

Dick C.A.
Bulterman
Oratrix
Development,
Amsterdam

SMIL 2.0

Part 2: Examples and Comparisons

This article is the second part of a two-part series on SMIL 2.0, the newest version of the World Wide Web Consortium's Synchronized Multimedia Integration Language.¹ Part 1 of this article looked in detail at various aspects of the SMIL specification and the underlying SMIL timing model.² This part looks at simple and complex examples of SMIL 2.0's use and compares SMIL with other multimedia formats. Here, we'll focus on SMIL's textual structure in its various implementation profiles.

SMIL's focus

The W3C's SMIL 2.0 specification, released in August 2001, is a collection of modules that describe how we can integrate and schedule media objects (in their most general form) for presentation in local and network-based environments. SMIL's focus isn't on data encoding—that is, it doesn't specify codecs (decoding engines and formats) 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:

- *Animation.* A collection of modules that we can use to define time-varying values to elements and attributes within the containing document. For example, animation elements can cause a layout region's background color or the position of media objects to change over time.
- *Content control.* A collection of modules that we can use to conditionally include media items (or SMIL structure hierarchies) in a presentation based on various system- and user-defined test attributes. For example, we can use the SMIL switch element to define a set of alternative encodings, one of which will be selected at runtime, based on the value of a system parameter (such as bandwidth, screen size, or language preference).
- *Layout.* A collection of modules that we can use to explicitly manage a presentation's rendering space and audio-visual resources. For example, the **regPoint** and **regAlign** attributes let us center collections of objects around a particular point on the screen.
- *Linking.* A collection of modules that let us define and manage both SMIL temporal and HTML-like nontemporal links. For example, we can define an anchor 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, we can use the **ref** element to reference external objects based on their uniform resource identifier (URI).
- *Metainformation.* A module, compatible with the Dublin Core (<http://dublincore.org/>), for encoding metainformation on an entire presentation or any subpart of that presentation. For example, the metainformation modules specify how we can add author, title, date, copyright, and other such information 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 more than 200 pages in the SMIL 2.0 specification) detailing the SMIL timing model's elements, attributes, and behavior.
- *Time manipulation.* A collection of modules that define how we can manipulate time within a presentation. For example, some profiles let time elapse faster or slower than normal play time.
- *Transitions.* A collection of modules that add visual transitions to a presentation. For example, these modules define how objects can appear using fades, wipes, and other forms of visual transitions.

SMIL 2.0 implementation profiles

The W3C designed SMIL 2.0's modules so that player manufacturers can combine relevant groups of modules into implementation profiles. Presently, three profiles have been implemented and distributed, and a fourth pseudoprofile is also in circulation.

The set of SMIL 2.0 profiles currently implemented include SMIL 2.0 language, SMIL basic, SMIL animation, and XHTML+SMIL. Of these, the SMIL 2.0 language and basic profiles are part of the SMIL 2.0 specification. Several vendors have released SMIL 2.0 language profile players and tools, including Oratrix (<http://www.oratrix.com/>) and RealNetworks (<http://www.real.com/player/>). The Third Generation Partnership Project (3GPP) consortium has used the SMIL basic profile as the basis for the wireless multimedia specification.³

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 Internet Explorer 5.5 and 6.0 browsers.⁴

The fourth implementation profile, SMIL animation, isn't technically a language profile but rather a module integration. The W3C's structured vector graphics (SVG) specification,⁵ which has been integrated by several vendors in their media formats and tools, uses SMIL animation as the basis for specifying timing and Extensible Markup Language (XML) element and attribute animation.

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

```
<?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>
    ...
    
    ...
    <video src="..." transOut="push"/>
    ...
  </body>
</smil>
```

Figure 1. General structure of a SMIL 2.0 language profile document.

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 SMIL 2.0's modules.

Figure 1 shows a presentation's structure in the language profile. Each document begins with an XML encoding string, followed by the SMIL element. The SMIL element can 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"`. Additional namespaces may be declared within the document (for use by particular SMIL 2.0 players).

The **SMIL** element contains an optional **<head>** element and a **<body>** element. The **<head>** element may contain layout and transition declarations as well as metainformation 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 lets us support SMIL on minimal devices, such as telephones and PDAs. A document's structure in the SMIL basic profile resembles that of the language profile. (Among other aspects, they share the same namespace definition.)

The differences between the profiles are in the module functionality set they support and the complexity of the document structure allowed.

Figure 2. Basic Internet Explorer structure as an example of the XHTML+SMIL profile.

```
<!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>
```

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 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 lets us integrate SMIL timing and presentation semantics in other, typically non-temporal XML documents. The SMIL 2.0 language profile provides a general time framework for all elements and attributes in a document, but the XHTML+SMIL profile provides a limited scope time framework for individual XHTML elements or a group of elements.

An XHTML+SMIL profile implementation is available in Microsoft's Internet Explorer. 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.⁴ (Microsoft uses

the marketing name HTML+TIME for their IE implementations.)

Figure 2 shows the basic structure of SMIL timing in IE. Because the document base is HTML, the outer document element isn't `<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 displayed according to SMIL's sequential time container semantics.

IE supports most of SMIL 2.0's timing semantics, plus animation, transitions, and basic content control. IE doesn't support timed hyperlinks or any of the SMIL layout modules. Browsers handle all layout locally using cascading style sheets (CSS).⁶ This choice has both pros and cons, a full discussion of which could easily fill the remainder of this issue.

SVG with SMIL 2.0 animation

The SMIL language, basic, and XHTML+SMIL profiles define rules for reusing significant portions of the entire SMIL 2.0 specification, but we can also reuse limited portions of the language.

An example of this is the integration of the W3C's SMIL 2.0 Animation Recommendation. The W3C packaged SMIL animation as a separate recommendation and expected to release it ahead of the full SMIL 2.0 specification. (It was ultimately released after SMIL 2.0.)

Just as XHTML+SMIL lets us integrate SMIL timing into non-native SMIL languages (like HTML), SMIL animation has been used by Adobe to integrate SMIL timing and animation into the SVG model.

Using SMIL 2.0

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

Crossing the Bridge

Although the W3C approved the SMIL 2.0 specification 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 was a collection of test cases that let candidate SMIL 2.0 players evaluate SMIL 2.0's timing, interaction, layout, transitions, and animation elements.

Figure 3 shows a view of the presentation as rendered by the Oratrix Grins player. The figure shows two top-level windows. When a user clicks on the "Press HERE to activate" button in the ViewerGuide window, the presentation in the main window continues.

Figure 4 (next page) illustrates the basic code structure of Bridge.smil, and highlights the fragment in Figure 3. The document starts by defining the SMIL 2.0 language namespace and then defines 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 subregions; 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 by defining two transitions: a quick fade and a fade lasting two seconds.

The body of the presentation (of which Figure 4 only shows an excerpt) contains an image that is rendered into the `BigLinkPicture` region. An anchor area is defined relative to the region's size, and a link to `TheWalkers` element is defined. This link will be active for 14 seconds, the duration of the `MainLinkP2` image element's rendering.



Figure 3. 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 © 2000 Oratrix).

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

When `TheWalkers` gets activated, the player renders two images: the help text in the right window in Figure 3 and 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 `Press7` rendering terminates. (Figure 4 shows both active.)

Inside `mugShots`, six images get drawn into various layout regions. None of the images have an explicit duration, but all have the `fill="freeze"` attribute set. This means 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 are active, but the `<excl>` itself will remain active until the child `W6v` ends. The children of the exclusive get activated when the user selects an image outside the `<excl>` associated with that child. For example, if the viewer selects `W1i` (one of the two images out-

Figure 4. Excerpts illustrating the basic code structure of Bridge.smil.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <meta name="generator" content="GRiNS Pro for SMIL 2.0, v2.0 b103"/>
    <layout>
      <topLayout id="BridgeDemo" 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" 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>
    ...
    
      <area href="#TheWalkers" coords="2,125,253,188"/>
    </img>
    ...
    <par id="TheWalkers" endsync="mugShots">
      
      
      <par id="mugShots" begin="Press7.activateEvent" endsync="WalkersVE">
        
        ...
        
        <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>
    ...
    
      <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>
```

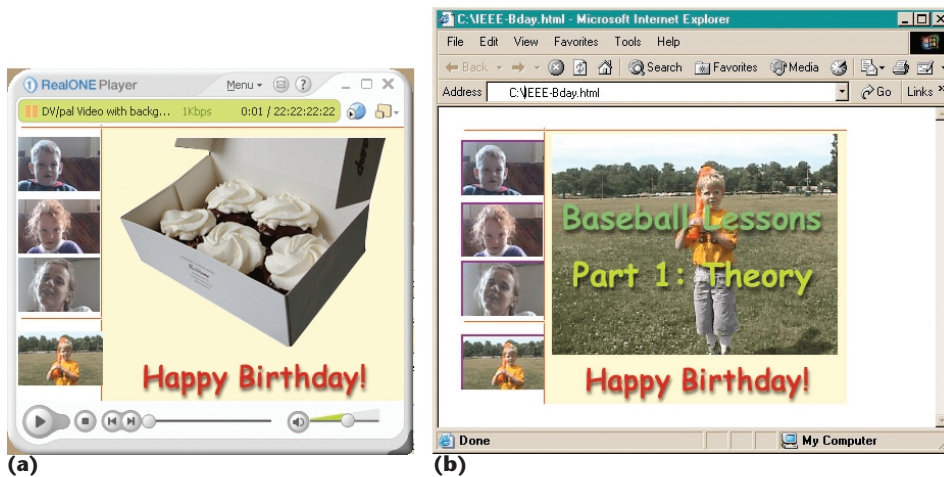


Figure 5. (a) A presentation in a SMIL 2.0 language player. (b) The same presentation—launched when the user selects the bottom-left button in (a)—using XHTML+SMIL in Internet Explorer. We do this using SMIL extensions to HTML, not via a plug-in.

side the `<exc1>`), the `W1v` element will start. It will be rendered into the same region as the image, covering the image for the time it's active. When the user selects `W6i`, the associated video plays. At the end of this video, the `WalkerVE <exc1>` 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. This element defines two links. One is internal, linking 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 homepage). When the viewer follows the external link, the `sourcePlaystate` (that is, the current Bridge presentation) pauses and a new window opens 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 object activation. This provides a mechanism for producing nonlinear presentations, where the viewer can select the content to be viewed. The Bridge example shows this.

The simple presentation in Figure 5 illustrates how IE encodes the same semantics. Here, the viewer can select one of the four elements on the left side, at which point a video appears in the main window at right. (This need not be a video, of course. It can be a single media object or a structured collection of objects and other SMIL containers.) Figure 5 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 6 (page 80) shows IE's encoding of an `<exc1>` container to define the selection functionality. Where the SMIL language profile uses

a separate layout format that lets us reference the most common layout properties with a region name, IE relies on CSS for layout control. Unfortunately, using CSS absolute positioning—for all its charms—doesn't make the example easier to read (or write), so we have removed most of the style definitions to conserve space. (The first style definition within the `<body>` section is expanded.)

The IE implementation partitions the document with `<div>`'s, which correspond in function to region declarations in SMIL layout. The SMIL `<par>` and `<exc1>` node use resembles that of the Bridge example. Note that anchors identify the sources for events. Also, note the use of transition filters and animation syntax.

Evening news

Using multiple profiles, as the Birthday example shows, 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 viewer's needs (and wishes). As an example of what you can do with SMIL 2.0, consider the evening news example in Figure 7 (page 81). 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. This picture shows English language titles and captions (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 encoded in the SMIL file as priority objects that can inter-

```

<!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="#">
            
          </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 6. XHTML+SMIL specification of an exclusive time container.

rupt 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 completes, the original item will pick up where it left off in the presentation. All the other icons are encoded in the SMIL file as images; after the images are rendered, an **excl** element containing all the news stories activates.

Each story starts when a user clicks its related

image. In addition, each story has a high- and low-bandwidth version—the resolution of the bandwidth switch determines the content used. (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



Figure 7. An interactive evening news broadcast, Web edition, with two versions of a single SMIL document. (a) In the Web version of the newscast, each of the images represent a story that a viewer can select. (b) In the mobile version, because space is at a premium, we partitioned the navigation information into groups, and viewers use the rocker panel to go from story to story. (Content © 2001 RTV Slovenia; presentation © 2001 Oratrix.)

determines if a separate image is generated for each story or if general story classes (such as national, local, or weather) are used. The most important aspect of this example is that we encoded both the PC and mobile versions in the same SMIL file. This lets us author the document once and use it in multiple configurations.

It's also important to note that because the SMIL file makes indirect references to media objects, and because much of the timing is derived from logical structure rather than timeline-based definitions, we can use a single SMIL file as a generic template. Each evening's version is essentially the same news template, with different media objects inserted nightly either manually or automatically. (See the European Broadcasting Union technical review⁷ for a discus-

sion of workflow issues related to generating this type of application.)

Flags

The final SMIL 2.0 example illustrates using SMIL primitives in the SVG format. Figure 8 shows two views of a presentation. If viewers select the base language as American English, they see Figure 8b. If viewers select any other language, they see the European flag (Figure 8c). Although it's difficult to see in this printed version, each flag has stars that rotate during the presentation. The rotation isn't done in the SMIL source code but by the SVG renderer using SMIL animation primitives.

Figure 9 (next page) shows the source for the European flag object. The animateTransform ele-

```
<?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">
        
        
      </switch>
    </par>
  </body>
</smil>
(a)
```

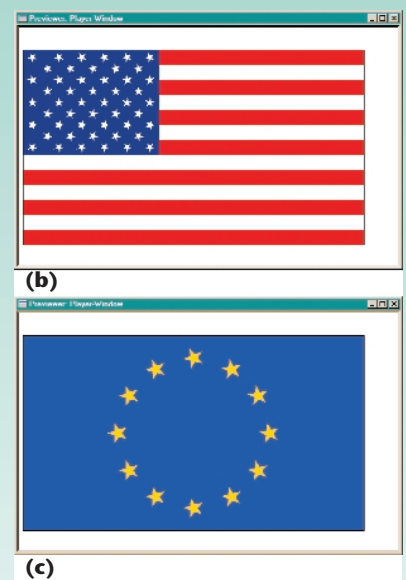


Figure 8. (a) This presentation selects one of two flags, based on the language preference setting available at runtime. The flags are SVG objects (here rendered in the Grins player). Depending on which base language the viewer selects, they see either the (b) US or (c) European flags.

```

<?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 9. SVG encoding of the European Flag.

ment and attributes are defined as part of the SMIL animation specification. Note that not all SVG renderers support accelerate and decelerate attributes, so use caution in copying this 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 people will consume information and entertainment in the future. Not surprisingly, SMIL isn't alone in the multimedia standardization space.

SMIL has a number of advantages over proprietary formats such as Macromedia's Flash: it's completely XML based; it can describe adaptive content; and with a single file, we can target several different target platforms (such as broadband, Web, and mobile). The format is also truly open because no one company controls its destiny. (Whether you see this as an advantage depends on your view of the often lengthy standardization process.)

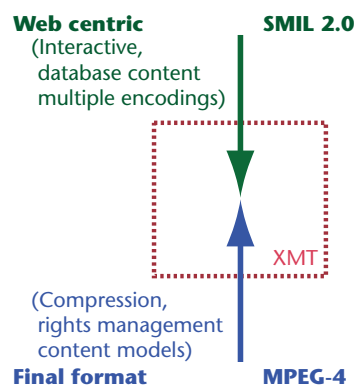
Many people believe that the biggest competitor for SMIL 2.0 is MPEG-4.⁸ While it's tempting to be drawn into a feature-by-feature comparison, most of SMIL 2.0 and MPEG-4 are actually complementary. During SMIL 2.0's development, the W3C maintained close cooperation with a portion of the MPEG-4 community. Because

MPEG-4 is a binary-encoded standard, the MPEG-4 community felt that SMIL 2.0 could be used as the text format for MPEG-4 presentations. There's a continuum of concerns among MPEG-4 and SMIL (addressed by the Extensible MPEG-4 Textual Format [XMT] specification⁹) 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 (see Figure 10).

The major differences between MPEG-4 and SMIL lie in their approach to document structuring and in the separation of structure and content. MPEG-4 is essentially a final-form presentation description. It's saved in a binary format and contains extensive content and control information. A SMIL document provides a specification of the high-level and detailed synchronization, layout, and content control requirements, but it contains no data. (The SMIL document references the actual media items.) 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

Figure 10. General partitioning of SMIL 2.0 and MPEG-4 functionality. XMT addresses the overlap region.



don't change on a daily basis—then using a protected final-form approach such as MPEG-4 can be attractive. On the other hand, if 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. SMIL's text-based format and 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. MPEG-4's profile architecture is significantly more complex than SMIL's, resulting in a wide variability in the components actually implemented in any particular MPEG-4 player. MPEG-4 also carries with it substantial licensing issues, because 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.)

Although the technical merits of the various Web multimedia formats can form the basis for interesting, impassioned, and lengthy comparison, perhaps the main advantage of 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

I selected the examples in this article 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 not only depends on the source document's syntax and semantics but also on the assumptions made about the (end-to-end) execution environment. Syntactically correct

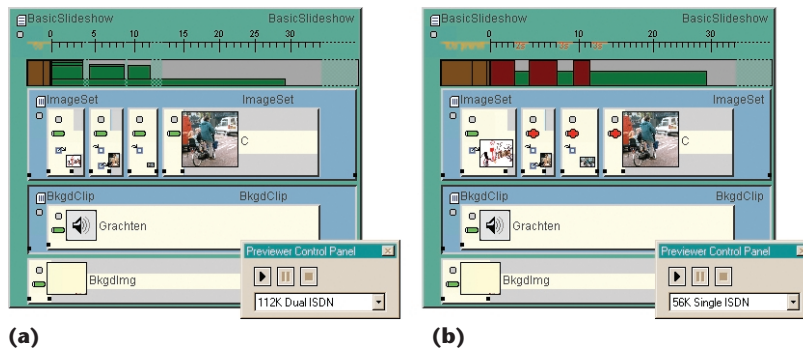


Figure 11. *The impact of the environment on the presentation (modeled by the Grins SMIL 2.0 Editor) (a) In one case, the presentation will behave as expected. (b) In the other, a syntactically correct specification will result in substantial delays for the user.*

SMIL source may still lead to an “incorrect” presentation from the viewer's perspective.

Figure 11 shows an example of the extra environmental dimension of presentation delivery. The figure shows the bandwidth implications of the same presentation under two conditions (an end-to-end bandwidth availability of 112 Kbytes versus 56 Kbytes) using the Grins SMIL 2.0 Editor. In one case, the presentation will behave as expected, but in the other, a syntactically correct specification will result in substantial delays for the user.

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

MM

Acknowledgments

I thank Aaron Cohen, Frank Nack, and Lloyd Rutledge for their comments. All the examples in this article are available in full at <http://www.oratrix.com/Demos>.

This article is adapted from the upcoming book *SMIL: Interactive Multimedia on the Web* to be published by Pearson Education.

References

1. J. Ayers et al., *Synchronized Multimedia Integration Language (SMIL) 2.0*, World Wide Web Consortium Recommendation, Aug. 2001, <http://www.w3.org/TR/2001/REC-smil20-20010807/>.
2. D.C.A. Bulterman, “SMIL 2.0 Part 1: Overview, Concepts, and Structure,” *IEEE MultiMedia*, vol. 8, no. 4, Oct.–Dec. 2001, pp. 82–88.

3. *Transparent End-To-End Packet Switched Streaming Service (PSS)*, Protocols and CODECs (Release 4), 3GPP TS 26.234, Third Generation Partnership Project, 2001, http://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_11/Docs/PDF/SP-010079.pdf.
4. *Time2 Behavior*, Microsoft, Readmond, Wash., 2002, <http://msdn.microsoft.com/library/default.asp?url=/workshop/author/behaviors/time.asp>.
5. J. Bowler et al., *Scalable Vector Graphics (SVG) 1.0 Recommendation*, World Wide Web Consortium Recommendation, Sept. 2001, <http://www.w3.org/TR/SVG>.
6. B. Bos et al., *Cascading Style Sheets, Level 2 (CSS2) Specification*, World Wide Web Consortium Recommendation, May 1998, <http://www.w3.org/TR/REC-CSS2/>.
7. D.C.A. Bulterman, *Repurposing Broadcast Content for the Web*, European Broadcasting Union Technical Review TR-287, EBU, Geneva, Feb. 2001, http://www.ebu.ch/trev_287-bulterman.pdf.
8. ISO/IEC JTC1/SC29/WG11, Note N4030, *Overview of the MPEG-4 Standard*, R. Koenen, ed., Mar. 2001.
9. M. Kim et al., "eXtensible MPEG-4 Textual Format," Contribution ISO-IEC JTC1/SC29/WG11 MPEG00/6110, May 2000, Geneva, Switzerland.

Readers may contact Dick Bulterman at Oratrix Development, Amsterdam, The Netherlands, email Dick.Bulterman@oratrix.com.

Contact Standards editor Peiya Liu, Siemens Corporate Research, 755 College Rd. East, Princeton, NJ 08540, email pliu@scr.siemens.com.

2002 Editorial Calendar

January/February: Information Visualization

Computer-based information visualization has emerged as a distinct field centered around helping people explore or explain data by designing software that exploits the properties of the human visual system. New methodologies and techniques are critical for helping people keep pace with the torrents of data.

March/April: Image-Based Modeling, Rendering, and Lighting

The field of image-based modeling and rendering has already established itself as an important tool for a wide range of computer graphics applications. Image-based techniques use real-world digital photographs to synthesize novel imagery, letting us creatively explore and reinterpret realistic geometry, surface properties, and illumination.

May/June: Graphics in Advanced Computer-Aided Design

Using computers in the design and manufacturing processes has come a long way from the first CAD systems in the automobile and aerospace industries, with the huge mainframes and enormously expensive displays. Current CAD systems exploit innovative uses of the technologies that help to move ideas from concept to model to prototype to product.

July/August: Virtual Worlds, Real Sounds

We only need to close our eyes for a moment to experience the amazing variety of information that our ears provide, often more quickly and richly than any other sense. Using real sounds in virtual worlds involves parametric computation; synthesis; and rendering sound for VR, entertainment, and user interfaces.

September/October: Computer Graphics, Art History, and Archaeology

Archaeologists can use computer graphics techniques to reconstruct and visualize archaeological data of a site that might otherwise be difficult to appreciate, with applications in analysis, teaching, and preservation. Similarly, art historians use computer graphics to analyze, study, and preserve great works of art.

November/December: Tracking

High-resolution tracking of user position and orientation (head, hand, feet, and so on) is increasingly a critical issue for virtual reality, augmented reality, modeling and simulation, and animation. Current tracking hardware is based on a variety of sensors including magnetic, optical, inertial, acoustic, and mechanical.

IEEE
Computer Graphics
AND APPLICATIONS