SMIL 2.0 – XML for Web Multimedia

- SMIL stands for Synchronized Multimedia Integration Language
- SMIL is pronounced "smile"
- SMIL is a language for describing audiovisual presentations
- Enables us to write interactive MM presentation
- SMIL is an XML tagging language- similar to HTML.
- SMIL presentations can be written using a text-editor
What Can SMIL Do?

- Although plug-ins and media players have the ability to show many different types of media with varying support for interaction, only SMIL offers the ability to define the presentation in the form of text as a script. This feature could be called media composition.
- Also, SMIL offers accessibility options and powerful features not present in other media players.
- Macro products like Flash
- Real audio's Realplayer
- Microsoft's powerpoint
- Apple's quicktime

Given that SMIL is extensible, the SMIL language has the ability to show many of proprietary objects which are used by the above players.
Implementing SMIL

Common SMIL implementations

* Internet or Intranet presentations.
* Slide show presentations.
* Presentations which link to other SMIL files.
* Presentations which have Control buttons (stop, start, next, ...)
* Defining sequences and duration of multimedia elements.
* Defining position and visibility of multimedia elements.
* Displaying multiple media types such as audio, video, text
* Displaying multiple files at the same time.
* Displaying files from multiple web servers.

Currently, SMIL's most widespread usage is with MMS. MMS (Multimedia Messaging System) is a mobile device technology
SMIL files need to be written according to the following rules:

* SMIL documents must follow the XML rules of well-formedness.
* SMIL tags are case sensitive.
* All SMIL tags are written with lowercase letters.
* SMIL documents must start with a `<smil>` tag and end with a `</smil>` closing tag.
* SMIL documents must contain a `<body>` tag for storing the contents of the presentation.
* SMIL documents can have a `<head>` element (like HTML) for storing metadata information about the document itself, as well as presentation layout information.
Layout
Within the head tags you must define layout tags

<layout>
</layout>

Within these layout tags you must define a root a root layout, this is where you define the size of the presentation and other attributes (such as the background colour).

<root-layout height="500" width="500" background-color="black" id="main" />

Parallel and Sequence
There are two main tags used when defining the order of the displaying media

<seq>
Simple SMIL file

<smil>
<head>
<layout>
...
</layout>
</head>
<body>
...
</body>
</smil>
A `<seq>` element defines a sequence to display. The `repeatCount` attribute defines an **indefinite loop**. Each `<img>` element has a `src` attribute to define the image source and a `dur` attribute to define the duration of the display.
**Regions**
Still within the layout tags you need to define regions within your presentation. A region is a rectangular/square shaped object which media appears in. To define a region with the layout use the following.

```
<region id="someID" width="200" height="200" top="10" left="10" background-color="white" z-index="1" />
```

The **id** is the **unique identifier** you give this region, the **height and width** define the size of the region, the top and left attributes define how far off the top and left margins of the presentation area the region lies, the background-color defines the **background colour** and the **z-index** attribute defines the order of the stacking of the regions.
Layout

We will make the difference between:

Absolute Positioning,

Relative Positioning
Absolute Positioning

Positioning in upper left corner

<smil>
<head>
  <layout>
    <root-layout width="300" height="200" background-color="Red" />
    <region id="IISc_logo" left="75" top="50" width="32" height="32" />
  </layout>
</head> [...]

Image to our presentation

<smil>
<head>
  <layout>
    <root-layout width="300" height="300"
      background-color="white"/>
    <region id="cows" top="0" left="0"
      width="300" height="197"/>
  </layout>
</head>
<body>
  <img src="cows.jpg" region="cows"/>
</body>
</smil>
Copy and paste this URL, SMIL parser, i.e., RealPlayer, Quickstep, for an example of the SMIL presentation created
To insert a media within our presentation we must specify the region (the place) where it will be displayed. Let's say we want to insert the cow icon (32x32 pixels) at 75 pixels from the left border and at 50 pixels from the top border.
To insert IISc logo in the region called “IISc_logo”, use `<img>` tag

Attribute is region which is pointer to `<region>`

```xml
<smil>
  <head>
    <layout>
      <root-layout width="300" height="200" background-color="white" />
      <region id="IISc_logo" left="75" top="50" width="32" height="32" />
    </layout>
  </head>
  <body>
    <img src="IISc_logo32x32.gif" alt="IISc_logo" region="IISc logo" />
  </body>
</smil>
```
To insert the cows icon in the region called "cows_icon", we use the <img> tag. Region attribute is a pointer to the <region> tag.

```smil
<smil>
  <head>
    <layout>
      <root-layout width="300" height="200" background-color="white" />
      <region id="cows_icon" left="75" top="50" width="32" height="32" />
    </layout>
  </head>
  <body>
    <img src="cows.jpg" alt="The cows_icon" region="cows_icon" />
  </body>
</smil>
```
Relative Positioning

You can also specify media positions relatively to the window's dimensions. For example, if you wish to display the *vim* icon at 50% from the left border and at 40% from the top border, modify the previous source and replace the left and top attributes.
<smil>
  <head>
    <layout>
      <root-layout width="300" height="200" background-color="white"/>
      <region id="vim_icon" left="50%" top="40%" width="32" height="32"/>
    </layout>
  </head>
  <body>
    <img src="vim32x32.gif" alt="The vim icon" region="vim_icon"/>
  </body>
</smil>
Multiple Region

We placed the new region, "text", 200 pixels from the top and 50 pixels from the left so that it's centered in our presentation.

TUCOWS
When two regions overlay

To be sure that one region is over the other, add z-index attribute to `<region>`.

When two region overlay, the one with the greater z-index is on top. If both regions have the same z-index, the first rendered one is below the other.

In the following code, we add z-index to region_1 and region_2.
<smil>
<head>
<layout>
=root-layout width="300" height="200" background-color="white" /> <region id="region_1" left="50" top="50" width="150" height="125" z-index="2"/> <region id="region_2" left="25" top="25" width="100" height="100" z-index="1"/>
</layout>
</head>
<body>
<par>
<text src="text1.txt" region="region_1" />
<text src="text2.txt" region="region_2" />
</par>
</body>
</smil>
Media types
SMIL can integrate many types of media (though this is sometimes limited by the application). Like HTML you need to define a source for your media. You also need to define a region of the presentation in which the play the media.

<text src="someText.txt" region="someRegion" />

We can also define the amount of time that the media plays for by using the `dur' attribute, we can set how far into the media we want it to start playing (only useful with sound and video), how much of the region we want the media to fill and give the media an id.

<video src="someVid.mpg" region="someRegion" begin="10s" dur="50s" repeat="2" fill="fit" id="someVid" />
The Parallel Element

A parallel group of clips can be stopped at the same time using the endsync attribute in the <par> tag.

endsync="first" stops all the clips in the <par> group when the shortest clip are finished regardless of any time parameters set for the other clips.

endsync="last" concludes the <par> group when all clips have finished playing. This is the default.

endsync="id(ID)" concludes the <par> group when the clip with the identified (ID) clip are finished. The ID is referring to the value of the clips id attribute.

Synchronization

---

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>begin</td>
<td>time</td>
<td>Sets the delay before the element is displayed</td>
</tr>
<tr>
<td>dur</td>
<td>time</td>
<td>Sets the duration for the display</td>
</tr>
<tr>
<td>endsync</td>
<td>&quot;first&quot;</td>
<td>Synchronizes the stopping of elements</td>
</tr>
<tr>
<td></td>
<td>&quot;last&quot;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;id(ID)&quot;</td>
<td></td>
</tr>
<tr>
<td>repeatCount</td>
<td>number</td>
<td>Sets the number of repetitions for the display</td>
</tr>
</tbody>
</table>
The following media elements can be used to include media objects in a SMIL document:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;animation&gt;</td>
<td>Defines an animation</td>
</tr>
<tr>
<td>&lt;audio&gt;</td>
<td>Defines an audio clip</td>
</tr>
<tr>
<td>&lt;brush&gt;</td>
<td>Defines a brush</td>
</tr>
<tr>
<td>&lt;img&gt;</td>
<td>Defines an image</td>
</tr>
<tr>
<td>&lt;param&gt;</td>
<td>Defines a parameter</td>
</tr>
<tr>
<td>&lt;ref&gt;</td>
<td>Defines a generic media reference</td>
</tr>
<tr>
<td>&lt;text&gt;</td>
<td>Defines a text</td>
</tr>
<tr>
<td>&lt;textstream&gt;</td>
<td>Defines a texstream</td>
</tr>
<tr>
<td>&lt;video&gt;</td>
<td>Defines a video</td>
</tr>
</tbody>
</table>
The type attribute is used to define the media type.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>erase</td>
<td>Defines the behavior of the element after any timing is complete</td>
</tr>
<tr>
<td>src</td>
<td>Defines the source of a media object</td>
</tr>
<tr>
<td>type</td>
<td>Defines the media type</td>
</tr>
</tbody>
</table>
Duration

The duration (dur="5s") attribute of an element defines how long the element will be visible:

```html
<html>
<head>
<style>.t {behavior: url(#default#time2)}</style>
</head>
<body>
<img class="t" src="image1.jpg" dur="5s" />
</body> </html>
```
3 sec
dur=4s
repeatCount=2
fill=freeze
Time Control

You can also control how long you want a certain piece of media to be displayed for by using the "dur" attribute. This attribute takes a numerical value in seconds with the letter "s" after the number. For example, let's say we want the cows to be displayed for 10 seconds and the rest of the images for only 5. We would add "dur="10s"" to the cow's img tag and "dur="5s"" to the rest of the img tags. Below is what our code would look like.

```xml
<smil>
  <head>
    <layout>
      <root-layout width="300" height="300"
        background-color="white"/>
      <region id="icon" top="10" left="270" z-index="2"
        width="20" height="20"/>
      <region id="cows" top="0" left="0" z-index="1"
        width="300" height="197"/>
      <region id="text" top="200" left="50" z-index="1"
        width="200" height="50" fit="fill"/>
    </layout>
  </head>
  <body>
    <par>
      <img src="hicon.gif" region="icon" dur="5s"/>
      <img src="cows.jpg" region="cows" dur="10s"/>
      <img src="tucows.gif" region="text" dur="5s"/>
    </par>
  </body>
</smil>
```
This should display the TUCOWS text image, then the HTML text image during a span of 10 seconds.
The `<animation>` element defines a reference to an animation object stored as vector graphics or in another animated format.

The `<audio>` element defines a reference to an audio object stored as recorded audio.

```html
<html xmlns:t="urn:schemas-microsoft-com:time">
<head>
  <import namespace="t" implementation="#default#time2" />
</head>
<body>
  <t:audio src="liar.wav" repeatCount="indefinite" type="wav" />
</body>
</html>
```
The <video> Element

The <video> element defines a reference to a video object stored as recorded video.

<video src="http://www.ananova.com/about/vap_windows_check.wmv" repeatCount="indefinite" type="wmv" />
# Fill types

The attribute fill defines how much of the region a particular type of media will fill.

<table>
<thead>
<tr>
<th>Fill Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remove</td>
<td>Specifies that the element will not extend past the end of the last instance of the simple duration.</td>
</tr>
<tr>
<td>Freeze</td>
<td>Specifies that the element will extend past the end of the last instance of the simple duration by &quot;freezing&quot; the element state at that point. The parent time container of the element determines how long the element is frozen (as described immediately below).</td>
</tr>
<tr>
<td>Hold</td>
<td>Setting this to &quot;hold&quot; has the same effect as setting to &quot;freeze&quot;, except that the element is always frozen to extend to the end of the simple duration of the parent time container of the element (independent of the type of time container). For profiles that support a layered layout model (e.g., SMIL 2.0 Language Profile), held elements (elements with fill=&quot;hold&quot;) will refresh their display area when a layer is added on top then later removed.</td>
</tr>
<tr>
<td>Transition</td>
<td>Setting this to &quot;transition&quot; has the same effect as setting to &quot;freeze&quot;, except that the element is removed at the end of the lane.</td>
</tr>
</tbody>
</table>
We want to make the tucows.gif image fit perfectly within the region, so we would add fit="fill" to the text region. Below is an example.

```xml
<smil>
<head>
  <layout>
    <root-layout width="300" height="300" background-color="white"/>
    <region id="cows" top="0" left="0" width="300" height="197"/>
    <region id="text" top="200" left="50" width="200" height="50" fit="fill"/>
  </layout>
</head>
<body>
  <par>
    <img src="cows.jpg" region="cows"/>
    <img src="tucows.gif" region="text"/>
  </par>
</body>
</smil>
```
SMIL for phones

MMS uses SMIL to define the layout of multimedia content. SMIL was adopted because it was a well-defined, standard language to describe the layout and timing of the content inside MMS messages.
How to Play a SMIL File?

To view a SMIL presentation, you will need a SMIL player installed on your computer.

Apple's Quicktime player, Windows Media Player, and RealNetworks RealPlayer support SMIL.

It would be convenient to show SMIL files natively in web browser, eliminating the requirement of a separate SMIL player or plug-in