

# **SMIL 2.0 — Interactive Multimedia on the Web**

**Lloyd Rutledge**

*Multimedia and Human-Computer Interaction Group  
CWI, Amsterdam, The Netherlands  
W3C SYMM working group*

*Lynda Hardman, Jacco van Ossenbruggen: CWI  
Dick Bulterman, Jack Jansen, Sjoerd Mullender: Oratrix  
W3C SYMM working group*

<http://www.cwi.nl/~media/SMIL/Tutorial/{SMIL-4hr.html}>

*Includes material from the upcoming book "SMIL 2.0 — Interactive Multimedia on the Web"*

*Copyright © 2002, Lloyd Rutledge*

# Synchronized Multimedia Integration Language (SMIL)

## Main Points

Pronounced *smile*

Multimedia for the Web — for multimedia what HTML is for hypertext

Integration format for presentable mono-medium formats

## Structure

*SMIL 1.0* — W3C Recommendation on 15th June 1998

*SMIL 2.0* "meta-language" W3C Recommendation on 7th August 2001

SMIL 2.0 family formats *SMIL Profile* and *SMIL Basic* released with SMIL 2.0

SMIL 2.0 family format *XHTML+SMIL* comes after SMIL 2.0

## Main Themes

Powerful timing and synchronization

Adaptive to users and systems

Models a flexible but consistent presentation and user interface

## SMIL Isn't

Flash — Flash is mono-medium animation on steroids

MPEG-4 / 7 / 21 — MPEG looks at content and coding, and player architecture and a whole lot more, but is more media centric than web centric

D-HTML — D-HTML uses scripted definitions of local behaviors, without a notion of the presentation's context

## SMIL 2.0 Profiles

### What is a Profile?

A language for which a browser can be built

A combination of modules from the SMIL 2.0 "meta-language"

Possibly non-SMIL constructs with SMIL constructs

### SMIL 2.0 Language Profile (SMIL Profile)

What is typically thought of as SMIL 2.0

Most of SMIL 2.0 features in one profile

### SMIL 2.0 Basic Language Profile (SMIL Basic)

Intended for mobile devices

Assumes restricted processing ability

### XHTML+SMIL

Applies timing to text-based display

XHTML-based layout

### SMIL 1.0

Backwards-compatible — can be played on SMIL Profile browsers

# SMIL Implementors

## RealNetworks

RealOne for SMIL 2.0

Clear leader for SMIL players

## Oratrix

GRiNS player and editor

Support for all profiles

## Microsoft

Internet Explorer 5.5 and up plays XHTML+SMIL

## Apple

QuickTime 4.1 and up supports SMIL 1.0

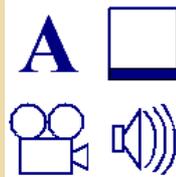
## Adobe

Adobe's SVG Viewer supports SMIL animation in SVG



# What We Need to Specify

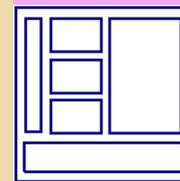
## Media



## Adaptivity



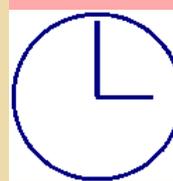
## Layout



## Interaction



## Timing



## miniFiets 2.0 — The Layout

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.0//EN"
    "http://www.w3.org/TR/REC-smil/SMIL20.dtd">
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
  <head>
    <layout>
      <topLayout title="Fiets Amsterdam Tour"
        backgroundColor="black" width="1010" height="665">
        <region regionname="splashScreen" top="5" left="5" bottom="5" right="5"/>
        <region regionname="buildingImage" top="5" right="5" width="875" height="655"/>
        <region regionname="closedCaptioning" bottom="5" left="5" right="5" height="60"/>
        <region title="Thumbnail Bar" top="5" left="5" bottom="5" width="120">
          <region regionname="museumThumb" fit="meet" height="90" top="65"/>
          <region regionname="weighhouseThumb" fit="meet" height="90" top="280"/>
          <region regionname="CWI-INSTThumb" fit="meet" height="90" top="495"/>
        </region>
      </topLayout>
    </layout>
    <transition id="fade1s" type="fade" dur="1s" />
  </head>
```

Color Key

Media

Layout

Timing

Interaction

Adaptivity

## miniFiets 2.0 — The Greeting Section

```
<body>
  <seq>
    <par title="Greeting Section" end="greet.end+1s">
      
      <par id="greet" begin="1s">
        <switch>
          <par systemlanguage="en">
            <audio src="welcome.wav" region="buildingImage"
              alt="welcome to Fiets, your self-guided tour of Amsterdam (spoken)" />
            <text src="welcome.html" region="closedCaptioning" systemcaptions="on"
              alt="welcome to Fiets, your self-guided tour of Amsterdam (captions)"/>
          </par>
          <par systemlanguage="nl">
            <audio src="welkom.wav" region="buildingImage"
              alt="welkom bij Fiets, uw eigen stadswandeling door Amsterdam (gesproken)"/>
            <text src="welkom.html" region="closedCaptioning" systemcaptions="on"
              alt="welkom bij Fiets, uw eigen stadswandeling door Amsterdam (ondertiteling)"/>
          </par>
        </switch>
      </par>
    </par>
  </seq>
```

Color Key

Media

Layout

Timing

Interaction

Adaptivity

# miniFiets 2.0 — The Thumbnail Section

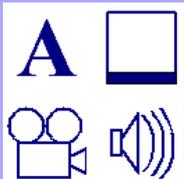
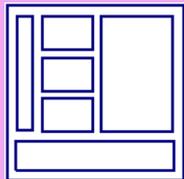
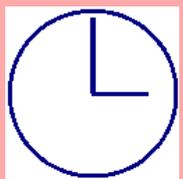
```

<par title="Thumbnail Section" dur="indefinite">
  <par>
    <a href="#museum" alt="Show the Rijksmuseum">
      
    </a>
    <a href="#weighhouse" alt="Show the weighhouse">
      
    </a>
    <a href="#CWI-INS" alt="Show the CWI-INS building">
      
    </a>
  </par>
  <excl dur="indefinite">
    
    
    
  </excl>
</par>
</seq>
</body>
</smil>

```

Color Key  
 Media  
 Layout  
 Timing  
 Interaction  
 Adaptivity

## Outline

	Media	Layout	Timing	Interaction	Adaptivity
Beginner					
	<b>Integration</b> <ref> URIs modes and types	<b>Containing Boxes</b> box elements fit= z-index=	<b>Timelines</b> composition time boundaries time types	<b>Linking</b> <a> and href= playstates volume	<b>Selectivity</b> principles system test <switch>
	<b>Transitions</b> fades and wipes alteration <transitionFilters>	<b>Beyond Boxes</b> hierarchical regions registration alignment opening and closing, sound	<b>Animation</b> targets values functions	<b>Exclusion</b> <excl> <priorityClass> time graphs	<b>Customization</b> custom test skip-content= <prefetch>
Expert	<b>Alteration</b> fragmentation <param> time manipulation	<b>CSS</b> CSS in native SMIL text-flow XHTML+SMIL	<b>Beyond Timelines</b> fill and restart synchronization behavior min= and max=	<b>User Events</b> principles keyboard interaction DOM events	<b>Accessibility</b> core attributes metadata principles

# SMIL Structure

## SMIL Document Classes

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

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd"
<svg xmlns="http://www.w3.org/2000/svg">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+SMIL //EN"
"http://www.w3.org/2001/SMIL20/wd/xhtml1plusmil.dtd"
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:smil="http://www.w3.org/2001/SMIL20">
```

## The <smil> Element

Contains all of a native SMIL document

## The <head> Element

Layout and meta-information, as in HTML

## The <body> Element

The content and linking (as in HTML), and *timing*

## Integration — <ref>

### Media Object Elements

The Waag is a  
medieval buiding in  
Amsterdam

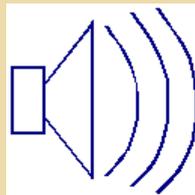


<text>

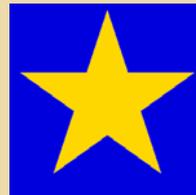
<image>



<video>



<audio>



<animation>

The Waag is a  
medieval buiding in  
Amsterdam. It was  
originally a gate into  
the city through the  
city wall. Later it  
became a  
weighhouse for  
goods being brought

<textstream>

### Brush Media — Draw a rectangle on the screen

The <brush> Element

The color= Attribute

Mostly replicable with <region>

## Integration — URIs

The `src=` attribute holds a URI locating the media

Local file system pathname URIs

relative and absolute

Remove URIs

protocol, server, path

`xml:base=`

base for relative URIs

inherited down subtree

## Integration — Modes and Types

Media Typing

The `mimetype` Construct

Communicating mimetypes through HTTP

Filename suffixes

The `type=` Attribute

Captured vs. Structured Media

General Types

Photographic Images

JPEG, GIF, RealPix, PNG

Video

MPEG, RealVideo

Audio

WAV, RealAudio

Graphics and Animation

GIF, SVG

Text

HTML, Timed Text?

# Containing Boxes — Box Elements

## Presentation Windows

The `<topLayout>` Element — General-purpose Window

The `<root-layout>` Element — The Main or Only Window

## The `<region>` Element

Each media object instance contains a region reference:

allows author to know where object will be played

```
<video src="anchor.mpg" region="V-main"/>
```

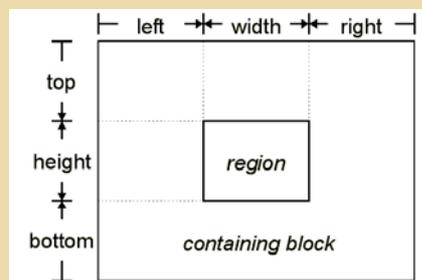
## The region is defined by:

A `regionName=` or `id=` attribute for each region is required

### Region positioning attributes

Length values are percentage values or pixels

The unit "px" may be omitted



# Containing Boxes — fit=



hidden (default)

media item  
not scaled



hidden (default)

media item  
not scaled



scroll

media item  
not scaled



meet

aspect ratio  
preserved



slice

aspect ratio  
preserved



fill

aspect ratio  
not preserved

## Containing Boxes — z-index=

When regions overlap, the z-index attribute determines which image appears on top

z-index= determines the *stacking order*

highest integer stacks on top

Stack level tie-breakers

most recently started is on top

Stacking context is inherited through nested regions

Stack levels are inherited by default

## Timelines — Composition

<seq> (sequence) — each child starts when previous ends

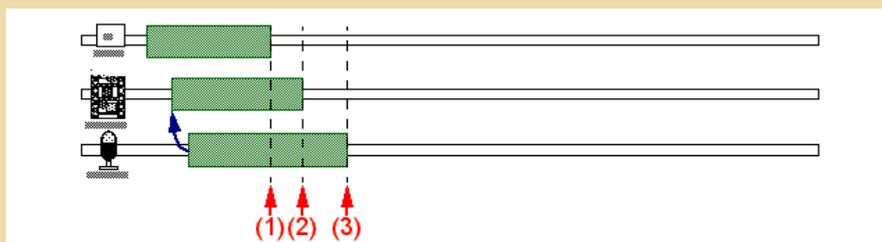
<par> (parallel) — all children start together

<par>'s and <seq>'s can be nested

endsync=

(1) <par> can end when the ="first" element to finish ends

```
<par endsync="first">  
  <text src="leader_title.html" region="m_title" dur="5s"/>  
  <video id="v1" src="cnn.mpg" region="v-Main" begin="1.4s"/>  
  <audio src="cnn.aiff" region="music" begin="id(v1)(0.5s)"/>  
</par>
```



(2) <par> can end when the referenced element ends: id( Id-value )

(3) <par> can end when the ="last" element to finish ends (default)

## Timelines — Time Boundaries

begin= &mdash in terms of default begin

end= &mdash in terms of default begin>

dur= &mdash *explicit* duration overrides implicit

repeatDur= — loop for specified period

repeatCount= — loop specified number of times

= "indefinite" — nothing from element or its content triggers this time boundary

## Timelines — Which Time?

### Durations



**Intrinsic**

**Explicit** — dur= clock-value, "indefinite"

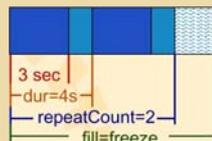


**Simple**



**Active**

**Perceived**



### Activation/Termination

**Resolved times**

**Unresolved times**

**Definite times**

**Indefinite times**

## Linking — <a> and href=

### The <a> Element

Contains media that is link trigger

### The href= Attribute

URI value refers to link destination

### Linking within SMIL Presentations

The href="#*idref*' Attribute Assignment

Moving back and forth along the timeline (instead of scrolling)

## Linking — Playstates

### Play Spaces for the Link Destination

The show= Attribute

The external= Attribute — open with external program

The target= Attribute — region or other display space to show in

### Play States

The sourcePlaystate= Attribute

The destinationPlaystate= Attribute

## **Linking — Sound**

**The sourceLevel= Attribute**

**The destinationLevel= Attribute**

## **Selectivity — Principles**

**Each element must pass a "playability" test**

**Those not selected are ignored and not played**

**Example: captions not played if user does not desire them**

## Selectivity — System Test Attributes

### System Test Attributes

Standard collection of conditions to test for for selectivity

Test attributes can be combined

### Adaptation to User

The systemLanguage= Attribute

The systemCaptions= Attribute

The systemOverdubOrSubtitle= Attribute

The systemAudioDesc= Attribute

### Adaptation to Hardware

The systemBitrate= Attribute

The systemCPU= Attribute

The systemScreenSize= Attribute

The systemScreenDepth= Attribute

### Adaptation to Software

The systemOperatingSystem = Attribute

The systemComponent= Attribute

The systemRequired= Attribute

## Selectivity — <switch>

At most one of the children of a switch element is played

The first acceptable element is chosen

Thus, ordering should be more preferred first

Last child can be always-accepted default

Acceptability based on anything the browser wants

Example: versions for different languages

## Transitions — Fades and Wipes

### The <transition> Element

In document head

Defines one transition for use in presentation

### Types of Transitions

The type= Attribute — Which of the fixed set of transition types to use

The subtype= Attribute

Some transition types have subtypes

Such as type = "barWipe" with subtypes = "leftToRight" and = "topToBottom"

## Transitions — Alteration

### Controlling the Transition

The direction= Attribute — = "forward" (default) or = "reverse"

The fadeColor= Attribute — color value for what is faded from/to

The startProgress= and endProgress= Attributes

How far through default transition to start/end

### Transition Modifiers

The horzRepeat= and vertRepeat= Attributes

Repeating the transition in each direction

The borderColor= and borderWidth= Attributes

Changing the appearance of wipe borders

## Transitions — <transitionFilter>

### Selecting a Transition for the Presented Element

The **transIn=** and **transOut=** Attributes — ID of Transition to Use

### Inline Transitions — Transitions Defined in the Presentation Body

#### The <transitionFilter> Element

Had transition-defining attributes

Applies to it's parent, by default

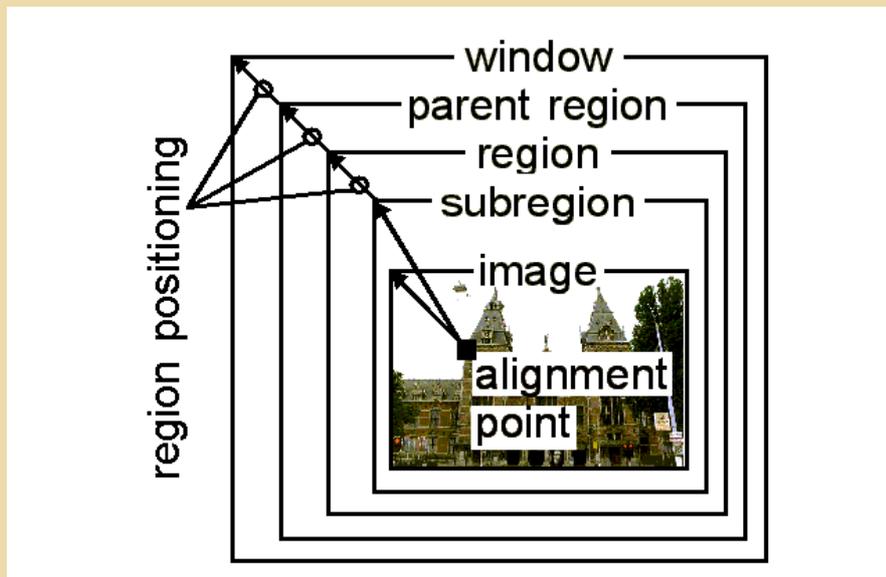
#### The **targetElement=** Attribute

ID of element other than parent to apply this transition to

#### The **mode=** Attribute

The parent/target element transitions = "in" or "out"

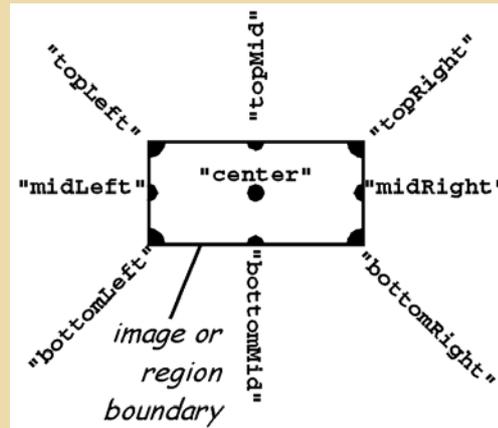
## Beyond Boxes — Hierarchical Regions



## Beyond Boxes — Registration Alignment

### Sub-regions — Precise Image Positioning within Regions

regPoint= and regAlign= Alignment



## Beyond Boxes — Opening and Closing, Sound

### Opening and Closing Regions and Windows

#### Active Regions and Windows

#### The showBackground= Attribute for Regions

The "always" and "whenActive" values

#### The open= Attribute — Opening Windows

The "onStart" and "whenActive" values

#### The close= Attribute — Closing Windows

The "onRequest" and "whenNotActive" values

### The soundLevel= attribute

Percentage value, purely relative

## Animation — Targets

### Animation Elements

The `<animate>` Element — change an attribute value over time

The `<animateMotion>` Element — move an object across the screen

The `<animateColor>` Element — change an object's color

The `<set>` Element — assign an attribute a value for a period of time

### Specifying the Animation Target

The `href=` or `targetElement=` Attribute — ID element with attribute to animate

The `attributeName=` Attribute

The `attributeType=` Attribute — `"XML"`, `"CSS"`, or `"auto"` automatically determine which

## Animation — Values

### Listing the Animation Values

The `from=` and `to=` Attributes — starting and ending values of attribute in animation

The `by=` Attribute — incremental value change

The `values=` Attribute — list of values the attribute goes through

## Animation — Functions

### Defining the Animation Function

#### The calcMode= Attribute

= "discrete", = "linear" or = "paced"

#### The accumulate= Attribute

Does repeating animation build on top of previous iterations?

#### The additive= Attribute

= "sum" with the assigned value or = "replace" it?

#### The origin= Attribute

Position <animateMotion> starts from

### Spline Animation

#### The path= Attribute

SVG path syntax for specifying a curve

#### The keyTimes= Attribute

A time offset for each value

#### The keySplines= Attribute

A curve for smooth value changing between each value

## Exclusion — <excl>

No more than one child can play at any one time

One child starting will cause any playing siblings to stop

Analogous to radio buttons

## Exclusion — <priorityClass>

Determines what happens when one child starts while other plays

### Priority Class Values

The ="stop" value

The ="pause" value

The ="defer" value

The ="never" value

### Priority Class Attributes

The peers= Attribute

The lower= Attribute

The higher= Attribute

### The pauseDisplay Attribute

The ="disable" value

The ="hide" value

The ="show" value

## Exclusion — Time Graphs

Temporal exclusion brings up beyond single time line

Time graph?

Time blob?

Children of <excl> can change while <par> siblings of <excl> keep on playing along their timelines

Enables *linking in context*

Example: background music keeps playing while images switched

Sync between parallel components only known at run-time

## Customization — Custom Test Attributes

### Principles

Anyone can define adaptive test attributes for use in SMIL

Won't be recognized by all browsers

Potential examples — knowledge level, audience profile, length of time

### The `<customAttributes>` Element

Located in the document head

Contains all `<customTest>` test definitions

### The `<customTest>` Element

Definition of one custom test for use in the document body

### The `customTest=` Attribute

IDs of all `<customTest>` test this body element must pass

### The `uid=` Attribute

URI of global definition of custom test

### The `defaultState=` attribute

### The `override=` Attribute

Can user override assignment to body element(s)

## Customization — `skip-content=`

### `skip-content` Attribute

How to adapt for SMIL "dialects"

Ignore unknown elements within sub-tree or ignore whole sub-tree

## Customization — <prefetch>

Control, timing, and adaptation of pre-loading media before its presentation

Helps whole presentations progress with fewer hitches

### The <prefetch> Element

Schedules the prefetching of a particular media item

src= attribute determines this item

### The mediaSize= Attribute

How much of media item's data to prefetch

### The mediaTime= Attribute

How much of media item's time to prefetch

### The bandwidth= Attribute

How much of the presentation's bandwidth to use for this prefetch

## Alteration — Fragmentation

### Spatial

coords= attribute (similar to HTML image maps)

The shape= and nohref= attributes

### Temporal

The clipBegin= and clipEnd= attributes — clip with *measured* moments

The "marker=" Substring — clip with *named* moments

Temporal subparts use the begin= and end= Attributes

### Nominal

Media Marker Timing — *Synchronization with Named Media Portions*

The ".marker()" Substring for begin= and end= Attributes

Object Linking — *Linking from Named Media Portions*

The fragment= Attribute for <area> Elements

XPointer for SMIL URIs in general

## Alteration — <param>

### The <param> Element — Passing One Parameter to the Media Presenter

The name= Attribute — the parameter's name

The value= Attribute

The valueType= Attribute

= "data", URI = "ref" or media = "object" element elsewhere in presentation

### The erase= Attribute — when to end display after active period ends

= "whenDone" — when the element ends

= "never" — after element ends, until replaced on display surface

### The mediaRepeat= Attribute

= "preserve" (default) or = "strip" repeat intrinsic in media

Example — strip animated GIF repeat so SMIL can repeat it instead

### The sensitivity= Attribute

Can user interaction pass through this display to underlying media?

## Alteration — Time Manipulation

### The speed= Attribute

What proportion of the intrinsic speed should the media be played back at?

### Acceleration and Deceleration

Duration of affected element remains the same

The accelerate= Attribute

What proportion of playback is accelerated, starting still at beginning

The decelerate= Attribute

What proportion of playback is decelerated to standstill at end

*Run-rate Interval* in between may be sped up to maintain overall duration

### The autoReverse= Attribute

Media plays in reverse after playing forward

Doubles simple duration of element

Any repeating applies after reversing

## **CSS — CSS in native SMIL**

### **CSS Code for SMIL Layout**

**The type= Attribute**

### **Scope of CSS vs scope of SMIL Layout**

**Non-modularized CSS induces too much overhead**

### **Conceptual limitations of CSS**

**Multiple top-level windows and independence of objects across layout windows is a problem**

## **CSS — Text-flow**

### **Differences in Text-Flow vs Time-Flow Documents**

**The XML nesting tells you a lot about text layout**

**The XML nesting tells you very little about temporal layout**

# CSS — XHTML+SMIL

## CSS for Media-based SMIL

### XHTML+SMIL and SVG

## Time Containers for Non-native SMIL

### The `timeContainer=` Attribute

Makes any element a time container, such as a `<p>` in XHTML+SMIL

`= "par", = "seq", = "excl" or = "none"`

### The `timeAction=` Attribute

What aspect of the element is affected by timing

`= "intrinsic", = "display", = "visibility", = "style", = "class" or = "none"`

## Beyond Timelines — Fill and Restart

### Time Slot Filling

#### Elements can be active longer than their media

Ended media play "echoes" in its still active element

Non-timed media (text, images) ends immediately

#### The `fill=` and `fillDefault=` attributes

`= "remove", = "freeze", = "hold" or = "transition"`

`= "auto", = "inherit" or = "default"`

### Restart Timing

#### The `restart=` Attribute — Can element start again from beginning if playing?

`= "always", = "whenNotActive" or = "never"`

or `= "default"` — use `restartDefault=` attribute

#### The `restartDefault=` Attribute

`= "always", = "whenNotActive" or = "never"`

or `= "inherit"` — get from parent

## Beyond Timelines — Synchronization Behavior

### The syncBehavior= Attribute

- = "locked" — for lip synchronization
- = "canSlip" — for background music
- = "independent" of seeks on parent

### The syncTolerance= Attribute

How much time media can fall out of sync by

### The syncMaster= Attribute

One element is "conductor" for whole container

## Beyond Timelines — min= and max=

### MinMax Timing — Constraining the Active Duration

- The min= Attribute — Minimum for active duration
- The max= Attribute — Maximum for active duration

## User Events — Principles

User Interaction as Unpredictable Events in Time

SMIL Link Elements are Forward-reaching Links

The `actuate="auto"` assignment

firing the link when element starts, regardless of interaction

SMIL Inline Synchronization Attributes are Backward-reaching Links

## User Events — Keyboard Interaction

Order of links to focus by consecutive tab key clicks

The `tabindex=` attribute for the `<a>` and `<area>` elements

What key click focusses this link

The `accesskey=` Attribute for the `<a>` and `<area>` elements

The `".accesskey()"` Substring for `begin=` and `end=` Attributes

## User Events — DOM Events

The “.event()” Substring

Non-Interactive Events

Interactive Events

Link Elements as Forward-reaching Events

## Accessibility — Core Attributes

id=, title=, alt=, longdesc=, class=, xml:base=, xml:lang=

Attributes on media object elements

**The alt= Attribute — contains alternative text**

**The longdesc= Attribute**

supplement to alt, but longer and should include descriptions of areas

**The readIndex= Attribute — Order to read aloud for sight-impaired**

XML Attributes for Adaptation

**The title= Attribute**

**The xml:lang= Attribute**

## Accessibility — Metadata

**<meta> element defines properties of a document**

**The name= attribute is the property and the content= attribute gives the value**

```
<meta name="title" content="web News, 15th June 1998"/>  
<meta name="base" content="http://www.cwi.nl/SMIL/webnews/" />
```

**The list of properties (values of name attribute) is open-ended**

**The <metadata> Element — RDF content**

## Accessibility — Principles

**W3C Web Accessibility Initiative (WAI)**

**Guidelines for accessible (text-based) Web documents**

**Meaningful values for attributes like alt=, title=, abstract= and longdesc=**

**Meaningful content of link triggers (<a> element)**

**How to apply these to a fixed timeline?**

**Temporal Adaptation**

**Handling delays of download and processing**

**Explicit and implicit time**

**Temporal hierarchy of parallel and sequential composites**

sets points in presentation progression for stronger synchronization

**Layout Adaptation in SMIL**

**SMIL documents can adapt to devices with different screen sizes**

layout relative to the dimensions of the player's viewport

alternative layout strategies

**Switch on layout and region**

Allow assigning test attributes to SMIL layout and region elements

**Examples**

make room for subtitles

rearrange for varying screen size

# What's Next?

## SMIL 2.0 is a Recommendation

**Draws attention to the standard**

**Stabilized to enable wide-spread implementation and adoption**

**First players scheduled for release with recommendation**

GRiNS Player for the SMIL 2.0 recommendation is already available

Internet Explorer 6.0 for XHTML+SMIL out now

RealOne emerging with increasingly more SMIL 2.0 support

## SMIL 2.0 becomes more implemented

**More browsers introduced**

**More existing Web browsers add SMIL to languages shown**

**SMIL browsers show more and more media**

SVG?, All show XHTML?

## SMIL 2.0 becomes more used

### SMIL Future Developments

**The SMIL Profile for High-End, Large-scale Media Distribution**

**SMIL Basic and the Emerging Mobile Media Market**

**Future SMIL Formats**

**Further Development of SMIL Itself**