SMIL 2.0 Summary

DAN OLTEANU

INSTITUT FÜR INFORMATIK, LMU MÜNCHEN

5.02.2001

SMIL 2.0 Modules

- Basic Layout Module
- MultiWindowLayout Module
- Hierarchical Module
- Structure Module
- Basic LinkingAttributes Module
- MediaObjects Module
- Timing and Synchronization Module
- Time Manipulations Module

Basic Layout Module

- **layout** element (type ='text/smil-basic-layout')
  - ➜ **region** (top, left, width, height - abs/rel values)
    - ➜ fit (fill, hidden, meet, scroll, slice)
    - ➜ z-index - specifies the visibility for overlapping regions
  - ➜ **root-layout** (backgroundColor, height, width)

MultiWindowLayout Module

- creation and control of multiple top level windows
- rendering containers, without temporal significance
- **viewport** element (backgroundColor, close, open)
  - ➜ close (never, whenNotActive)
  - ➜ open (always, whenActive)

HierarchicalLayout and Structure Module

- Structure
  - ➜ **smil** element (id, xmlns, title)
    - ➜ **head** - has layout information
    - ➜ **body** - has information related to the temporal and linking behaviour of document
- HierarchicalLayout
  - ➜ z-index conventions, while working with overlapped subregions
  - ➜ **regPoint** (right, bottom, id, top, left, regAlign)
    - ➜ regAlign (topMid, topLeft, topRight; midLeft, midRight, center;..)

Basic MediaObject Module

- Elements
  - ➜ **ref** - generic media reference
  - ➜ **audio**
  - ➜ **text, textStream**
  - ➜ **video**

- Attributes
  - ➜ abstract, alt, author, copyright, src, title, type ..

BasicLinking Module

- provides unidirectional, single-headed, inline link elements
- Elements
  - ➜ **a** - HTML behaviour
  - ➜ **area** - temporal/spatial segments

Timing and Synchronization Module

- Attributes
  - ➜ begin
  - ➜ dur (controls the simple duration)
  - ➜ end
  - ➜ min and max (controls the active duration)
  - ➜ endsync (controls implicit duration of time containers par and excl)
  - ➜ repeatCount and repeatDur (repeating elements)
  - ➜ fill (fills the temporal gaps between medias)
  - ➜ restart
- Elements (time containers)
  - ➜ par
  - ➜ seq
  - ➜ excl, priorityClass

Begin value semantics

```
begin-value-list ::= begin-value (S ";" S begin-value-list )?
begin-value      ::= (offset-value | syncbase-value
                      | syncToPrev-value | event-value
                      | repeat-value | accessKey-value
                      | media-marker-value | wallclock-sync-value
                      | "indefinite" )
```

BEGIN EXAMPLES

- syncbase begin timing

```
<par>
   <audio src="song1.au" begin="5s" dur="4s" />
</par>
```

- Begin time in the past

```
<par>
   <video id="vid" begin="-5s" dur="10s" src="movie.mpg" />
   <audio begin="vid.begin+2s" dur="8s" src="sound.au" />
</par>
```

- Unresolved begin time

```
<ref begin="foo.click-8s" dur="3s" repeatCount="10" .../>
```

DUR ATTRIBUTE

- Values : (clock-value | media | indefinite)
- Examples
  - ➜ Simple offset begin timing

```
<par>
   <audio src="song1.au" begin="5s" dur="4s" />
</par>
```

  - ➜ Unresolved begin time

```
<text id="show" ... />
<img begin="show.click" dur="3.5s" ... />
```

END ATTRIBUTE: CONTROLLING THE ACTIVE DURATION

- End value semantics

```
end-value-list ::= end-value (S ";" S end-value-list )?
end-value      ::= (offset-value | syncbase-value
                    | syncToPrev-value | event-value
                    | repeat-value | accessKey-value
                    | media-marker-value | wallclock-sync-value
                    | "indefinite" )
```

- Examples
  - ➜ Simple ending time

```
E1: <video end="10s" src="8-SecondVideo.mpg" .../>
E2: <video dur="media" end="click" src="movie.mpg" .../>
E3: <audio src="music.au" dur="20s" end="10s" ... />
```

  - ➜ Medias which might not start

```
<par>
   <audio src="music.au" begin="gobtn.click" repeatDur="indefinite"
          end="30s" ... />
   <img src="foo.jpg" dur="40s" ... />
</par>
```

MIN AND MAX ATTRIBUTES: MORE CONTROL OVER THE ACTIVE DURATION

- Value semantics

```
Min : (clock-value | media )
Max : (clock-value | media | indefinite )
```

- Examples

```
E1: <video id="video_of_15s" max="10s".../>
E2: <video id="video_of_15s" repeatDur="indefinite" end="click" min="10s".../>
E3: <par endsync="first" min="12s" fill="freeze" >
        <video id="video_of_15s" end="click" ...>
        <video id="video_of_10s" .../>
    </par>
E4: <par dur="5s" min="12s" > {fill=''remove'' by default}
              <video id="video_of_15s"/>
              <video id="video_of_10s" />
    </par>
E5: <img id="img" begin="-5s" min="7s" dur="5s" .../>
E6: <img id="img" begin="-5s" min="4s" dur="2s" .../>
```

EndSync attribute: the implicit duration of time containers

- Value semantics

```
EndSync : (first | all | last | media | Id-value of a child)
```

REPEATING ATTRIBUTES: REPEATCOUNT AND REPEATDUR

- Value semantics

```
repeatCount : (numeric-value | indefinite)
repeatDur : (clock-value | indefinite)
```

- Examples (Active/Simple Duration)

```
E1: <par dur="9s">
         <audio src="3second_sound.au" repeatCount="100" />
    </par>
E2: <audio src="background.au" dur="2.5s" repeatCount="2" />
E3: <audio src="music.mp3" dur="2.5s" repeatDur="7s" />
E4: <img src="foo.jpg" repeatCount="2" />
    <img src="bar.png" dur="indefinite" repeatCount="2" />
E5: <text src="intro.html" repeatDur="10s" />
E6: <seq begin="5s" repeatCount="3" >
              <img src="img1.jpg" dur="5s" />
              <img src="img2.jpg" dur="4s" />
              <img src="img3.jpg" dur="4s" />
    </seq>
```

FILL ATTRIBUTE: EXTENDING AN ELEMENT

- Value semantics

```
fill: (remove | freeze | hold | transition | auto | default)
```

- Examples

```
E1: <par endsync="last">
              <video src="intro.mpg" begin= "5s" dur="30s" fill="freeze" />
              <audio src="intro.au"  begin= "2s" dur="40s"/>
    </par>
E2: <seq> (while v2 is loaded, v1 is freezed)
              <video id="v1" fill="freeze" src.../>
              <video id="v2" begin="2s" src.../>
    </seq>
```

RESTART ATTRIBUTE

- Value semantics

```
restart: (always | whenNotActive | never | default)
```

- Examples

```
E1: <img id="go_btn" dur="indefinite" .../>
    <video id="foo" begin="go_btn.click" ... />
    <audio id="bar" begin="foo.begin+2s" dur="10s"
           restart="whenNotActive" ..." />
E2: <img id="foo" begin="bar.click" end="bar.click"
         restart="whenNotActive" ... /> (toggle activiation)
```

TIME CONTAINERS: DYNAMIC VIDEO SUB-TITLES EXAMPLE

```
<par>
   <video id="vid1" .../>
   <excl>
      <par begin="englishBtn.click" >
         <audio begin="vid1.begin" src="english.au" />
      </par>
      <par begin="frenchBtn.click" >
         <audio begin="vid1.begin" src="french.au" />
      </par>
      <par begin="swahiliBtn.click" >
         <audio begin="vid1.begin" src="swahili.au" />
      </par>
   </excl>
</par>
```

PriorityClass:the behaviour of the medias inside a monitor

- Attributes

```
peers: (stop | pause | defer | never)
higher: (stop | pause)
lower: (defer | never)
```

- Example (computed times and pause queue)

```
E1: <img "joe" end="10s" .../>
    <excl dur="indefinite">
      <priorityClass peers="pause">
          <img id="foo" end="joe.end" .../>
          <img id="bar" begin="8s" dur="5s" .../>
      </priorityClass>
    </excl>
E2: <excl>
      <priorityClass peers="pause">
          <img src="image1.jpg" begin="0s" dur="5s"/>
          <img src="image2.jpg" begin="0s" dur="5s"/>
          <img src="image3.jpg" begin="0s" dur="5s"/>
      </priorityClass>
    </excl>
```

Time Manipulations Module

- Goal
  - ➜ manipulation of time behaviour
  - ➜ suited to animation and non-linear or discrete media
  - ➜ model of cascading time
- Attributes
  - ➜ *speed* - controls the pacing/speed of element active time
  - ➜ *accelerate, decelerate* - "Ease-In, Ease-Out"
  - ➜ *autoReverse* - "Play Forward, Then Backward"

Speed Attribute

- regards the element *active duration*
  - ➜ affected attributes: *repeatDur, min, max*
  - ➜ not affected attributes : *begin, end*

- specifies a value (number) *relative* to the parent tc speed

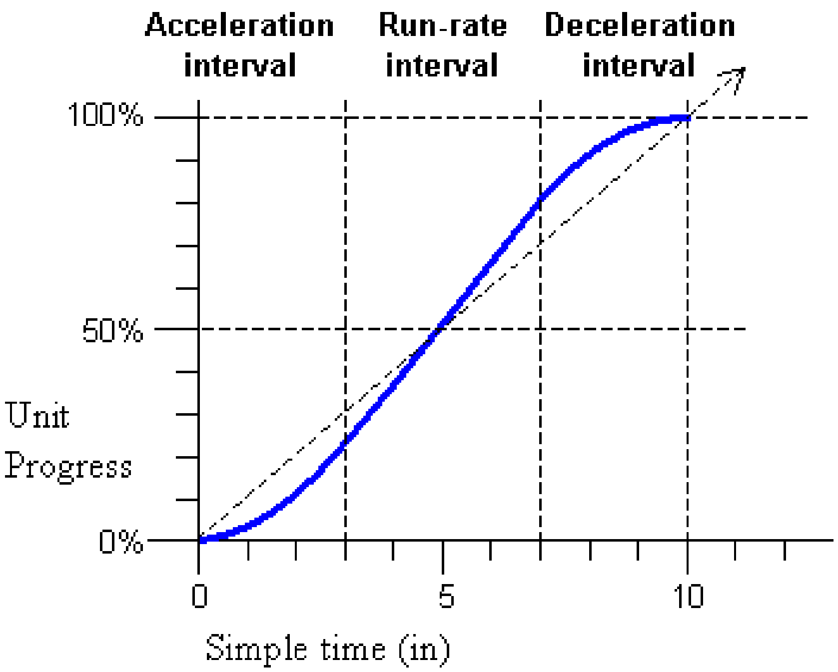
```
E1: <animateMotion dur="10s" repeatCount="2" speed="2.0" path=.. />
E2: <par speed=2.0>
      <animate begin="2s" dur="9s" speed=0.75 ../>
    </par>
E3: <par speed=2.0>
      <animate begin="2s" dur="9s" speed=0.75 repeatCount="4" end="click" ../>
    </par>
E4: <par speed=2.0>
      <img id="foo" dur="30s" .../>
      <animate dur="9s" speed=0.75 repeatCount="4" end="click; foo.end" .../>
    </par>
E5: <par speed=2.0>
      <img id="foo" dur="30s" .../>
      <animate begin="2s" dur="9s" speed=0.75 repeatCount="4" end="foo.end+6s"../>
    </par>
```

Accelerate and Decelerate Attributes

- no effects upon the length of the *simple/active duration*
- *run rate* - modified speed to compensate acc/dec
  ➜ $runrate = \frac{1}{1 - \frac{acc}{2} - \frac{dec}{2}}$, acc + dec ≤ 1

```
E1: <animateMotion dur="8s" accelerate=".25" decelerate=".25" .../>
E2: <animate attributeName="left" dur="4s" decelerate=".25"
             from="-1000" to="0" additive="sum" />
```

Example




```
<animation dur="10s" accelerate="0.3" decelerate="0.3" .../>
```

Figure 1: Effect of acceleration and deceleration upon the rate of progress.

AUTOREVERSE ATTRIBUTE

- regards the element *simple/active duration*
- doubles a declared or implicit *simple duration*

```
E1: <animateMotion by="20, 0" dur="5s" autoReverse="true" repeatCount="3"/>
E2: <animateMotion by=.. dur=.. autoReverse=.. repeatDur="15" end="click"/>
E3: <animateMotion by=.. dur=.. autoReverse=..  accelerate=".25" decelerate=".25" />
E4: <animateTransform type="rotate" from="20" to="-20" dur="1s"
     repeatCount="indefinite" accelerate=".5" decelerate=".5" autoReverse="true" />
```

BASIC TRANSITION EFFECTS MODULE

- Goal
  - ➜ passing from one media object to another, in a *progressive* way
  - ➜ provides *visual transitions* only
  - ➜ specifies transitions via a shorthand method
- Elements and Attributes
  - ➜ transition
  - ➜ transIn, transOut

THE TRANSITION ELEMENT ATTRIBUTES

- type - (barWipe, irisWipe, clockWipe, snakeWipe)
- subtype
- dur - duration of transition
- startProgress (0.0, 1.0)
- endProgress (0.0, 1.0)
- direction (forward, reverse)

TransIn and TransOut Attributes

- transIn - at the beginning of the media's active duration
- transOut - at the end of the media's active duration

```
E1: <transition id="toblack1s" type="fade" subtype="fadeToColor"
                fadeColor="#000000" dur="1s"/>
    <par>
      <img   ... dur="10s" transOut="toblack1s" fill="freeze"/> {fill="remove"}
       <video ... dur="30s" transOut="toblack1s"/>
    </par>
E2: <transition id="awipe"   type="barWipe"    dur="1s" ... />
    <transition id="toblack" type="fadeToColor dur="1s" ... />
    <img src="img1.jpg" dur="1.5s" transIn="awipe" transOut="toblack" .../>
E3: <seq>
     <img src="img1.jpg" dur="5s" fill="transition" .../>
     <img src="img2.jpg" dur="5s" repeatCount="3" transIn="awipe" transOut="toblack"/>
     <img src="img3.jpg" dur="5s"/>
    </seq>
```

MORE TRANSITION EXAMPLES

- Slideshow example with transitions

```
<transition id="wipe1" type="barWipe" subtype="leftToRight" dur="1s"/>
<seq>
 <img src="butterfly.jpg" dur="5s" fill="transition" ... />
 <img src="eagle.jpg"     dur="5s" fill="transition" transIn="wipe1" ... />
 <img src="wolf.jpg"      dur="5s" fill="transition" transIn="wipe1" ... />
 <img src="seal.jpg"      dur="5s"                   transIn="wipe1" ... />
</seq>
```

- Exclusive children and fill="transition"

```
<transition id="wipe1" type="barWipe" subtype="leftToRight" dur="1s"/>
<excl>
 <img src="butterfly.jpg" begin="0" dur="5s" fill="transition" transIn="wipe1"/>
 <img src="eagle.jpg"     begin="b1.click" dur=.. fill=.. transIn=../>
 <img src="wolf.jpg"      begin="b2.click" dur=.. fill=.. transIn=../>
 <img src="seal.jpg"      begin="b3.click" dur=.. fill=.. transIn=../>
</excl>
```

Inline Transitions Module

- Goal
  - ➜ passing from one media object to another, in a *progressive* way
  - ➜ provides *visual transitions* only
  - ➜ specifies transitions via an inline method
- Element
  - ➜ transitionFilter

The transitionFilter Element Attributes

- type - (barWipe, irisWipe, clockWipe, snakeWipe)
- subtype
- mode - (in, out)
- begin, dur, end, repeatCount, repeatDur
- from, to, by - (0.0, 1.0)
- values - list of (0.0, 1.0)
- direction - (forward, reverse)
- calcMode - (discrete, linear)

TRANSITIONFILTER EXAMPLES

```
E1: transition filter slideshow
   <seq>
     <img id="butterfly" src="butterfly.jpg" dur="5s" fill="transition" />
     <img id="eagle" src="eagle.jpg" dur="5s" fill="transition" >
      <transitionFilter type="barWipe" subtype="leftToRight" dur="1s" />
     </img>
     <img id="wolf" src="wolf.jpg" dur="5s" fill="transition" >
      <transitionFilter type="barWipe" subtype="leftToRight" dur="1s" />
     </img>
     <img id="seal" src="seal.jpg" dur="5s" >
      <transitionFilter type="barWipe" subtype="leftToRight" dur="1s" />
     </img>
    </seq>
E2: transitionFilter discrete clock transition
  <video id="video1" src="car.avi">
   <transitionfilter id="trans1" type="ellipseWipe" subtype="circle"
    begin="2" dur="12" calcMode="discrete" values="0.083; 0.166;
    0.250; 0.333; 0.416; 0.500; 0.583; 0.666; 0.750; 0.833; 0.916; 1.000;" />
  </video>
```