

Time and Animations in Web Applications

Guido Grassel

Lecture at Helsinki University of Technology

20 Feb 2008

About Myself

Guido Grassel

Email: guido.grassel@nokia.com

Research Team Leader, Web Technology and Usability
Nokia Research Center, Helsinki

Ex. co-chair W3C Synchronized Multimedia Working Group,
and co-editor of SMIL 2.1

Co-creator of S60 Open Source Browser
and Mini Map Web page navigation method.

Contents

- Motivation and Use Cases
- Procedural approaches
 - Do-it-yourself
 - JavaScript frameworks: YUI Animation
- Declarative Approaches
 - CSS Animations and Transformations in WebKit
 - SMIL and SVG Animations
 - Timesheets (by Prof. Vuorimaa)
- Conclusions

Example: Multimedia on the Web

lonelygirl15 Song / Video parody / lonelygirl: DEE



This is a video response to [My Surprise For Daniel](#)

Rate this video: ☆☆☆☆ ☆ 1404 ratings

[Save to Favorites](#) [Add to Groups](#) [Share Video](#) [Post Video](#) [Flag as Inappropriate](#)

Views: **48,228** | Comments: **707** | Favorited: **737** times

Honors: [3](#) | Links: [5](#) | Responses: [7](#)

Added **November 09, 2006** [SUBSCRIBE](#)
From [Deemontreal](#) to Deemontrea

Provided By: [Deemontreal](#)

GET THIS SONG:
[Myspace.com/deemontrea...](#) [\(more\)](#)

Category [Music](#)

Tags [song](#) [lonelygirl15](#) [video](#) [parody](#) [\(more\)](#)

URL <http://www.youtube.com/watch?v=Fc...>

Embed `<object width="425" height="350"><`

Related [More from this user](#) | [Playlist](#)

Showing 1-20 of 30 [See All Video](#)

- [Re: lonelygirl15 Song / Video parody / lonelygirl: DEE](#)
01:34
From: [GiZmozor](#)
Views: 607
- [RE:lonelygirl15 Song / Video parody / lonelygirl: DEE](#)
00:33
From: [kindaz](#)
Views: 1880
- [Re: lonelygirl15 Song / Video parody /](#)

Example: Multimedia on the Web

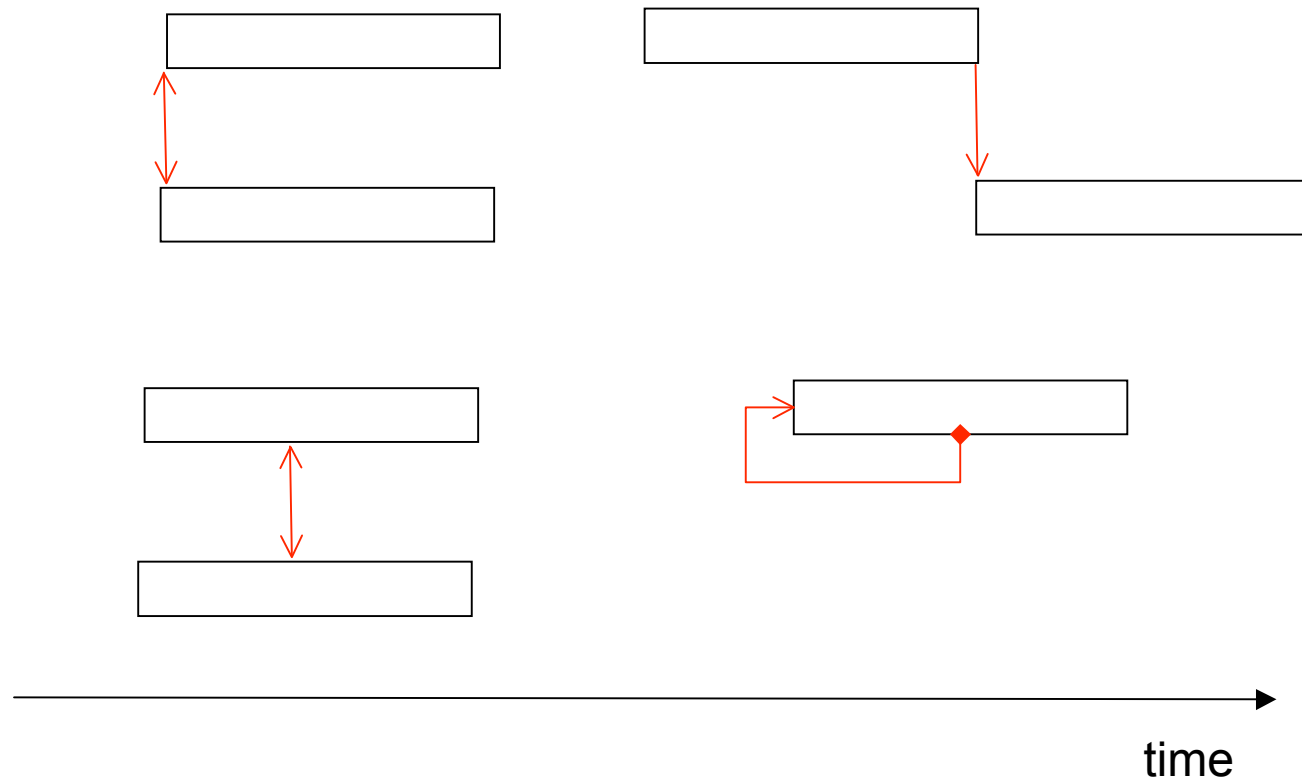


Time and Animations on the Web

- Continuous media introduces a notion of time to Web pages.
 - Trigger the start of an audio/video
 - Trigger on the end of an audio/video
 - Trigger on audio/video reaching a specific temporal position / or an event.
- Slide shows
 - Time-dependent and user-initiated forward
- Time and timers
- (Useful) eye-candy
 - Gradual style changes instead of hard cut.
 - Raising attention.

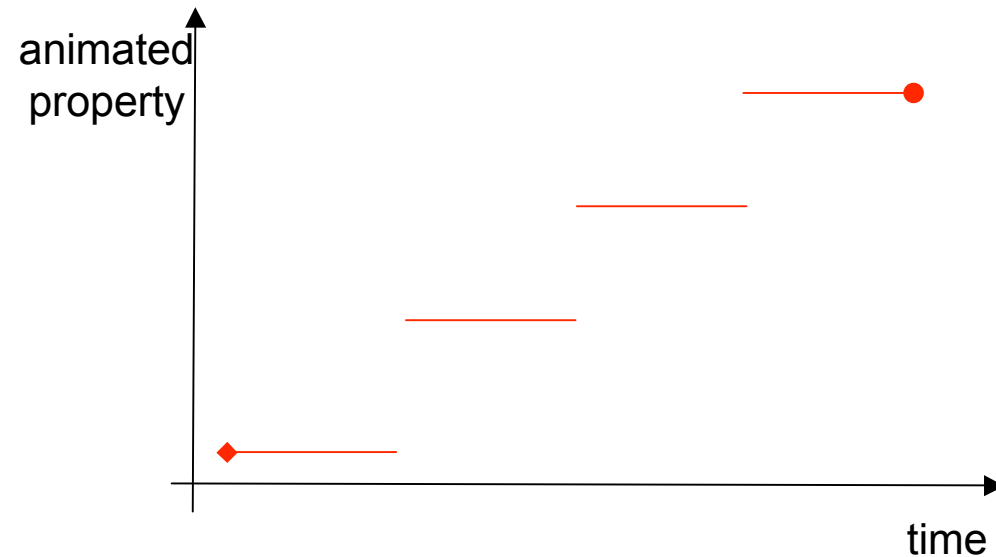
Time-dependent synchronization

- Common types of time-dependent synchronization:
 - start-start
 - end-start
 - strictly parallel
 - user interaction



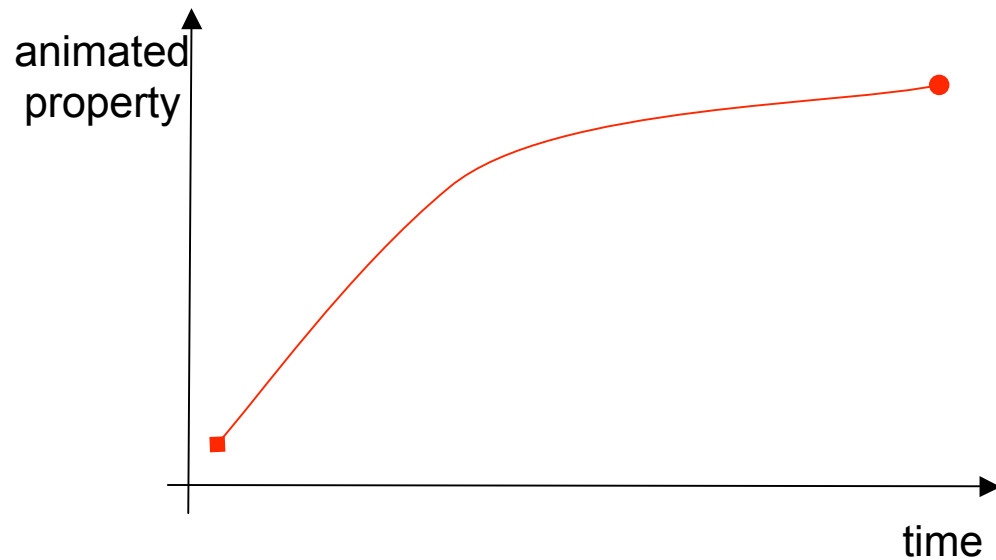
Animations

- Animation functions
 - Discrete / continuous animations
 - Start / intermediate / end values
 - E.g. Linear, ease-in, ease-out, ease-in-out, etc.



- Examples of properties that can be animated:

- position
- size
- z-index
- alpha blending
- Color



Procedural approaches - Do-it-yourself JavaScript

- Use a timed function to update document or CSS properties.

```
function animate (progress) {  
    start = ...; end = ...;  
    elmt.style.property = compValue(start, end, progress);  
}  
var i=0;  
setInterval("animate(i++);",100);
```

- Discussion
 - Ultimately flexible.
 - Works as good as browsers support AJAX programming.
 - Inefficiency of JavaScript.
 - Inefficiency of continued CSS styling and layout updates.

Yahoo UI toolkit: Animate utility.

- See <http://developer.yahoo.com/yui/examples/animation/colors.html>
- What defines an animation:
 - Trigger
 - Animated element and styling property
 - Begin and end state
 - Animation function
- Discussion:
 - Reduce the programming effort, “pseudo declarative”.
 - Otherwise the same as Do-it-by-yourself.

CSS Animations and Transformations in WebKit

- Webkit
 - browser engine of Safari, Mobile Safari, S60, others.
 - see www.webkit.org
- Webkit latest versions include experimental support for CSS Animations and Transformations
 - See <http://webkit.org/blog/138/css-animation/> and <http://webkit.org/blog/130/css-transforms/>

Animating style changes

```
div {  
  color: grey;  
  -webkit-transition-property: color;  
  -webkit-transition-duration: 2s;  
  -webkit-transition-timing-function: linear;  
}
```

```
div:hover {  
  color: red;  
}
```

- `Div:hover` rule is applied while the user designates an element, the rule overwrites the default defined by earlier `div` rule.
- Animate CSS property changes
 - Parameters:
 - properties, their start and end value, duration, choice of pre-defined animation function

Transformation of elements

```
div {  
  -webkit-transform: rotate(90);  
}
```

- Transformation is computed after following CSS layout.
 - Currently 2D transformations: scale, rotate, skew, translate.

- Animating a transformation

```
div {  
  -webkit-transform: rotate(90);  
  -webkit-transition-property: -webkit-transform;  
  -webkit-transition-duration: 2s;  
  -webkit-transition-timing-function: linear;  
}
```

```
div:hover {  
  -webkit-transform: rotate(180);  
}
```

Discussion of CSS Animations and Transformations

- Minimum authoring effort.
- Browser engine extension.
- Maintains compatibility with browsers not supporting it.
- Transformation are useful as such
 - But how far should transformations go? E.g. 3D transformations?
- Future unclear
 - W3C
 - Gecko, Opera, IE support.
- MS IE 5.5 introduced filters,
see <http://msdn2.microsoft.com/en-us/library/ms532847.aspx>

SMIL

- SMIL - Synchronized Multimedia Integration Language.
- Pronounced *smile*.

- A W3C Specification
- Latest ready version: SMIL 2.1 Recommendation of 2005.
- Work on SMIL 3 close to completion, in Candidate Recommendation state
- SMIL 1.0 from 1998.

- SMIL Purpose:
 - Simple authoring of interactive, multimedia presentations,
 - Rich multimedia presentations that integrate audio-visual content with text, images, graphics, etc.
 - Use-cases beyond the Web:
 - Messaging, multimedia content delivery, etc.
 - Many users perceive a SMIL presentation as an "interactive video", they do not perceive it as a "document" or "application".

SMIL Structure

- Basic Structure of a SMIL presentation:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!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">
  <head>
    <layout>
      ..
    </layout>
    <transition .../>
  </head>
  <body >
    ...
  </body>
</smil>
```

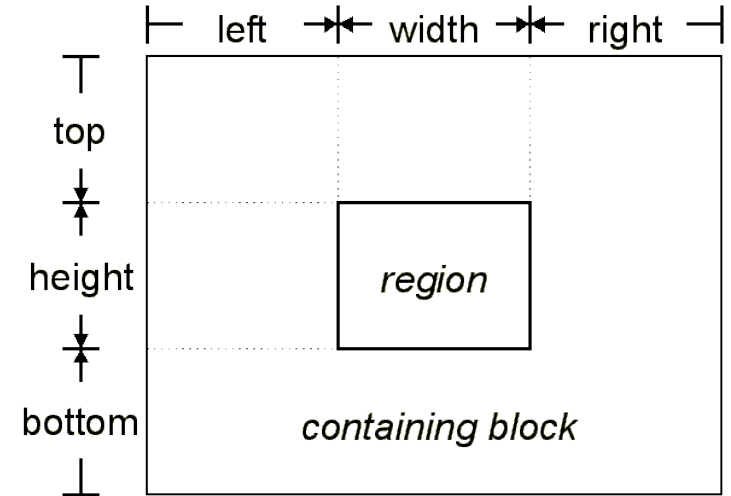
SMIL Media Integration

- All media content must be contained in separate files, a SMIL presentation refers to these files
- Note handling of text:
 - A SMIL presentation may refer to an HTML file. The HTML content will be laid out within the specified rectangular area, called a “region”.
- The SMIL specification does not require support for any specific media formats.
- The following elements are synonyms:
 - <ref>, <text>, , <video>, <audio>, <animation>, <textstream>
- Example

```
<text id="foo" src="foo.html" alt="foo bar" />
```

SMIL Layout

- SMIL layout is the same as CSS fixed position layout, but uses an XML syntax, in the document head.
 - SMIL defines layout of regions in 3-dimensional space.
 - Position of a region within the root layout or containing region.
 - units: percent or points.
 - All media elements are shown in a region.
 - The media object refers to the region in which it should be rendered.



```
<smil xmlns="...">  
  <head>  
    <layout>  
      <root-layout width="320" height="480" />  
      <region id="a" top="5" right="10%" width="80%" />  
    </layout>  
  </head>  
  <body>  
    <text region="a" src="text.html" ... />  
  </body>  
</smil>
```

size of the view port

selector mechanism in SMIL

SMIL Timing

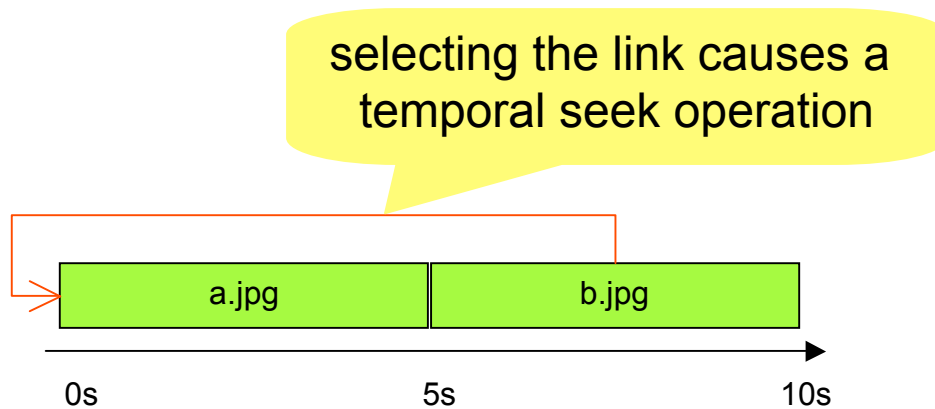
- Defines temporal synchronization of elements.
- Timing attributes
 - begin
 - defines when the element starts
 - end
 - defines when the element ends
 - dur
 - defines the duration of an element.
 - the default duration is the media object's intrinsic duration or indefinite when it does not have an intrinsic duration.
- synchronization options
 - relative to the parent element - example: `begin="5s"`
 - relative to a specific object - example: `begin="otherObject.end+5s"`
 - on an event - example: `begin="object.activateEvent"`

SMIL Timing - time containers

- The <seq> element plays the child elements one after another in a sequence.
 - Children of a seq begin by default when the previous child ends its active duration (equivalent to begin="0s"); the first child begins by default when the parent seq begins
- The <par> element plays child elements as a group (allowing "parallel" playback).
 - Children of a par begin by default when the par begins (equivalent to begin="0s").
 - A par ends when all its children have ended, unless its duration is explicitly specified.
- The <excl> element plays one child at a time, but does not impose any order.
 - Children of an excl default to a begin value of "indefinite".
 - Activation of one child automatically de-activates any other active child object.

Interaction – Temporal Linking - Example

```
<seq>  
    
  <a href="#a">  
      
  </a>  
</seq>
```



SMIL / SVG - Animations

- SMIL's animation primitives allow the value of attributes to be changed during the active duration of their associated elements.
- Examples of attributes that can be changed
 - position, size, opacity level, sound level, background color .

```
<region id="M" top="10" left="10" height="218" width="280"  
      backgroundColor="red" />
```

...

```
<animateMotion targetElement="M" attributeName="top" from="10" to="50"  
  by="2" dur="5s" />
```

```
<animateColor targetElement="M" attributeName="backgroundColor" dur="10s"  
  calcMode="discrete" values="green; yellow; red" />
```

- SplineAnimations define a spline animation function for computing the animated attribute value.

Discussion on SMIL

- Pro
 - Provides sophisticated multimedia synchronization possibilities.
 - Concept done right.
 - W3C specs exists.
- Cons:
 - More functionality than most people ever need.
 - Steep learning curve for developers.
 - Complexity of the spec.
 - Complexity of SMIL documents.
 - Does not integrate as such to HTML, CSS, JavaScript
 - Scripting SMIL likely to lead to unmanageable complexity.
 - Relies on fixed position layout model.
 - Lack of implementation support.

Timesheets by Prof Vuorimaa

- See separate slides

Conclusions

- Choice of declarative or procedural - it depends on the case.
- JavaScript Frameworks, like YUI Animate, seem a good compromise between development effort and flexibility, especially if the Web app uses scripting anyhow.
- Future of CSS Animations and Transformations unclear.
 - Apple team promised a scripting interface.
 - Transformation provide a benefit of its own.
- Any syntax extension is a hard sell
 - Browser support
 - Old documents on the Web must work with updated browsers.
 - New documents must interoperate with old browsers.
 - CSS Animations seem more realistic than Timesheets, work with legacy browsers.