SMIL 2.0

Part 2: Examples and Comparisons

Dick C.A. Bulterman
Oratrix Development, Amsterdam

This article is the second part of a two-part series on SMIL 2.0, the newest version of the W3C's Synchronized Multimedia Integration Language. This part looks at simple and complex examples of SMIL 2.0's use, and compares SMIL with other multimedia formats.

The W3C's Synchronized Multimedia Integration Language (SMIL) 2.0 specification [1], released in August, 2001, is a collection of modules that describe how media objects (in their most general form) can be integrated and scheduled for presentation in local and network-based environments. The focus of SMIL is not on data encoding (that is, it does not specify codecs for any one media type, such as video or audio), but on media integration: SMIL specifies how components relate temporally and spatially during a presentation.

The SMIL 2.0 specification defines approximately 50 modules, grouped into 10 major functional areas. The functional areas, and their scope, are:

- *animation*: a collection of modules that can be used to define time-varying values to elements and attributes within the containing document. For example, animation elements can cause the background color of a layout region or the position of media objects to change over time.
- *content control*: a collection of modules that can be used to conditionally include media items (or SMIL structure hierarchies) in a presentation based on various system- and user-defined test attributes. For example, the SMIL switch element can be used to define a set of alternative encodings, one of which will be selected at run-time, based on the value of a system parameter (such as bandwidth, screen size, language preference, etc.).
- *layout*: a collection of modules that are used to explicitly manage rendering space and audio/visual resources in a presentation. For example, the *regPoint* and *regAlign* attributes allow collections of objects to be centered around a particular point on the screen.
- *linking*: a collection of modules that allow both SMIL temporal and HTML-like non-temporal links to be defined and managed. For example, an anchor can be defined to point at all or a part of an object, for all or part of its rendering time, and link to internal or external media objects when activated.
- *media objects*: a collection of modules that define the elements and attributes associated with the definition and inclusion of media elements into a presentation. For example, the *ref* element can be used to reference external objects based on their URI.
- *meta-information*: a module compatible with the Dublin Core [2], to encode meta-information on an entire presentation or any sub-part of that presentation. For example, the meta-information modules specify how author, title, date, copyright, etc., can be added to a presentation.
- *structure*: a collection of modules that define the basic structure of native SMIL documents. The structure module defines how the SMIL element is encoded, and its content.

- *timing and synchronization*: an extensive collection of modules (spanning over two hundred pages in the SMIL 2.0 specification) detailing the elements, attributes and behavior of the SMIL timing model.
- *time manipulation*: a collection of modules that define how time can be manipulated within a presentation. For example, some profiles allow time to elapse faster or slower than normal play time.
- *transitions*: a collection of modules that add visual transitions to a presentation. For example, these modules can define how objects can appear using fades, wipes and other forms of visual transitions.

In part 1 of this two-part series [3], we looked in detail at various aspects of the SMIL specification and the underlying SMIL timing model. In this second part, the focus will be on the textual structure of SMIL in its various implementation profiles.

SMIL 2.0 Implementation Profiles

SMIL 2.0's modules have been designed so that relevant groups of modules can be combined into *implementation profiles*. At present, three profiles have been implemented and distributed, and a fourth pseudo-profile is also in circulation.

Figure 1 shows the set of SMIL 2.0 profiles currently implemented. Of these, the SMIL 2.0 *Language Profile* and the SMIL 2.0 *Basic Profile* were defined as part of the SMIL 2.0 specification. Several vendors have released SMIL 2.0 Language profile players and tools (including Oratrix [4] and RealNetworks [5]). The SMIL Basic profile has been used as the basis for the wireless multimedia specification by the 3GPP consortium [6].

The XHTML+SMIL profile is still under development by the W3C, but an early implementation is already available on tens of millions of desktops because of its integration into Microsoft's InternetExplorer-5.5 and 6.0 browsers [7].

The fourth implementation profile of SMIL 2.0 is not technically a language profile, but rather a module integration: the W3C's Structured Vector Graphics (SVG) specification [8], which has been integrated by several vendors in their media formats and tools, uses SMIL Animation as the basis for specifying timing and XML element/attribute animation.

SMIL 2.0 Document Formats

The goal of the profile model was to be able to customize the integration of SMIL's functionality into a wide variety of XML-based languages without requiring language authors to learn totally new timing semantics for each variant.

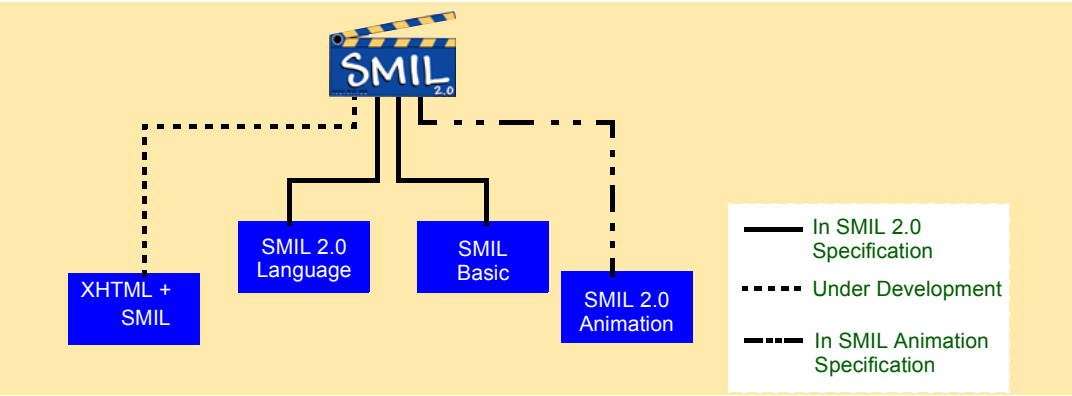


Figure 1: *SMIL 2.0 Implementation Profiles*.

SMIL 2.0 Language Profile

The SMIL 2.0 Language Profile is the follow-on format of the SMIL 1.0 specification. The language profile implements nearly all of SMIL 2.0's modules.

The structure of a presentation in the Language Profile is shown in Figure 2. Each document begins with an XML encoding string, followed by the SMIL element. The SMIL element may have several attributes, the most important of which is the XML namespace declaration. For SMIL 2.0 Language Profile documents, this is:

```
xmlns="http://www.w3.org/2001/SMIL20/Language"
```

There may be additional namespaces declared within the document (for use by particular SMIL 2.0 Players).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
    ...
    </layout>
    <transition id="fade" type="fade" dur="1s"/>
    <transition id="push" type="pushWipe" dur="0.5s"/>
  </head>
  <body>
    ...
      <img src="..." transIn="fade"/>
      ...
      <video src="..." transOut="push"/>
    ...
  </body>
</smil>
```

Figure 2: *General Structure of SMIL 2.0 Language Profile Document.*

The SMIL element contains an optional <head> element and a <body> element. The <head> element may contain layout and transitions declarations, as well as meta-information and custom test attribute definitions. The <body> contains collections of timing, media and other declarations.

SMIL 2.0 Basic Profile

The SMIL 2.0 Basic Profile is a collection of modules that allow SMIL to be supported on minimal devices, such as telephones and PDAs. The structure of a document in the SMIL Basic Profile is essentially similar to that of the Language Profile. (Among other aspects, they share the same namespace definition.) The differences are in the module functionality set they support and the complexity of the document structure allowed. SMIL Basic supports the lowest complexity modules of each of the SMIL 2.0 functional groups, such as basic layout, basic timing, and basic content control. The Basic Profile defines the minimum baseline for documents that can still be considered to be members of the SMIL language family; individual implementations may choose to add other parts of SMIL 2.0 functionality if appropriate to their devices.

SMIL 2.0 XHTML+SMIL Profile

The XHTML+SMIL profile defines a collection of modules that allow SMIL timing and presentation semantics to be integrated in other—typically non-temporal—XML documents. Where the SMIL 2.0 Language Profile provides a general time framework for all elements and attributes in a document, the XHTML+SMIL Profile provides a limited scope time framework for individual XHTML elements or group of elements.

An implementation of the XHTML+SMIL profile is available in Microsoft's Internet Explorer (IE). IE-5 provided SMIL support based on SMIL 1.0 using Microsoft's *time behaviors*, and IE-5.5 and IE-6 provide support based on the SMIL 2.0 specification using the *time2 behaviors*. (Microsoft uses the marketing name HTML+TIME for their IE implementations.) The basic structure of SMIL timing in IE is shown in Figure 3. Since the document base is HTML, the outer document element is not <smil> but <html>. The <html> element contains an XML namespace declaration for the "t" prefix used to denote SMIL's HTML extensions. The <head> contains a short-hand style definition for the *time2* behavior, plus an import tag pointing to the implementation. The <body> contains a sequence of images that are displayed according to SMIL's *sequential time container* semantics.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns:t = "urn:schemas-microsoft-com:time">
 <head>
  <meta http-equiv="content-type" content="text/html;charset=ISO-8859-1"/>
  <meta name="generator" content="GRiNS Pro for SMIL 2.0, v2.0 win32 b103"/>
  ...
  <style type="text/css">
    .time {behavior: url(#default#time2); }
  </style>
  <?import namespace = t urn = "urn:schemas-microsoft-com:time"
      implementation = "#default#time2" />
 </head>
 <body>
  <div id="Player_Window" style="position:absolute;
      overflow:hidden;left:0;top:0;width:480;height:240;
      background-color:black;">
    <t:seq id="m1">
     <t:img dur="3" id="img1" src="..." style="..." />
       ...
     <t:img dur="9" id="img4" src="..." style="..." />
    </t:seq>
  </div>
 </body>
</html>
```

Figure 3: *Basic Internet Explorer structure as example of the XHTML+SMIL profile*.

IE provides support for most of SMIL 2.0 timing semantics, plus support for animation, transitions and basic content control. IE does not support timed hyperlinks or any of the SMIL Layout modules. All layout is done locally using Cascading Style Sheets (CSS) [9]. This choice has both pro's and con's, a full discussion of which could easily fill the remainder of this issue.

SVG with SMIL 2.0 Animation

Where the SMIL Language, Basic and XHTML+SMIL profiles define rules for reusing significant portions of the entire SMIL 2.0 specification, it is also possible to reuse only limited portions of the language. An example of this is the integration of the W3C's SMIL 2.0 Animation Recommendation. SMIL Animation was packaged as a separate W3C Recommendation because it was expected to be released ahead of the full SMIL 2.0 specification. (It was ultimately released after SMIL 2.0 was completed.)

Just as XHTML+SMIL allows SMIL timing to be integrated into non-native SMIL languages (like HTML), SMIL Animation has been used by to integrate SMIL timing and animation into the Structured Vector Graphics (SVG) model.

Using SMIL 2.0: Examples

This section presents four examples illustrating how SMIL 2.0 can be used. We'll look at how different SMIL 2.0 profiles handle similar applications.

Crossing the Bridge

Although the SMIL 2.0 specification was finally approved by the W3C as a Recommendation in mid-August, 2001, the basics of the specification had existed since mid-2000. Based on the interim specification, the first full SMIL 2.0 presentation was released in September, 2000. *Crossing the Bridge*[1] was a collection of test cases that allowed SMIL 2.0's timing, interaction, layout, transitions and animation elements to be evaluated by candidate SMIL 2.0 players.

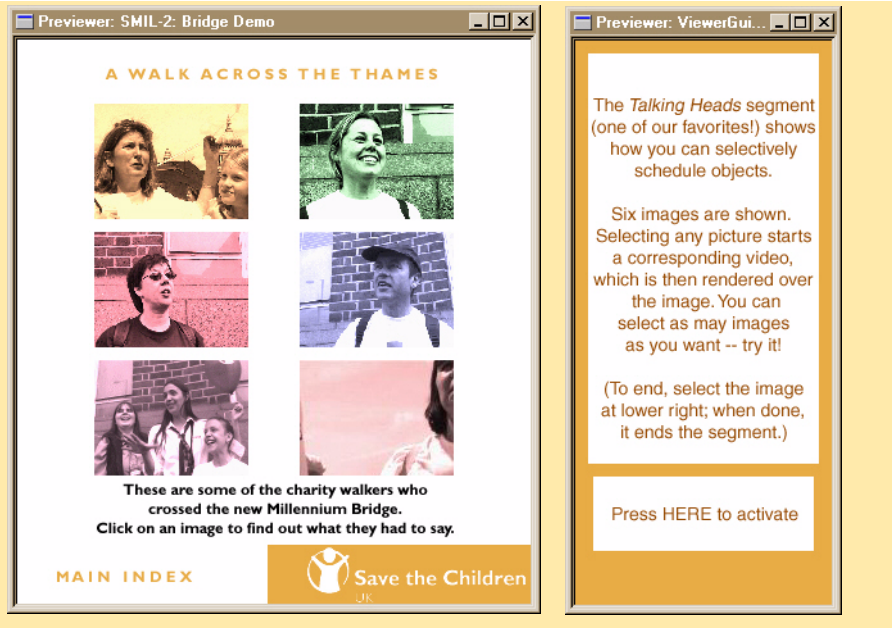


Figure 4: *Multiple top-level windows.*
This figure shows how two top level windows can split navigation, control and content in a presentation. Also, clicking on any of the images starts an associated video. (Presentation ((C) Oratrix, 2000).

Figure 4 shows a view of the presentation as rendered by the Oratrix GR*i*NS player. Two top-level windows are shown: when a user clicks on the *Press HERE to activate* button in the *ViewerGuide* window, the presentation in the main window continues.

Figure 5 illustrates the basic code structure of *Bridge.smil*. The structure of the fragment in Figure 4 is given in detail. The document starts with the definition of the SMIL 2.0 Language namespace, and is followed by the definition of meta-information in the <head> section. The <layout> section uses the *topLayout* container to define regions that are collected into individual top level windows. (Two top level windows are defined.) Some of the regions are defined as a hierarchical collection of sub-regions; this collection is useful when relative layout is used, or when groups of regions are to be animated (moved) in concert. The head section concludes with the definition of two transitions: a quick fade and a fade lasting two seconds.

The body of the presentation (of which only an excerpt is shown), contains an image that is rendered into the *BigLinkPicture region*. An anchor area is defined relative to the size of the

1. The *Bridge* presentation showcased the opening of the Millennium Bridge across the river Thames in London. As part of the ceremonies, several thousand people came to London as part of a drive organized by charitable organizations, one of which — Save the Children, UK — was highlighted in the presentation. A day later, the bridge was closed to the public for safety reasons.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
 <head>
  <meta name="title" content=" SMIL "/>
  <meta name="generator" content="GRiNS Pro for SMIL 2.0, v2.0 b103"/>
  <layout>
   <topLayout id="BridgeDemo" title="SMIL-2: Bridge Demo"
       backgroundColor="white" open="whenActive" close="whenNotActive"
       width="400" height="440">
    <region id="BigLinkPicture" width="400" top="40" height="190"/>
    <region id="PL"left="2" width="140" top="4" height="320" z-index="2">
     <region id="P3" left="10" width="120" top="10" height="90"/>
     <region id="P7" left="10" width="120" top="210" height="90"/>
    </region>
    <region id="BLT" width="400" top="394" height="46" z-index="1"/>
   </topLayout>
   <topLayout id="ViewerGuide" backgroundColor="#e7ac44"
       open="whenActive" width="200" height="440">
    <region id="Help" left="10" width="180" top="10" height="320"/>
    <region id="button" left="14" top="340" height="58"/>
   </topLayout>
  </layout>
  <transition id="q_fade" dur="0.5" type="fade"/>
  <transition id="t_fade" dur="2" type="fade"/>
 </head>
 <body>
  ...
  <img id="MainLinkP2" region="BigLinkPicture" dur="14s" src="L.gif">
   <area href="#TheWalkers" coords="2,125,253,188"/>
  </img>
  ...
 <par id="TheWalkers" endsync="mugShots">
  <img region="Help" src="Help7.GIF" transIn="quick_fade"/>
  <img ID="Press7" region="button" end="mugShots.begin" src="pressB.GIF"/>

  <par id="mugShots" begin="Press7.activateEvent" endsync="WalkersVE">
   <img id="W1i" region="P3" fill="freeze" src="Walker1.gif"/>
      ...
   <img id="W6i" region="P7" fill="freeze" src="Walker6.gif"/>
   <excl id="WalkersVE" endsync="W6v">
    <video id="W1v" region="P3" begin="W1i.activateEvent" src="W1.avi"/>
     ...
    <video id="W6v" region="P7" begin="W6i.activateEvent" src="W6.avi"/>
   </excl>
  </par>
 </par>
     ...
  <img id="LinkBack" region="BLT" src="Bottext2.gif">
   <area show="new" sourcePlaystate="pause" coords="209,0,395,43"
      href="http://www.oneworld.org/scf"/>
   <area href="#MainLinkP2" coords="0,9,195,40"/>
  </img>
 </body>
</smil>
```

Figure 5: *Excerpts Illustrating the Basic Code Structure of Bridge.smil.* ((C) Oratrix, 2000).

region, and a link is defined to *TheWalkers* element. This link will be active for 14 seconds, the duration of the rendering of the *MainLinkP2* image element.

The SMIL *par* (parallel) structure container *TheWalkers* illustrates several important aspects of the SMIL 2.0 specification. First, note that the <par> does not define an explicit duration, but instead declares that it will end when the element with ID *mugShots* terminates. (We don't know when this will be, since the duration is not determined by a presentation timeline — the timeline is determined by the runtime behavior of the presentation.)

When *TheWalkers* gets activated, two images get rendered: the first is the help text shown in the right window in Figure 4, and the second is the "Press" button at the bottom right. Look

carefully at the end conditions of the "Press" button (*Press7*) and the following <par> (*mugShots*). When the user clicks the "Press" button, the *mugShots* <par> starts. As soon as *mugShots* starts, the rendering of *Press7* is terminated. (Figure 4 shows both active.)

Inside *mugShots*, six images get drawn into various layout regions. (Only two are shown.) None of the images have an explicit duration, but all have the fill="freeze" attribute set. This means that they remain on the screen as long as their parent time container (*mugShots*) is active. Along with these images, the <par> also contains a SMIL 2.0 *excl* (exclusive) container. The <excl> has the semantic that at most one of its children is active at any one time. When the <excl> starts, none of the children is active, but the <excl> itself will remain active until the child *W6v* ends. The children of the exclusive get activated when an image outside the <excl> associated with that child is selected. For example, if the viewer selects *W1i* (one of the two images outside the <excl>), the *W1v* element will start. It will be rendered into the same region as the image, covering the image for the time it is active. When *W6i* is selected, the associated video plays. At the end of this video, the *WalkerVE* <excl> ends. This, in turn, ends the *mugShots* <par>, which ends the *TheWalkers* container.

The last part of the document fragment we'll consider is the *LinksBack* image element at the bottom of the page. This element defines two links: one is internal, back to the linking image at the top of the fragment. The other is a link to an external file (in this case, the Save the Children UK home page). When the external link is followed, the *sourcePlaystate* (that is, the current Bridge presentation) is paused and a new window is opened to render the HTML link via the default browser on the user's system.

Happy Birthday

One of the major new features of SMIL 2.0 is support for event-based activation of objects. This provides a mechanism for producing non-linear presentations, where the viewer can select the content to be viewed. An example of this was shown in the Bridge example above.

In order to illustrate how Internet Explorer encodes the same semantics, consider the simple presentation shown in Figure 6. Here, the viewer can select one of the four elements on the left side, at which point a video is shown in the main window at right. (This need not be a video,

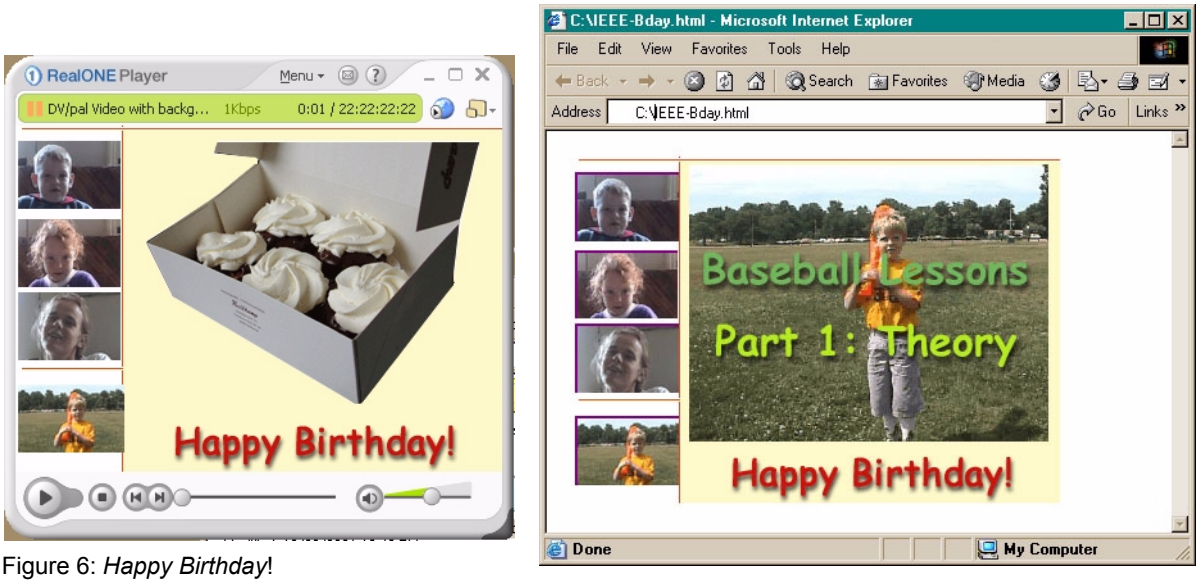


Figure 6: *Happy Birthday*!

The top left illustration shows a presentation in a SMIL 2.0 Language player, while the image at right shows the same presentation (after selecting the bottom-left button) using XHTML+SMIL in Internet Explorer. It is important to note that this is done using SMIL extensions to HTML, not via a plug-in.

of course: it can be a single media object or a structured collection of objects and other SMIL containers.) Figure 6 shows the presentation in both a SMIL Language player (in this case, Real's RealONE player) and in an XHTML+SMIL renderer (IE-6).

Figure 7 shows IE's encoding of an <excl> container to define the selection functionality. Where the SMIL Language profile uses a separate layout format that allows most common layout properties to be referenced by a region name, IE relies on CSS for layout control. Unfortunately, the use of CSS absolute positioning — for all of its charms — does not make the example easier to read (or write!), so most of the style definitions have been removed to conserve space. (The first style definition within the <body> section *is* expanded.)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html xmlns:t = "urn:schemas-microsoft-com:time">
 <head>
  <meta name="generator" content="GRiNS Pro for SMIL 2.0, v2.0 win32 b103"/>
  <style type="text/css">
    .time {behavior: url(#default#time2); }
    t\:*  {behavior: url(#default#time2); }
  </style>
 </head>
 <body>
  <div id="Player_Window" style="position:absolute;left:0;top:0;
     overflow:hidden;width:380;height:270;background-color:black;">
    ...
    <t:par id="MenuImages" dur="indefinite">
     <div id="Menu" style="..." class="time" fill="freeze"
          timeContainer="par">
       <div id="F1s" style="..." class="time">
        <a href="#">
          <img id="F1s_m" src="F1s.jpg" style="..."/>
        </a>
        <t:transitionFilter type="slideWipe" subtype="fromLeft"
          targetElement="F1s" dur="1" begin="Menu.begin" mode="in"
          from="0" to="1"/>
       </div>
     </div>
<!-- This exclusive contains the media items for the menu above -->
     <t:excl id="Videos" dur="indefinite" fill="freeze">
       <t:par id="Fz3" begin="F1s.click" fill="freeze">
        <div id="main" style="..." class="time" timeContainer="par">
         <div id="Fz3_0" style="..." fill="freeze" class="time">
          <t:video src="Fz3.avi" style="..."/>
         </div>
        </div>
       </t:par>
       ...
     </t:excl>
    </t:par>
  ...
 </body>
</html>
```

Figure 7: XHTML+SMIL specification of an exclusive time container.

The IE implementation partitions the document with 's, which correspond in function to region declarations in SMIL layout. The SMIL <par> and <excl> node use is similar to that of the Bridge example. Note that anchors are used to identify the sources for events. Note also the use of transition filters and animation syntax.

The Evening News

The use of multiple profiles, as shown in the Birthday example above, gives one dimension of adaptability of a SMIL 2.0 source document. Within a profile, many opportunities also exist to make content that adapts itself to the needs (and wishes) of the viewer. As an example of what

Figure 8: *An interactive evening news broadcast, Web edition*.
This figure shows two versions of a single SMIL document. In the left figure, a Web version of the newscast is shown: each of the images represent a story that a viewer can select. In the right figure, we see Mobile version of the document: since space is at a premium, the navigation information is partitioned into groups, and the rocker panel is used to go from story to story.
(Content (C) 2001, RTV Slovenia; presentation (C) 2001 Oratrix.)

you can do with SMIL 2.0, consider the evening news example in Figure 8. The left portion of the image shows a 'PC' version of the newscast. The user is presented with a number of picture icons, each representing a segment of the 35 minute news. In this picture, the English language titles and captions are shown (the other option would have been to use the original Slovenian titles). Each of the assets for all the available languages and resolutions are encoded in a single SMIL file, with the SMIL player selecting the appropriate media based on configuration files or user preference dialogues. Note the three icons at the bottom-left of this display. These are items that are encoded in the SMIL file as 'priority objects that can interrupt the main presentation' by defining a *priority class* inside of an *excl* element. When selected, the object will interrupt the running item, and when the interrupting object is completed, the original item will pick up where it left off in the presentation. All of the other icons are encoded in the SMIL file as images; after the images are rendered, an *excl* element is activated containing all of the news stories. Each story is started when its related image is clicked. For each story, there is a high-bandwidth version and a low bandwidth version, the content used being determined by the resolution of the bandwidth switch. (The actual SMIL file is a bit more complex, because it also has a switch that determines whether captions are shown, and if so, which language should be used.)

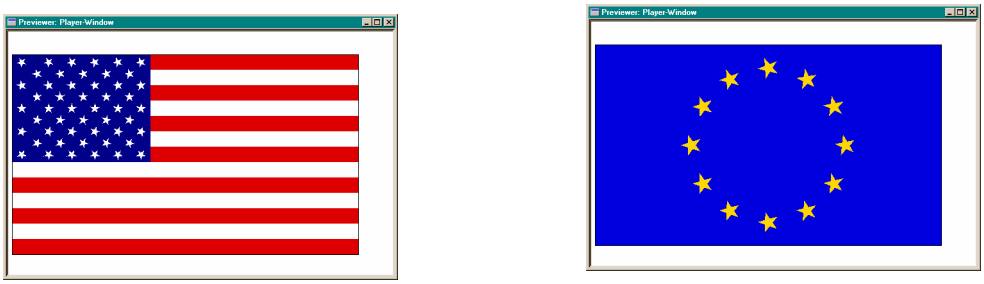
The structure of the mobile version is essentially the same, except that another outer switch determines if a separate image is generated for each story, or if general 'story classes' (such as *National*, *Local*, *Weather*, etc.) are used. Perhaps the most important aspect of this example is that both the PC and the mobile versions are encoded in the same SMIL file. This allows the document to be authored once, and used in multiple configurations.

It is also important to note that since the SMIL file makes indirect references to media objects and that since much of the timing is derived from logical structure rather than timeline-based definitions, a single SMIL file can be used as a generic template — each evening's version is

essentially the same news template, with different media objects being inserted nightly manually or automatically. (For a discussion of workflow issues related to generation of this type of application, see [10].)

Flags

The final SMIL 2.0 example in this article illustrates the use of SMIL primitives in the SVG format. Figure 9 shows two views of a presentation: if the base language selected is American English, the flag at left is shown. If any other language is selected, then the European flag at right is shown. (All of the other possibilities are removed to conserve space.) Although it is difficult to see in this printed version, each of the flags have stars that rotate during the presentation. The rotation is not done in the (very simple) SMIL source code, but by the SVG renderer using SMIL Animation primitives.




```
<?xml version="1.0" encoding="ISO-8859-1"?>
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <root-layout id="Player-Window" width="600" height="380"/>
      <region id="F1" left="0" width="600" top="0" height="380"/>
    </layout>
  </head>
  <body>
    <par id="Simple_Slideshow">
      <switch id="Images">
        <img region="F1" src="usFlag.svg" systemLanguage="en-us"/>
        <img region="F1" src="euFlag.svg" />
      </switch>
    </par>
  </body>
</smil>
```

Figure 9: *Flags*.
This presentation selects one of two flags, based on the language preference setting that was available at run-time. The flags themselves are SVG objects (here rendered in the GRiNS player).

Figure 10 shows the source for the European flag object. The animateTransform element and attributes are defined as part of the SMIL Animation specification. Note that not all SVG renderers support the accelerate/decelerate attributes, so use caution in copying the example.

SMIL and Other Specifications

The potential of multimedia on the Web is tremendous. Even the most conservative commentators realize that the introduction of audio/video assets as first-class Web objects can have a major impact on the way that people will consume information and entertainment in the future. Not surprisingly, SMIL is not alone in the multimedia standardization space.

SMIL has a number of advantages over proprietary formats such as Macromedia's Flash: it is fully XML based, it is able to describe adaptive content, and a single file can be targeted to several different target platforms (such as broadband, Web and mobile.) The format is also

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE svg ... >
<svg viewBox="0 0 1000 600">
  <g transform="translate(10 60)">
    <defs>
      <rect id="f" width="900" height="520" style="fill:#0000dd"/>
      <polygon id="s" style="fill:gold;fill-rule:nonzero;"
        points="0,-28 16.5,22.6 -26.6,-8.6 26.6,-8.6 -16.5,22.6">
        <animateTransform attributeName="transform" type="rotate"
          values="0;360" dur="4s" repeatDur="22s" accelerate="0.5"
          decelerate="0.5" />
      </polygon>
      </defs>
      <use xlink:href="#f"/>
      <rect id="border" width="900" height="520"
         style="fill:none;stroke:black;stroke-width:0.85"/>
      <g transform="translate(450 60)">
        <use xlink:href="#s" />
        <use xlink:href="#s" x="-100"  y="30"/>
          ...
        <use xlink:href="#s" x="100"  y="30"/>
      </g>
  </g>
</svg>
```

Figure 10: *SVG encoding of the European Flag*.

truly open: no one company controls its destiny. (Whether you see this as an advantage depends on your view of the often lengthy standardization process.)

It is often stated that the biggest competitor for SMIL 2.0 is MPEG-4 [11]. While it is tempting to be drawn into a feature-by-feature comparison, most of SMIL 2.0 and MPEG-4 are actually very complementary. During the development of SMIL 2.0, close cooperation was maintained with a portion of the MPEG-4 community. Since MPEG-4 is a binary-encoded standard, it was felt that SMIL 2.0 could be used as the text-format for MPEG-4 presentations. As shown in Figure 11, there is a continuum of concerns among MPEG-4 and SMIL (addressed by the XMT specification [12]), that allows MPEG-4 to be used for low-level object encoding and SMIL 2.0 to be used for high-level, XML-based object composition.

The major differences between MPEG-4 and SMIL lie in the approach to document structuring and in the separation between structure and content. MPEG-4 is essentially a final-form description of a presentation. It is saved in a binary format, and contains extensive content and control information. A SMIL document provides a specification of the high-level

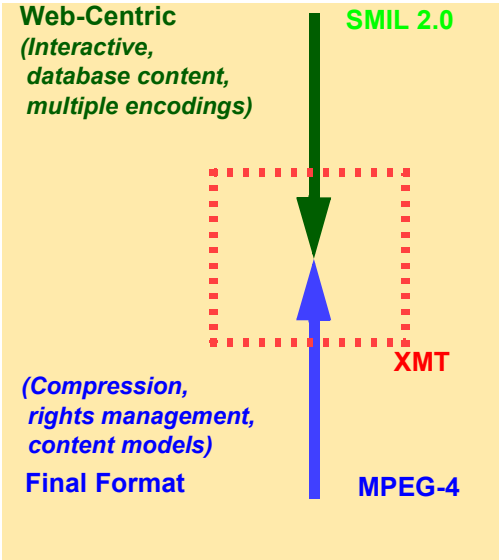


Figure 11: General partitioning of SMIL 2.0 and MPEG-4 functionality. XMT addresses the region of overlap.

and detailed synchronization, layout and content control requirements, but it contains no data. (The actual media items are included by reference.) The combination of content and control can provide a more precise implementation of a presentation, but the separation of content and control can provide a more flexible and reusable presentation architecture. If you own all the content and if your presentation assets are relatively static (that is, they don't change on a daily basis), then the use of a protected final-form approach such as MPEG-4 can be attractive. If, on the other hand, you want to create dynamic presentations or ones where you need to reference content you don't have control over, the SMIL-based approach is probably more attractive. The text-based format of SMIL and its native support for transitions and other high-level content manipulation also provide a lower entry barrier than MPEG-4 based tools.

Both SMIL and MPEG-4 use a profile-based architecture. The profile architecture of MPEG-4 is significantly more complex than SMIL's, resulting in a wide variability in the components that are actually implemented in any particular MPEG-4 player. MPEG-4 also carries with it substantial licensing issues, as much of its component standards contain intellectual property that must be licensed from the IP owners. (SMIL 2.0 has been implemented under a royalty-free agreement with its developers.)

While the technical merits of the various Web multimedia formats can form the basis for interesting, impassioned and lengthy comparison, perhaps the main advantage available to SMIL 2.0 in the short run is the massive deployment of SMIL 2.0 players. The support of SMIL by major player, browser and mobile device vendors will mean that hundreds of millions of SMIL 2.0 players across several SMIL 2.0 profiles will be available by mid-2002.

Closing Comments: Creating SMIL 2.0 Documents

The examples discussed in this article have been selected to illustrate some of the important principles of the SMIL 2.0 language and its implementations. While each of these presentations could have easily been hand-authored using the XML-syntax of SMIL in its various profiles, doing so brings with it a potential for disaster. A streaming media presentation's behavior does not only depend on the syntax and semantics of the source document, but also on the assumptions made about the (end-to-end) executions environment. Syntactically correct SMIL source may still lead to an "incorrect" presentation from the viewer's perspective.

An example of the extra environmental dimension of presentation delivery is shown in figure 12. Here, the bandwidth implications of the same presentation under two conditions (an end-to-end bandwidth availability of 112K vs 56K) are shown using the GR*i*NS SMIL 2.0 Editor [13]. In one case, the presentation will behave as expected, while in the other, a syntactically correct specification will result in substantial delays for the user.

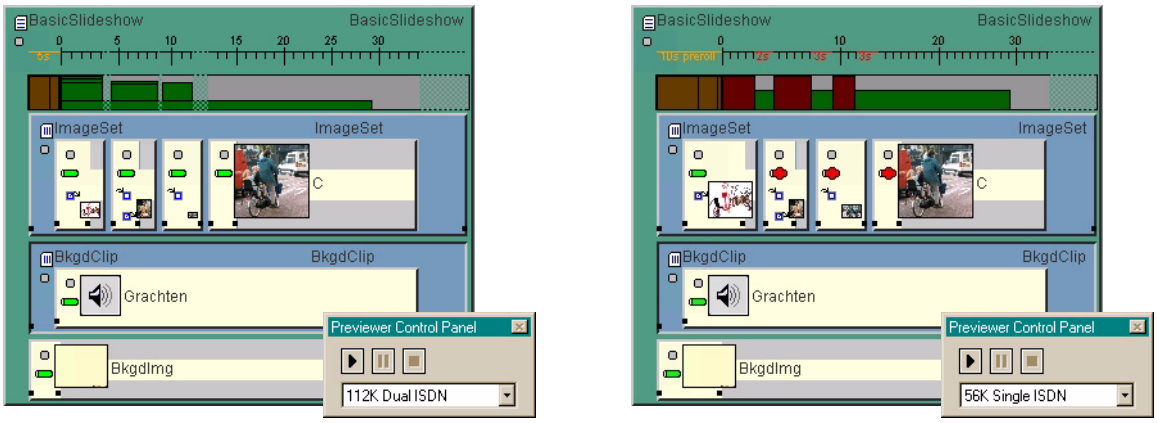


Figure 12: *The impact of the environment on the presentation, as modelled by the GRiNS SMIL 2 Editor.*

The SMIL 2.0 specification provides network-savvy authors (or authors using good tools) extensive abilities to creating compelling presentations that can adapt to the distribution environment. The ability to multi-target content and reuse structure and assets will probably be SMIL 2.0's most lasting contribution.

Acknowledgements

The author would like to that Aaron Cohen, Frank Nack and Lloyd Rutledge for their comments. All of the examples discussed in this article are available in full at

`http://www.oratrix.com/Demos`.

References

1. J. Ayers, D.C.A. Bulterman, A. Cohen, et al, "Synchronized Multimedia Integration Language (SMIL) 2.0", World Wide Web Consortium Recommendation, 7 August 2001. http://www.w3.org/TR/2001/REC-smil20-20010807/
2. DCMI, "Dublin Core Metadata Initiative," http://dublincore.org/ .
3. D.C.A. Bulterman, "SMIL 2.0: Part 1 - Overview, Concepts and Structure," IEEE Multimedia, <plz add info >
4. Oratrix Development BV, "GRiNS/SMIL-2.0 Player", http://www.oratrix.com/ .
5. RealNetworks, Inc., "RealONE Player," http://www.real.com/player/ .
6. 3rd Generation Partnership Project, "Transparent end-to-end packet switched streaming service (PSS), Protocols and CODECs (Release 4)," 3GPP TS 26.234, http://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_11/Docs/PDF/SP-010079.pdf
7. Microsoft, Inc, "Time2 Behavior", http://msdn.microsoft.com/library/default.asp?url=/workshop/author/behaviors/time.asp .
8. J. Bowler, C. Brown, M. Capsimalis, et al, "Scalable Vector Graphics (SVG) 1.0 Recommendation," World Wide Web Consortium Recommendation, 4 September 2001. (J. Farraiolo, Ed.) http://www.w3.org/TR/SVG.
9. B. Bos, H.W. Lie, C. Lilley, I. Jacobs, "Cascading Style Sheets, level 2 (CSS2) Specification," World Wide Web Consortium Recommendation, 12 May 1998. http://www.w3.org/TR/REC-CSS2/ .
10. D.C.A. Bulterman, "Repurposing Broadcast Content for the Web," European Broadcasting Union Technical Review, TR-287, February 2001. http://www.ebu.ch/trev_287-bulterman.pdf.
11. ISO/IEC JTC1/SC29/WG11, "Overview of the MPEG-4 Standard," R. Koenen, Ed., ISO/IEC JTC1/SC29/WG11 Note N4030, March 2001.
12. M. Kim, O. Avaro, M. Bourges-Sevenier, L-T. Cheok, A. Cohen, A. Eleftheriadis, R. Rafey, J. Smith, S. Wood, "eXtensible MPEG-4 Textual Format", Contribution ISO-IEC JTC1/SC29/WG11 MPEG00/6110, May 2000,Geneva, Switzerland (52th MPEG meeting).
13. Oratrix Development BV, "GRiNS/SMIL-2.0 Editor Pro", http://www.oratrix.com/ .

Dick Bulterman is a founder and president of Oratrix Development in Amsterdam and served as one of the editors of the SMIL 2.0 specification. Readers may contact him at: Dick.Bulterman@oratrix.com. *This article is adapted from the up-coming book* SMIL: Interactive Multimedia on the Web, *to be published by* Pearson Education.