" region="Symptoms"/>
[25]        <video id="v136"  src="Video136.mpg"    region="Video" />
[26]      </par>
[27]      <img id="links2"     src="links.symp.png"  region="Links" >
[28]        <area id="lh1" href="http://www.sfp+p.nl/history136.smil"
[29]          shape="rect" coords="3,3,25,83" />
[30]        ...
[31]        <area id="ll1" href="http://www.sfp+p.nl/links136.smil"
[32]          shape="rect" coords="340,3,25,83" />
[33]      </img>
[34]    </par>
[35]  </body>
[36]</smil>
```

Figure 4. SMIL 2.0 Fragment for Figure 1.

```
[25]<video id="v136"  src="Video136.mpg"
          region="Video" />
```

(a) The original statement.

```
[25]<excl id="v136A1">
[26]  <priorityClass id="Annotate" peers="defer">
[27]    <img id="x2A1" src="xray136-2.png"
          region=”VideoA" begin=”8s” dur=”5s” />
[28]  </priorityClass>
[29]  <priorityClass id="baseDocVideo">
[30]    <video id="v136"  src="Video136.mpg"
          region="Video" />
[31]  </priorityClass>
[32]</excl>
```

(b) After adding an ‘interrupting’ annotation.

Figure 6. Attaching an *interrupting* annotation.

nature of the base content may influence how the annotation is to be presented:

- if the annotation refers to a contiguous portion of the continuous media, the annotation can be displayed ‘in parallel’ to the base content;
- if the annotation refers to a singe moment in the continuous media, the base content should be paused and frozen while the annotation is displayed.

The first case is essential similar to that shown in Figure 5. The second case is illustrated in Figure 6. In this figure, the video in the base document’s line 25 is extended using a SMIL *exclusive* element. The <excl> has the semantic that only one of its children can be active at a time:

- the use of priority classes allows one class of content (the annotation) to pause another class (the base content);
- the video is played until the point at which the annotation should be presented;
- the annotation then “interrupts” the video;
- the video continues after the annotation has been presented.

(Note that this could also have been accomplished with SMIL linking or using exclusives without priority classes, but both solutions often require more authoring overhead.)

Continuous media annotations to discrete content

Attaching continuous annotations (such as an audio object) to discrete media base content can be accomplished in a manner similar to that shown in Figure 5. Some care must be taken to make sure that the continuous content will be given enough time to complete (that is, if an outer timing container limits the duration of the discrete media, then the audio annotation may be cut off — or never presented. In practice, this is usually avoided because the (human) author can preview the annotation: if a timing conflict occurs, the annotation can be scheduled as interrupting content.

Continuous media annotations to continuous content

Attaching continuous media annotations to continuous media base content has many of the same issues as discussed with Figure 6; a choice needs to be made if the annotations should be presented in parallel with the base content (in which case, a <par> content wrapper is usually sufficient) or if the base content should be paused during presentation of the annotation (in which case the <excl> with priority classes can be used.

4.3.2 Content-control support

The cases for temporal annotation described above can be easily extended to support multi-author, or multi-role annotations. This is accomplished by applying SMIL’s content control facilities.

The basic content control facilities available in SMIL are <switch> element and a set of system test attributes (such as systemBitrate or systemLanguage). The <switch> is a useful construct for annotation, but the system test attributes have only limited applicability. To allow customization of content presentation based on non-system attributes, SMIL provides an architecture for *custom test attributes*. Custom test attributes are dynamic attributes that take the values *true* or *false*. They can be evaluated in-line, or as a predicate on a <switch> statement.

Consider the SMIL fragment in Figure 7. Here we see an application of custom test attributes for the presentation of multi-author comments and the combination of custom tests with a <switch> for multi-role annotations.

Figure 7(a) illustrates a combination of SMIL timing and content control. The image *C1* is annotated by two users (*Dick* and *Jack*). The user agent can determine that none, one or both of the annotations are presented during rendering. If none of the annotations are shown, then the image will be presented for 10 seconds. If the user agent has enabled viewing of Dick’s annotations, the image will be shown for a minimum of 10 seconds or for the duration of Dick’s comment (which ever is longer). Similarly, if Jack’s comments are “enabled”, the image will be shown in parallel with this audio. If both Dick and Jack’s comments are enabled, then the duration of the image will extend

```
<par endsync=”annotations”>
  <img id="C1" src="x2.png"
       region=”Image" min=”10s” />
  <seq id=”Annotations”>
    <audio id="A1" src="A1.mpg" region="Audio"
       customTest=”Dick” />
    <img id=”I1” src=”I2.png” region=”ImageA”
       customTest=”Jack” dur=”10s”/>
  </seq>
</par>
```

(a) An example of multi-user annotations.

```
<par endsync=”annotations”>
  <video id="Z1" src="v1.mpg" region=”Video”/>
  <switch>
    <audio id="D" src="D.mpg" region="Audio"
       custonTest=”Doctor” />
    <audio id="O" src="O.mpg" region="Audio"
       custonTest=”Owner” />
    <audio id="E" src="G.mpg" region="Audio"/>
  </switch>
</par>
```

(b) An example of multi-role annotation.

Figure 7. Attaching an *interrupting* annotation.

to the composite duration of the annotations (or 10 seconds, whichever is longer).

Figure 7(b) illustrates a combination of SMIL timing and <switch>-based content control. The video *Z1* is annotated for three audiences: a doctor, the owner and everyone who is not an owner or doctor. While the video is presented, the <switch> is evaluated based on the role set by the user agent. Note that if no general annotations were desired, the final default (non-guarded) <switch> child could be removed.

4.3.3 Layout support

SMIL layout has always been somewhat of a problem child. Within the W3C, CSS layout has always received more favor, while outside the W3C layout is usually embedded inside of scripted objects. For annotation purposes, SMIL layout does present advantages over other languages because of its simplicity, its extensibility (within a presentation) and its explicit nature.

The principal layout concerns that exist within the context of an annotation application are:

- the ability to place annotation information on top of an object without changing the base object;
- the ability to place annotations 'near' base objects;
- the ability to place annotations in pop-up notes;
- the ability to move annotations along with base objects if the base object is animated; and
- the ability to adjust the spatial characteristics of the annotation if the base object changes.

SMIL 2.0 supports these needs through the application of its hierarchical layout features and through the use of sub-region positioning of objects and annotations.

Figure 8 illustrates the layout implications of attaching an annotation to a base object. In Figure 8(a), we see a video region defined inside of a top-level layout window. (A document may have multiple top-level windows; this allows pop-up note functionality for annotations to be easily supported.) When an annotation is created (such as was done in Figure 6), the dimensions of the original window can be wrapped in a new hierarchical element containing two child regions: the original video window and a video overlay window.[3] These are illustrated in Figure 8(b). Both windows are anchored to the same screen position, so that if the base window is moved, the annotations move along with the base content.

```
<layout>
  <topLayout id="MainWindow" width="400"
      height="500" backgroundColor="coffee" >
      ...
      <region id="Video" left="335" width="160"
        top="245" height="120" z-index="5"/>
      ...
  </topLayout>
</layout>
```

(a) A layout fragment from the base document.

```
<layout>
  <topLayout id="MainWindow" width="400"
      height="500" backgroundColor="coffee" >
      ...
      <region id="VideoC" left="335" width="160"
        top="245" height="120" z-index="5">
        <region id="Video" width="100%"
           height="100%" z-index="1"/>
        <region id="VideoA" width="100%"
           height="100%" z-index="2"/>
      </region>
      ...
  </topLayout>
</layout>
```

(b) A layout fragment expanded with hierarchical regions for annotation.

Figure 8. Defining annotation regions.

The annotation content has been assigned a z-index stacking order that is higher than the base window. SMIL defines that stacking order as being local to the hierarchical region. As a consequence, if content in another window that had a higher stacking order was refreshed or displayed after the annotation in the video window, correct rendering behavior would result.

The scope of the annotation window can also fall outside the dimensions of the original rendering window by having the outer (container) window define the combined area for the annotation and the base content. In the way, the annotation can be place fully/partially outside the screen region used to render the base content.

The location of the parent window can be animated. If there are annotations attached, these can move with the base content. The annotator can also chose to animate only the annotations: in this case, the position and size attributes of the child annotation region can be changed independently of the base content. (Note that sophisticated authoring support is usually required to have these features accessible to users.)

4.3.4 Linking support

Many authors have argued the case for separating links from the content that serve as source or destination anchors for those links. (See [16] for a recent discussion.) SMIL supports this separation from linking anchors and linked content via the <a> and the <area> attributes. Figure 4 gives an example of associating SMIL links with background content. The same structure can be used to introduce link-based annotations.

The primary advantage of SMIL linking over other linking architectures [20] is that SMIL links allow the specification of temporal extents that can be used to control the activation period of a link. In this way, a given anchor may be conditionally active, or an anchor may change its link target over the span of the annotation. This is exceptionally useful.

Figure 9 illustrates the use of temporal anchors for conditional specification of links via the <a> element and the use of timed anchors for time-varying link targets via the <area> element.

[3] SMIL provides facilities for making overlay windows transparent, and controlling their activation to be on-demand. These features go beyond the scope of this paper.

```
[1] <a id="links1" begin="10s" dur="15s"
        href="http://www.sfp+p.nl/a1.smil" >
[2]   <img id="x-ray" src="gasp.png"  region="X" />
[3] </a>
```

(a) Specifying timed layout via the <a> element.

```
[1] <img id="x-ray" src="wow.png"  region="X" >
[2]   <area id="lh1" begin="3s" dur="4s"
          href="http://www.sfp+p.nl/6.smil" />
[3]   <area id="ll1" begin="13s" dur="7s"
          href="http://www.sfp+p.nl/3.smil" />
[4] </img>
```

(b) A linking fragment showing timed activation via <area>.

Figure 9. Defining temporal links.

SMIL links also have a set of activation conditions that allow linked content to pause or augment the base content, and that allow links to be directed into separate viewing windows if desired. For users of SMIL 1.0, linking was the only method available for defining supplemental content in a presentation, but with SMIL 2.0's event-based timing facilities, the <excl> element has replace much of linking's use for activation of content within the scope of one document.

4.3.5 Animation support

Animation support is typically not viewed as a critical element of annotation, but the ability to easily include animated annotation content can be very helpful in expressing complex messages.

A detailed description of how animation can be used to enrich annotations is beyond the scope of this paper. For our purposes, we simply note that the integration of basic SMIL and SVG animation primitives to control the rendering of otherwise static annotation content is a major advantage of use SMIL as a base language.

4.4 Disadvantages of using SMIL for Annotation

The primary disadvantage of using SMIL for supporting interactive, peer-level annotations is that annotations require full support for the functionality in the SMIL 2.0 language. Many commercial SMIL players (such as the RealOne player, HTML+TIME and Quicktime) provide only partial support for SMIL 2.0. Certain functional areas — such as content selection and layout — are not completely supported. This could make actually using SMIL as a target language more of a problem than creating SMIL documents themselves.

5. THE AMBULANT ANNOTATOR

One of the practical, non-document issues with using SMIL as an annotation base is that there are 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 trying to address this issue in two ways:

- *we are building an open-source, public domain SMIL 2.0 player*: the Ambulant Player [2] project started in early 2003, and is expected to produce a fully compliant SMIL 2.0 player by early 2004.
- *we are building an annotation engine on top of the Ambulant player*: the Ambulant Annotator [3] is being built to implement all aspects of the annotation requirements specified in this paper.

The distribution policy for the Ambulant Annotator is still under discussion. We expect binary releases of the initial versions to be available by the end of 2003. The most recent status of both of these project is available at the Ambulant web site (www.ambulantPlayer.org).

6. FUTURE WORK

The primary goal of our work on evaluating SMIL as a base language for supporting medical annotations has been to determine if there are inherent limitations within the SMIL language that preclude its use in annotation work. We have not found any language-based problems that would inhibit this work, although we recognize that our investigation has been of only limited scope. This is due primarily to the difficulty in getting (medical) area experts to think 'out of the box' of the current applications and tools they have available. We expect it will be easier to obtain feedback from potential users once a base level annotation tool has been implemented.

Our study has restricted itself to the use of annotation for presentation enrichment. While we have only looked at the medical domain, we feel that there are strong links that can be made to work in the areas of education, digital libraries, video (and other single-media) annotation and news content annotation and customization. We expect to expand our own focus as a follow-up to our present work.

We also expect to investigate opportunities for combining the use of annotation for presentation enrichment with annotation for content classification (for searching, indexing and semantic analysis). An area of interest is the study of broad-based annotations in a mobile information infrastructure. We look forward to forging partnerships with other research groups to broaden our study.

7. ACKNOWLEDGEMENTS

Our research into temporal annotation interfaces is supported under the TOPIA-II project. 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.

8. REFERENCES

[1] Abowd, G.D., Harvel, L.D. and Brotherton, J.A., Building a Library of Captured Educational Experiences, Proc. Kyoto Int. Conf. on Digital Libraries, 2000.

[2] The Ambulant SMIL 2.0 Player, http://www.cwi.nl/projects/AmbulantPlayer/ .

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

[4] Bulterman, D.C.A., SMIL 2.0: Overview, Concepts and Structure, IEEE Multimedia, December 2001.

[5] Cohen, A., Bulterman, D.C.A., et al, The SMIL 2.0 Specification, http://www.w3.org/AudioVideo/ .

[6] Van Dam, A., ReMarkable Texts, NSDL proposal, Brown University, 2002.

[7] Davenport, G., Smith, T.A. and Pincever, N., Cinematic Primitives for Multimedia, IEEE Computer Graphics and Applications, July 1991.

[8] 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.

[9] Furuta, R., Quint, V. and Andre, J., Interactively Editing Structured Documents, Electronic Publishing 1(1), 1988.

[10] Halasz, F., Reflections on NoteCards: Seven Issue for the Next Generation of Hypermedia Systems, CACM, July, 1988.

[11] Hardman, H.L, v. Ossenbruggen, J. and Rutledge, L., Multimedia Meets the Semantic Web, CWI, June 2001

[12] MacKay, W.E. and Davenport, G., Virtual Video Editing in Interactive Multimedia Applications, Comm. ACM, 32 (7), July 1989, pp. 802-810.

[13] IBM Corp, IBM Video Ann, Annotator Demonstration Application, 2002.

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

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

[16] Neto, R.B., Izeki, C.A., Pimentel, M. de Graca, Fortes, R,P., Tuong, K.N., An Open Linking Service Supporting the Authoring of Web Documents, Proc. *DocEng '02*, Nov. 8-9, McLean VA, 2002, pp. 66-73.

[17] Oratrix, The GR*i*NS/SMIL-2.0 Pro Editor, http://www.oratrix.com/ .

[18] Phelps, T.A. and Wilensky, Multivalent Annotations: Anywhere, Anytime, Any Type, Every Way User-Improvable Digital Documents, CACM 43(6), June 2000.

[19] Pimentel, M. da Graca, Abowd G.D. and Ishiguro, Y., Linking by Interacting: a Paradigm for Authoring Hypertext, Proc. ACM Hypertext '00, 2000.

[20] De Rose, S., XML Linking, ACM Computing Surveys, 31(4es) December 1999.

[21] Wactlar, H.D., "Informedia - Search and Summarization in the Video Medium, Proc. Imagina 2000, Monaco, 2000.

[22] W3C, Timed Text Working Group. http://www.w3.org/AudioVideo/TT/ .