

Running head: VIDEO TOOLS

Let SMIL be your umbrella:
Computerized tools for automating presentation and analysis of
digital video in behavioral research

Sujai Kumar

Kevin Miller

University of Illinois at Urbana-Champaign

Preparation of the software tools reported here was supported by NSF grant REC-0089293.

Address correspondence to: Kevin F. Miller, Department of Psychology and Beckman Institute,

University of Illinois, 405 N. Mathews, Urbana, IL 61801 (email: kevinmil@uiuc.edu)

Abstract

Video-based techniques have become central to many areas of social science research, although their use has been limited by the expense and complexity of tools for working with video information. New standards for the representation of digital video make the manipulation of video for observational research a far less time-consuming and expensive process. We provide an overview of SMIL, a cross-platform markup standard, and how it can be used to edit, synchronize, caption and present video clips without modifying the original digital video files. We also present TransTool – a free Windows program which can generate SMIL files for playing video clips of interest along with captions and codes. TransTool can also be used as a transcribing and coding tool that synchronizes video and text such as transcripts. These tools greatly facilitate tasks such as creating video events with multi-language transcripts, showing synchronized views of the same event, and the incorporation of video clips into presentations and web pages.

Let SMIL be your umbrella: Computerized tools for transcribing, analyzing,
and presenting digital video

Video data provide compelling access to behavioral phenomena, enabling researchers to study events that are too quick to comprehend in a single viewing and helping them to develop reliable observational methods and apply them to complex interactive phenomena. Yet the use of video techniques in behavioral research has been limited by a variety of obstacles, including expensive hardware, the complexity of specialized editing tools, storage, portability, and problems of access to archives of video data. The advent of digital video technology has dramatically reduced the barriers to entry into video-based research techniques for data collection.

Once video data are collected, problems of transcribing, editing, storing, and coordinating those data remain. New software standards for representing video on computers promise to overcome this final hurdle to the easy incorporation of video data into a variety of research paradigms and reports.

This paper will provide an introduction to the Synchronized Multimedia Integration Language (SMIL, pronounced "smile"), an increasingly accepted cross-platform markup language for working with video and other multimedia information. We will provide examples of how SMIL can be used to simplify common research tasks. We will also present TransTool, a free Windows application that simplifies the transcription and indexing of video files, and also generates SMIL output for use in presentations and analysis tasks. In addition to being a useful tool in and of itself, TransTool can serve as a prototype that demonstrates how the combination of SMIL, embedded software for playing digital video, and rapid application development

environments (such as Visual Basic), can be used to automate a large number of research tasks involving digital media.

What is SMIL?

SMIL is a markup language (like HTML) rather than a programming language (such as C or Java). It uses plain text to tag certain types of multimedia content, creating a set of commands that are recognized by media players such as RealPlayer, QuickTime player, or Internet Explorer. SMIL files do not themselves store media; they only reference them (in the same way an HTML page references applets, images and other elements). The function of a SMIL file is to specify exactly how, where and when each media element should be used in a multimedia presentation. Although multimedia files come in many forms, SMIL files are plain text files, a fact that facilitates their creation, modification, and incorporation into other programs.

SMIL is an XML-based standard of the World Wide Web Consortium (W3C), designed to provide a common framework for the integration of different video and audio formats (Hoschka, 1997). SMIL provides a presentation and timing model that can work with a variety of media formats, including several compressed video and audio formats (including RealMedia, QuickTime, and MPEG, depending on the player used). Background information and the full specification can be found on the official SMIL website (W3C, 2002a).

Working with SMIL: The fool's mate example

This discussion of SMIL will be framed in terms of a simple example (all files can be accessed at www.psych.uiuc.edu/~kmiller/smil, along with production details and other tips) that shows what can be done with this language. We chose the RealMedia compressed format for this example as it provided the optimal video quality for its size, but we could have used other formats as well (see Appendix A). The code listings in this paper all conform to the SMIL

standard as specified by W3C (2002b), and will play without modification in RealOne player (RealPlayer 9) which may be downloaded for free from www.real.com. Some minor modifications to the syntax in these files are required to play them in the QuickTime player (these changes and tips on producing files that will play in both players are described in Appendix A).

We videotaped a short chess game, illustrating the four-move "Fool's Mate." The event was videotaped from three camera angles, one focusing on the board, and one focusing on each player. Commentaries were also produced, in both written and spoken forms. The following sections will describe how these media sources can be synchronized and combined to produce increasingly sophisticated representations of this event.

Playing part of a clip

Figure 1 shows a very rudimentary SMIL file for playing a file called "board.rm" that shows the chessboard camera angle. The example shows the final move made by Black (the player using the black pieces), starting 16 seconds into the clip and ending 11 seconds later (27 seconds into the clip). The file is a plain text file that can be typed in any text editor (for example, Notepad or Wordpad on Windows, SimpleText on Macintosh systems, vi on Unix) and then saved with an extension of ".smi" or ".smil."

```
<smil>
  <body>
    <video src="board.rm" clip-begin="16s" clip-end="27s"/>
  </body>
</smil>
```

Figure 1. Playing a short segment of a video clip using SMIL

SMIL files must begin and end with the `<smil>` and `</smil>` tags, respectively. The body of the file (between the `<body>` and `</body>` tags) describes the content to be played and where and how it should be presented. In this case, the single body line tells the player the type of content (video), where the content is to be found (in a file called "board.rm" in the default directory), where in the file to begin playing (16 seconds into the clip) and when to stop (27 seconds into the clip). The total duration of this edited sequence would thus be 11 seconds. As is often the case with SMIL, there are other means to accomplish the same effect. In this case, one could replace `clip-end="27s"` with `dur="11s"` (the "dur" attribute specifies the duration of the clip).

Although this is a very simple SMIL file, it could be called from a PowerPoint presentation or Web page to present a specific segment from a longer video clip. Because compressed digital video is difficult to edit, the ability to extract a clip with a few commands can come in handy.

Coordinating perspectives

One of the most appealing applications of SMIL for behavioral science researchers is its ability to coordinate the presentation of multiple views of the same event. The ability to look at the same event from different points of view is critical to understanding complex interactional processes, such as those that go on in conversations, classrooms, and other social situations. The use of SMIL to coordinate different perspectives will be demonstrated with our three camera angles on the fool's mate example.

```
<smil>
<head>
  <layout>
    <root-layout width="1080" height="240"/>
```

```

<region id="video_left" width="360" height="240" left="0" top="0"/>
<region id="video_center" width="360" height="240" left="360" top="0"/>
<region id="video_right" width="360" height="240" left="720" top="0"/>
</layout>
</head>
<body>
  <par dur="30s">
    <video src="black.rm" clip-begin="1.09s" region="video_left"/>
    <video src="board.rm" clip-begin="0s" region="video_center"/>
    <video src="white.rm" clip-begin="1.10s" region="video_right"/>
  </par>
</body>
</smil>

```

Figure 2. Playing multiple video clips simultaneously using SMIL

Figure 2 introduces a powerful feature of SMIL: the ability to define playback regions. The general layout of the playback environment is described between the `<head>` and `</head>` tags. Within an overall layout window that is 1080 pixels wide and 240 pixels high, three regions are defined with a height of 240 pixels and a width of 360 pixels each. All three are positioned at the top of the screen, and each is positioned at a different distance from the left so that they appear in a straight line from left to right.

The body of the SMIL file again specifies the clips to be played and their starting and ending points, but now each video tag also includes a region directive that specifies where the clip should be played ("video_left", "video_center", and "video_right" respectively).

The `<par>` and `</par>` tags specify that the media elements contained within are to be played in parallel, and not sequentially. The `dur` attribute within the `<par>` tag limits the length of the presentation to exactly 30 seconds, irrespective of the lengths of the individual clips within it. One of the many advantages of this type of presentation is that the video recordings need not

be precisely timed to begin with because we can use SMIL to start playing each of the clips at a different point (using the *clip-begin* attribute) in order to maintain synchrony.

Adding voice-overs

The next example shows how one can add voice-overs, transcripts and commentary to any SMIL presentation. In Figure 3, we took the three camera view (Fig. 2) and added the voiceover file as an audio media source (the rest of the SMIL file remains the same). This presentation does not start the video clips until 5 seconds of the audio track have already been played, so that the introductory commentary can be presented. This delay is specified with the *begin* attribute, which tells the player the exact time after which the clip should begin to play (note that this is different from the *clip-begin* attribute that specifies how many seconds into the clip it should begin playing). The types of audio files that can be played back will be determined by the media player that you choose.

```
<par dur="35s">
  <video src="black.rm" begin="5s" clip-begin="1.09s" region="video_left"/>
  <video src="board.rm" begin="5s" clip-begin="0s" region="video_center"/>
  <video src="white.rm" begin="5s" clip-begin="1.10s" region="video_right"/>
  <audio src="voicetrack.rm"/>
</par>
```

Figure 3. Adding an audio track and delaying video playback using SMIL

Captioning in multiple languages

A particularly useful feature of SMIL is the ability to flexibly present any of a variety of captions in synchrony with video information. This is especially useful for researchers who wish to present examples in languages unfamiliar to their viewers. The segments can be transcribed in the original language and the transcripts synchronized to the video (e.g., using TransTool, as

described in the next section). The transcripts can then be translated into another language, independent of the video. As long as each statement in the original (synchronized) text file is replaced by its equivalent in the translation, the second-language captions will play in synchrony with the video.

Transcripts or captions can be stored in a separate file in a plain text format such as RealText or QuickTime Text (see University of Illinois, 2000, for an overview of using QuickTime Text). Figure 4 shows a very simple RealText file that defines a text window 560 pixels wide and 100 pixels high, and specifies a time point when each line of text should be shown within that window. The `` and `<center>` tags specify basic display information, but see RealNetworks (2002) for additional text display options.

```

<window width="560"
      height="100"
      bgcolor="black"/>
<font color="white" face="arial" size="+2">
<b>
<center>
<time begin="00:00:00.00"/><clear/>
The Fool's Mate is often tried on newcomers to the game of chess.
<time begin="00:00:05.00"/><clear/>
White begins, and opens up his King to a fatal attack.
<time begin="00:00:12.50"/><clear/>
Black moves a pawn to give her Queen room to move out.
<time begin="00:00:19.50"/><clear/>
White moves the other pawn forward, leaving a clear line of attack on the King.
<time begin="00:00:31.46"/><clear/>
Black checkmates using her Queen.
<time begin="00:00:34.90"/><clear/>
It is rarely a good idea to move the pawns on f2, g2 and h2 so early in the
game as the King normally castles on this side. If the pawns have been moved,
they can no longer offer him adequate protection.

```

Figure 4. Sample RealText file for Fool's Mate example.

This RealText file can be played on its own in the RealOne player as a standalone window with timed text, but its utility becomes apparent when it is combined with a video clip. The SMIL code in Figure 5 plays back a RealText file as a media source, in synchrony with the other video and audio media in the previous example (Fig. 3). Figure 6 shows what this file would look like when played back.

```

<smil>
<head>
  <layout>
    <root-layout width="1080" height="350"/>
    <region id="video_left" width="360" height="240" left="0" top="0"/>
    <region id="video_center" width="360" height="240" left="360" top="0"/>
    <region id="video_right" width="360" height="240" left="720" top="0"/>
    <region id="text_subtitle" width="560" height="100" left="260" top="250"/>
  </layout>
</head>
<body>
  <par dur="55s">
    <video src="black.rm" begin="5s" clip-begin="1.09s" region="video_left"/>
    <video src="board.rm" begin="5s" clip-begin="0s" region="video_center"/>
    <video src="white.rm" begin="5s" clip-begin="1.10s" region="video_right"/>
    <textstream src="text.rt" region="text_subtitle"/>
    <audio src="voicetrack.rm"/>
  </par>
</body>
</smil>

```

Figure 5. Presenting text and audio commentaries using SMIL



Figure 6. Sample screenshot of three coordinated views and subtitled text using SMIL

In order to display text as a caption or subtitle, a new region needs to be defined where the text track can be displayed. This is done by defining region "text_subtitle" in the layout section of the SMIL file. This region is positioned 260 pixels from the left and the top of the window, and is 100 pixels high and 560 pixels wide. The `<textstream>` tag in the main body of the SMIL file tells the player which text file should be played along with the video clips, and the region in which it is to be played. Multiple caption or transcript tracks can be presented by defining new regions and assigning different RealText files to each of them.

Putting it all together

The final example shows how SMIL can be used to produce a unified presentation from multiple sources of video, audio, and text information, obviating the need to use a video editing program to produce presentation materials.

In Figure 7, a single video region is defined because we want the presentation to appear within a single frame. The `<seq>` and `</seq>` tags ensure that all the media clips listed within will be played one after another in sequence, creating a perfectly synchronized, edited presentation. The outermost `<par>` and `</par>` tags are used to play the voiceover track in parallel with the edited video clips.

In order to give the viewer enough time to see each chess player, the presented event is longer than the actual event. Shots of a player sometimes continue after the other player has already begun their move. The presentation then switches to show the other player's move from the beginning. Thus, what took 30 seconds in real life takes 36 seconds in this video presentation.

```

<smil>
  <head>
    <layout>
      <root-layout width="360" height="240"/>
      <region id="video_main" width="360" height="240" left="0" top="0"/>
    </layout>
  </head>
  <body>
    <par>
      <seq>
        <video src="black.rm" begin="2s" clip-begin="0s" dur="2.2s"
region="video_main"/>
        <video src="board.rm" clip-begin="0s" dur="4.2s" region="video_main"/>
        <video src="black.rm" clip-begin="5s" dur="3.2s" region="video_main"/>
        <video src="board.rm" clip-begin="8s" dur="4.5s" region="video_main"/>
        <video src="white.rm" clip-begin="13.5s" dur="2s" region="video_main"/>
        <video src="board.rm" clip-begin="14s" dur="3.7s" region="video_main"/>
        <video src="black.rm" clip-begin="18s" dur="6s" region="video_main"/>
        <video src="board.rm" clip-begin="23s" dur="3.5s" region="video_main"/>
        <video src="white.rm" clip-begin="27s" dur="4.5s" region="video_main"/>
        <video src="board.rm" clip-begin="29s" dur="0.5s" region="video_main"
fill="freeze"/>
      </seq>
      <audio src="voicetrack.rm"/>
    </par>
  </body>
</smil>

```

Figure 7. Editing multiple clips into a single frame using SMIL

Other capabilities

SMIL supports several additional capabilities beyond those demonstrated here. Although the examples shown so far have dealt with the issue of creating static multimedia content, SMIL can be used to define transitions (such as fades, wipes, etc.), create animations, switch between different media elements based on user settings (such as default language, region, bandwidth, etc.), provide hyperlinking within a presentation, and create interactive content that responds to user events such as mouse clicks and keyboard inputs. Further examples can be found at www.psych.uiuc.edu/~kmiller/smil, along with links to additional SMIL resources.

All of the video presentation and editing tasks shown here could have been accomplished using any video editing suite. SMIL has two major advantages over solutions that rely on expensive hardware or software editing systems. The first is that there is a significant saving of time, processing power and media storage as SMIL files only store instructions rather than copies of the digital media themselves. With a traditional video editing approach, changing captions or selecting clips would require editing and re-rendering to produce new files, whereas with SMIL, only the text instructions would have to change.

The second advantage is particularly significant for those developing a number of presentations for research purposes, such as multiple video stimuli for different conditions in an experiment. Rather than use complex application programming interfaces to work directly with raw digital video data, SMIL provides a convenient way to access and control what is shown with just the help of a few lines of text that can be generated from within any programming or scripting language.

SMIL also has an interesting consequence for multimedia files that are accessible from the internet. Video and audio files can not currently be indexed by search engines, but SMIL files

are text files that can be indexed. Thus SMIL can provide a method for indexing multimedia information for retrieval from the internet.

In the next section, we will describe an application that demonstrates the ease with which video data can be manipulated by a software program.

Analyzing and Presenting Video Data with TransTool

TransTool is an open source Windows application that was developed by the Cognitive Development Lab of the University of Illinois (Kumar, 2002) as a prototype for automating research tasks such as the transcription, coding and presentation of digital video data. It is essentially a database application that links individual records with specific time points in a video clip. Unlike bigger packages with more comprehensive video manipulation features, TransTool was designed to be a lightweight tool that meets the most basic needs of researchers who analyze video clips. It can be used for transcribing and coding video files, as well as for generating some simple types of SMIL files. The transcription and coding functionalities demonstrate how media players can be precisely controlled from within software programs, and the SMIL generation feature shows an example of how a database of transcript information can be used to automatically create captioned video presentations in different languages.

The application was designed with Visual Basic 6.0, a popular rapid application development environment. It has been tested with Windows 98, 2000 and XP running on Pentium II, III and 4 processors with clock speeds of 200 MHz or better. The program works with video files in the MPEG, MPEG4, and RealMedia formats, and requires the RealOne player. The combination of Visual Basic and the RealPlayer ActiveX control allowed us to create a prototype application in very little time as the methods and properties of the ActiveX control gave us complete and precise cueing and playback information for the video clip.

The next two sections will describe how TransTool can be used for transcribing and coding a video clip, and how the software can be used to generate SMIL presentations that display the video along with the transcript and the codes based on the user's choices.

Transcribing features

TransTool's main window (Figure 8) provides a video window and a data table with entries linked to the video clip by means of a timestamp. The video window is really a RealP layer window that has its own VCR-like controls for starting, stopping, pausing and scrolling along any MPEG or RealMedia clip. When the program starts, the user must first load a video clip using the File menu. A blank transcript is opened by default, but the user can open an existing transcript database file. Each row or record of the transcript file contains 1) a timestamp that corresponds to a precise time point in the video, 2) a column labeled "Actor", 3) a column for transcribed text, and 4) other user-defined columns for codes or any other information the user wishes to synchronize with the video.

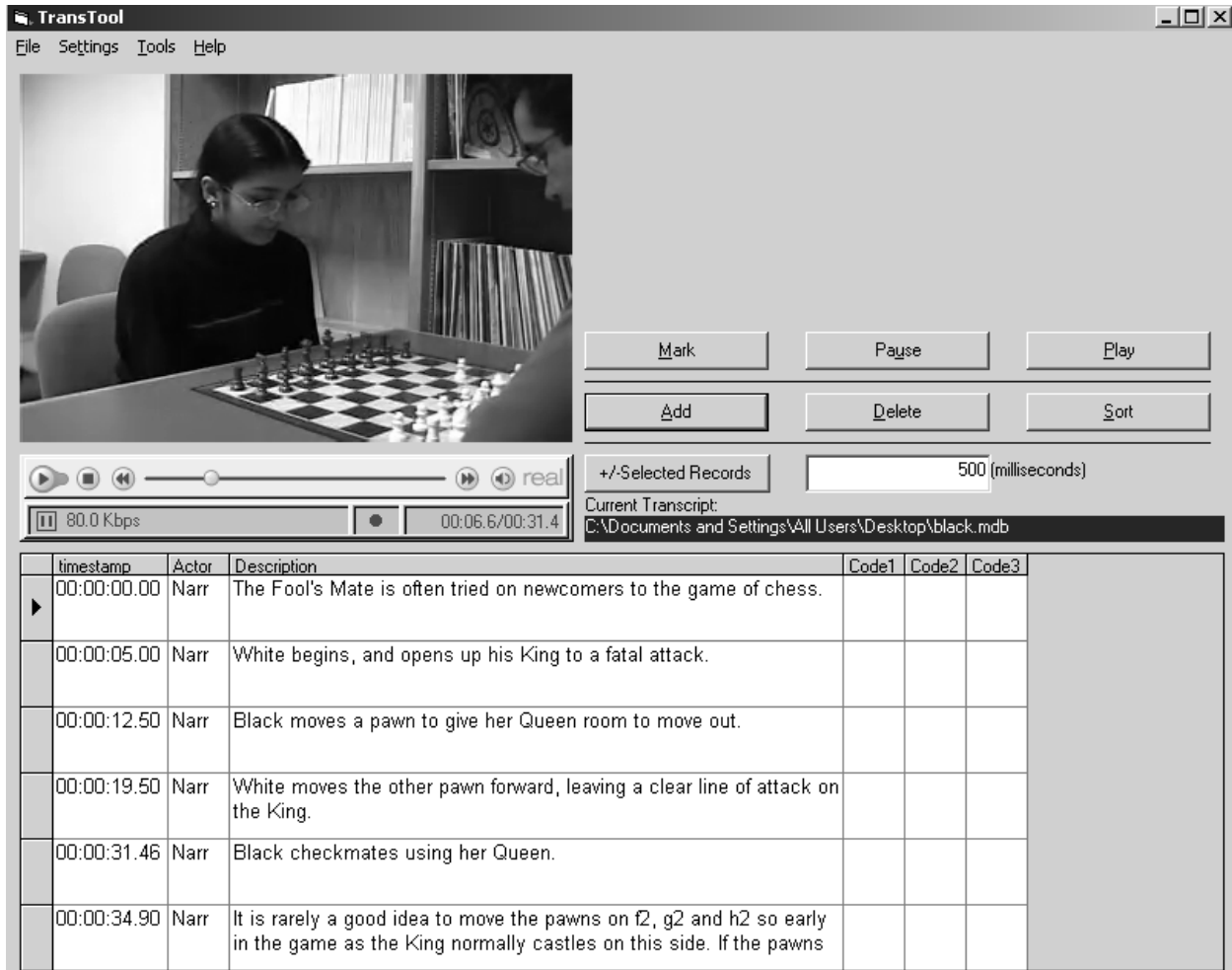


Figure 8. TransTool layout

The buttons on the right are used to work with the video clip and the data table. The first set of buttons 1) mark the current record with the timestamp of the video clip's current position, 2) play a video clip from the exact point specified in the current record, and 3) pause the video clip. The lower set of buttons allows the user to manipulate the data table by adding, deleting, or sorting the records according to the timestamp. All the buttons have keyboard shortcuts that make it convenient for a transcriber to pause and play the video clip, cue it instantly to a specific

time-point, and replay a clip repeatedly from a specific point (in order to catch hard-to-hear audio information).

The Menu bar on top provides access to other TransTool functionalities. Apart from the basic loading and saving of media clips and transcripts, the File Menu also allows users to import text data from other applications into new transcripts, and to export existing transcripts as tab-delimited text files that can be read by most spreadsheet and analysis packages.

Further details and a copy of the program are available on our website at www.psych.uiuc.edu/~kmiller/transtool.

Automatic SMIL generation

TransTool generates two kinds of files on the basis of a transcript. The first is a RealText file (see Fig. 4. for an example) that includes each record of the transcript along with the timestamp for that record. The second is a SMIL file that creates two regions, one for the video clip and the other for the RealText file created in the previous step, and specifies the parts of these two media sources that should be played back in synchrony. This SMIL file looks very much like the file shown in Figure 5. Both files are needed along with the original video clip in order to create an integrated presentation.

We have so far only implemented SMIL generation for the case where a transcript needs to be presented along with a video clip. However, the principle is the same in all cases as some text is static (header information such as the `<smil>` and `<head>` tags) while the rest is dynamic (picked up from the database).

Summary & Conclusions

SMIL provides a flexible and standards-based format for synchronizing multimedia sources. It can be used to coordinate the presentation of different views of the same event,

present video synchronized with audio and text information, and combine segments of different files to produce a single integrated presentation. Most importantly, SMIL is free and requires nothing more than a text editor and a player (most are available online for free) to get started. Several graphical user interface tools for authoring SMIL also exist (see W3C, 2002a for a partial listing), but the language is easy enough to learn and the examples in this paper should provide a reasonable set of templates for some common research needs.

SMIL provides important new capabilities to organize, annotate, and present multimedia information. Coupled with inexpensive techniques for recording and storing video records, these capabilities make complex social interactions increasingly accessible to researchers for analysis and presentation.

References

Hoschka, P. (1997). Toward synchronized multimedia on the web. *World Wide Web Journal*, March 1997. Retrieved November 21, 2002 from <http://www.w3journal.com/6/s2.hoschka.html>

Kumar, S. (2002). DVGuide: Analysis Tools. Retrieved December 5, 2002 from University of Illinois, Cognitive Development Laboratory Web site: http://www.psych.uiuc.edu/~kmiller/dvguide/analysis_tools.htm

RealNetworks Inc. (2002). RealNetworks Production Guide. Retrieved December 5, 2002 from <http://service.real.com/help/library/guides/realone/ProductionGuide/HTML/realpgd.htm>

University of Illinois (2000). Overview of QuickTime Text. Retrieved December 5, 2002 from University of Illinois at Urbana-Champaign, Center for Instructional Technology Accessibility Web site: <http://cita.rehab.uiuc.edu/quicktime/Qtext.html>

W3C (2002a). Synchronized multimedia. Retrieved November 21, 2002 from <http://www.w3.org/AudioVideo/>

W3C (2002b). SMIL 2.0 Timing and Syntax. Retrieved November 21, 2002 from <http://www.w3.org/TR/smil20/smil-timing.html>

Appendix A

SMIL modifications for different players

Specifying the SMIL version

If just the `<smil>` tag is used at the start of the SMIL file, then the default version is assumed to be SMIL 1.0. In order to use features specified in SMIL 2.0 (see <http://www.w3.org/TR/smil20/>), replace the `<smil>` tag with `<smil xmlns="http://www.w3.org/2001/SMIL20/Language">`. The *xmlns* attribute specifies the XML namespace to be used for the document.

QuickTime player

In order to present the simplest possible SMIL files in this paper, we used code that conforms to the standard and plays correctly in the RealOne player. However, the QuickTime player (one of the more popular media players available) is not as flexible about some SMIL default values, and needs to be modified in order to work correctly.

All clip-begin and clip-end times should be prefixed by "npt=". For example, one would write:

```
clip-begin="npt=3s"
```

The "npt" prefix stands for "normal play time" (see <http://www.w3.org/TR/smil20/extended-media-object.html> for more information). A SMIL file with this modification will be playable in both the RealOne and QuickTime players.

QuickTime also allows several extensions to SMIL that can be accessed by setting the *xmlns* attribute inside the `<smil>` tag to `<smil xmlns:qt="http://www.apple.com/quicktime/resources/smilextensions">`. For more information, see <http://www.apple.com/quicktime/authoring/qtsmil.html>.

Internet Explorer

SMIL files do not play as standalone files in Internet Explorer the way they do in the RealOne and QuickTime players. Internet Explorer supports SMIL timing, animation and transition modules, but they have to be invoked from within HTML tags in a fairly complicated way. See <http://msdn.microsoft.com/library/en-us/wmsrvsdk/htm/introductiontosmil.asp> for more details.