

# Introduction to SMIL

Martin Halvey  
January 2008



## Overview

- Introduction to W3C and XML
- Introduction to SMIL
- Writing a SMIL file
- Adding Media Objects
- Clipping media files
- Timing and Synchronising
- Layout

## World Wide Web Consortium W3C

- Develop common protocols that promote WWW's evolution and ensure its interoperability
  - repository of information about the WWW for developers and users
  - sample code implementation to embody and promote standards
  - various prototype and sample applications to demonstrate use of new technology

## World Wide Web Consortium W3C

- User Interface Domain
  - Hypertext Markup Language (HTML)
  - Cascading Style Sheets (CSS)
  - Document Object Model (DOM)
- Technology and Society Domain
  - Resource Description Framework (RDF)
- Architecture Domain
  - Hypertext Transfer Protocol (HTTP)
  - Extensible Markup Language (XML)

## Extensible Markup Language XML

- XML is a method for putting structured data in a text file
- XML is text, not binary, but is not meant to be read
- XML is license-free, platform-independent and well-supported, W3C standard since February 1998

## XML Syntax

- A document is made up of elements
- Each element has either a start-tag and end-tag or an empty-element tag
- Tags are identifiers enclosed within less-than (<) and greater-than (>) characters
- Elements are ended with a forward-slash (/) character, either at the beginning of an end-tag or at the end of an empty-element tag

## XML Syntax

- Elements may have one or more attributes. Each attribute is a name/value pair in the form name="value"
- A set of start and end tags surround the text (or other elements) which they're marking up
- Elements may contain other elements (this is called nesting)

## XML vs HTML

- XML tags are case-sensitive. This means that the tags <smil> and <SMIL> are not the same
- Empty-element tags require a forward-slash (/) at the end of the tag (before the greater-than (>) character)
- Elements with a start-tag must have a corresponding end-tag

## What is SMIL

- Set of XML modules
- They are used to describe
  - Temporal behaviour
  - Positional behaviour
  - Interactive behaviour
    - Animation
    - Events

## Why use SMIL

- It is free and open
- Easy to learn and use
- Provides capabilities beyond those of any multimedia format yet seen on the web
- No need for additional software
- W3C adopted and published the SMIL 2.0 recommendation August 2001
- Used in MMS and HD DVD

## Who uses SMIL

- RealNetworks
- Intel
- Macromedia
- IBM
- Intel
- Macromedia
- Microsoft
- Netscape/AOL
- Nokia
- Ericsson
- Canon
- Panasonic
- Phillips
- Many Others

## How to create a SMIL file

- Text Editor is sufficient
- Many companies are providing editors
  - Adobe
  - RealNetworks
  - DoCoMo
  - SMIL Media

## SMIL Player Software

- Players available for a variety of devices
- Players available from
  - Ambulant
  - QuickTime Player
  - RealPlayer
  - Windows Media Player
  - KMPlayer
  - Helix Player

## Writing a SMIL File

Every SMIL should start with a line identifying it as an XML document

```
<?xml version="1.0"?>
```

## Writing a SMIL File

The next line in your SMIL file should identify it to be a SMIL document

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL  
2.0//EN"  
http://www.w3.org/2001/SMIL20/SMIL20.dtd>
```

This is called the XML prolog and/or Document Type Declaration

## Writing a SMIL File

The next line in your SMIL file you should place the SMIL element

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

It requires at least one attribute, the xmlns (XML Namespace)

## Writing a SMIL File

It is a good idea to provide some additional information in the root of your SMIL file

```
<smil
xmlns="http://www.w3.org/2001/SMIL20/Language"
xml:lang="en"
title="SMIL template">
</smil>
```

## Writing a SMIL File

Contains <head> and <body> elements

```
<?xml version="1.0"?>
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0/EN"
"http://www.w3.org/2001/SMIL20/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language"
xml:lang="en" title="SMIL template">
  <head>
  </head>
  <body>
  </body>
</smil>
```

## Adding Media Objects

There are a number of types supported by SMIL

- img
- video
- audio
- animation
- text
- textstream

## Adding Media Objects

Additional types

- ref
- brush

Two types of type

- Discrete
- Continuous



## Adding Media Objects

Each media object has a src attribute where you give the location of a media object

```

```

```
<video src="movie.mpg"/>
```

## Making Media Accessible

Number of attributes exist to add additional information

- alt
- longdesc
- readIndex

## Making Media Accessible

```

```

```
<video src="game.mpg" alt="a movie of one of the games from my birthday party" longdesc="game.txt"/>
```

## Clipping Files

- Easy to play part of a file in SMIL
- Clipping can only be applied to continuous media such as audio, video, and text streams
- Only need to specify when the clip should begin and end
- Give time values in NPT or SMPTE

## Clipping Files

Easy to do in SMIL, use one of two attributes

- clipBegin
- clipEnd

Give values in one of two formats

- Normal Play Time (NPT)
- SMPTE, from Society of Motion Picture and Television Engineers

## Clipping Files

For the hypothetical movie.mpg file, which is 30 minutes in length

Starting 20 seconds into the video:

```
<video src="movie.mpg" clipBegin="20s"/>
```

Cutting out the last 3-minute, 30-second scene:

```
<video src="movie.mpg" clipEnd="26:30"/>
```

## Clipping Files

For the hypothetical movie.mpg file, which is 30 minutes in length

Showing an interesting frame from the middle:

```
<video src="movie.mpg" clipBegin="14:55.7" clipEnd="14:55.7"/>
```

## Time Formats

- "16"
  - 16 seconds (the same as "16s")
- "02:45:14.273"
  - 2 hours, 45 minutes, 14 seconds, and 273 milliseconds
- "07:00"
  - 7 minutes
- "30m"
  - 30 minutes

## Time Formats

- "1h"
  - 1 hour
- "1.48"
  - 1 second and 480 milliseconds
- "1480ms"
  - 1480 milliseconds (the same as above)

## Clipping Files

- Clipping files is easy, use one of two attributes
  - clipBegin
  - ClipEnd
- They can also be used together
- Time can be given in numerous formats once it is in NPT or SMPTE

## Timing and Synchronisation

- Timing and synchronisation is one of the strengths of SMIL
- You can be as precise or vague as you like
- Relatively easy to present content in parallel or in sequence

## Timing and Synchronising

Possible to present a number of elements in sequence, using <seq>

```
<seq>  
  <video src="ad1.mpg" alt="ad for a new computer"/>  
  <video src="ad2.mpg" alt="ad for an ISP"/>  
  <video src="ad3.mpg" alt="ad for an ad blocker"/>  
</seq>
```



## Timing and Synchronising

Possible to present a number of elements in sequence, using <par>

```
<par>
  <video src="ad1.mpg" alt="ad for a new computer"/>
  <video src="ad2.mpg" alt="ad for an ISP"/>
  <video src="ad3.mpg" alt="ad for an ad blocker"/>
</par>
```

## Timing and Synchronising

It is possible to specify start times for elements in a sequence

```
<seq>
  <video src="ad1.mpg" alt="ad for a new computer"/>
  <video src="ad2.mpg" alt="ad for an ISP"
    begin="00:02"/>
  <video src="ad3.mpg" alt="ad for an ad blocker"
    begin="00:02"/>
</seq>
```

## Timing and Synchronising

It is possible to achieve the same outcome with par

```
<par>
  <audio src="ad_music.mp3" alt="subliminal
    advertising enhancer"/>
  <video src="ad1.mpg" alt="ad for a new computer"/>
  <video src="ad2.mpg" alt="ad for an ISP"
    begin="00:17"/>
  <video src="ad3.mpg" alt="ad for an ad blocker"
    begin="00:34"/>
</par>
```

## Timing and Synchronising

A element can also be timed to start relative to another

```
<par>
  <audio src="ad_music.mp3" alt="subliminal advertising
    enhancer"/>
  <video id="ad1" src="ad1.mpg" alt="ad for a new computer"/>
  <video id="ad2" src="ad2.mpg" alt="ad for an ISP"
    begin="ad1.end+00:02"/>
  <video id="ad3" src="ad3.mpg" alt="ad for an ad blocker"
    begin="ad2.end+00:02"/>
</par>
```

## Timing and Synchronising

Could achieve the same result with nested elements

```
<par>
<audio src="ad_music.mp3" alt="subliminal advertising
enhancer"/>
<seq>
  <video src="ad1.mpg" alt="ad for a new computer"/>
  <video src="ad2.mpg" alt="ad for an ISP" begin="00:02"/>
  <video src="ad3.mpg" alt="ad for an ad blocker"
begin="00:02"/>
</seq>
</par>
```

## Timing and Synchronising

It is possible to repeat a media file

```
<audio src="ring.wav" alt="telephone ring"
repeatCount="4"/>

<audio src="ring.wav" alt="telephone ring"
repeatDur="16s"/>
```

## Timing and Synchronising

Media objects can also be timed to repeat in parallel

```
<par>
<audio id="ring" src="ring.wav"
alt="telephone ring" repeatCount="4"/>

</par>
```

## Timing and Synchronising

To solve the problems with the previous code, specify begin time

```
<par>
<audio id="ring" src="ring.wav"
alt="telephone ring" repeatCount="4"/>

</par>
```

## Timing and Synchronising

It is also possible to synchronise containers as well as media objects

```
<seq>
<audio src="ring.wav" alt="telephone ring"
repeatCount="9"/>
<audio src="operator.wav" alt="operator
saying that the party is not answering"/>
</seq>
```

## Timing and Synchronising

It is also possible to add timers to attributes themselves

```
<seq repeatCount="indefinite">
<audio src="ring.wav" alt="telephone ring"
repeatCount="9"/>
<audio src="operator.wav" alt="operator
saying that the party is not answering"/>
</seq>
```

## Timing and Synchronising

- Possible to synchronise media in sequence and in parallel
- Objects can be timed to begin and/or end relative to another object
- Objects can be repeated, for a particular number of iterations, or a time period
- Iterations and time periods can be linked

## Layout

- SMIL has basic layout capabilities
- Possible to divide layout into regions and assign objects to regions
- Number of layout modules
  - BasicLayout
  - AudioLayout
  - MultiWindowLayout

## Dividing Space into Regions

- SMIL regions are always rectangular
- Arranging media in SMIL is a two step process
  - Region is created
  - Media object is tied to region
- To create a region use the element `<region>`

## Dividing Space into Regions

### Attributes of regions

- Height and Width
- Left, Right, Top and Bottom
- `backgroundColor` and `showBackground`
- `fit`
- `regionName`
- `z-index`

## Dividing Space into Regions

Attributes which are screen measurements can take the following values

- Relative values
  - px, em, or ex
- Absolute values
  - cm, mm, in ,pt, or pc
- Percentages

## Dividing Space into Regions

```
<region id="CIF-NTSC" width="352px" height="240px"/>
```

```
<region id="US-photo" width="6in" height="4in"/>
```

```
<region id="half-center" left="25%" top="25%" width="50%" height="50%"/>
```



## Grouping Regions into a Layout

- Region is not that useful until it is added to a layout
- Layout groups regions similarly to the synchronisation from earlier
- SMIL document only has one layout
- Layout is added to a SMIL document with the <layout> element
- Layout must be in documents <head>

## Grouping Regions into a Layout

```
<head>
  <layout>
    <region id="CIF-NTSC" width="352px"
      height="240px"/>
    <region id="US-photo" width="6in"
      height="4in"/>
    <region id="half-center" left="25%
      top="25%" width="50%" height="50%"/>
  </layout>
</head>
```

## Windows in Layout

- Window refers to a region on a screen
- Can control the appearance and behaviour of the windows your SMIL presentation using <topLayout>
- <topLayout> has five attributes
  - width
  - height
  - backgroundColor
  - open
  - close

## Windows in Layout

```
<layout>
  <topLayout width="640px"
    height="480px"
    backgroundColor="#FFFFFF"
    open="whenActive"
    close="whenNotActive"/>
</layout>
```



## Audio Layout in SMIL

- AudioLayout module the soundLevel attribute for a <region>
- soundLevel is specified as a percentage greater than or equal to 0
- As with other <region> attributes, soundLevel is relative

## Conclusions

- Introduction to W3C and XML
- Introduction to SMIL
- How to create a SMIL file
  - Adding Media Objects
  - Clipping media files
  - Timing and Synchronising
  - Layout
- Next Lecture
  - More about layout
  - Interaction with users
  - Animation
  - Content Control