

SMIL

(Synchronized Multimedia Integration Language)



Francisco José Suárez Alonso

UNIVERSIDAD DE OVIEDO

Área de Arquitectura y Tecnología de Computadores

Curso 2010/2011

Introducción





- **Objetivo: Multimedia para la web**
 - Es para multimedia lo que HTML para hipertexto
 - Formato de integración para presentar medios
 - Temporización y sincronización
- **Sintaxis**
 - SMIL es un meta-lenguaje basado en XML
 - Organizado por "perfiles" (subconjuntos)



Referencias

A ficheros externos

Internas

```
<region id="caption" />
<body>
  <text dur="3s" src="bienvenido.txt" region="caption" />
</body>
```

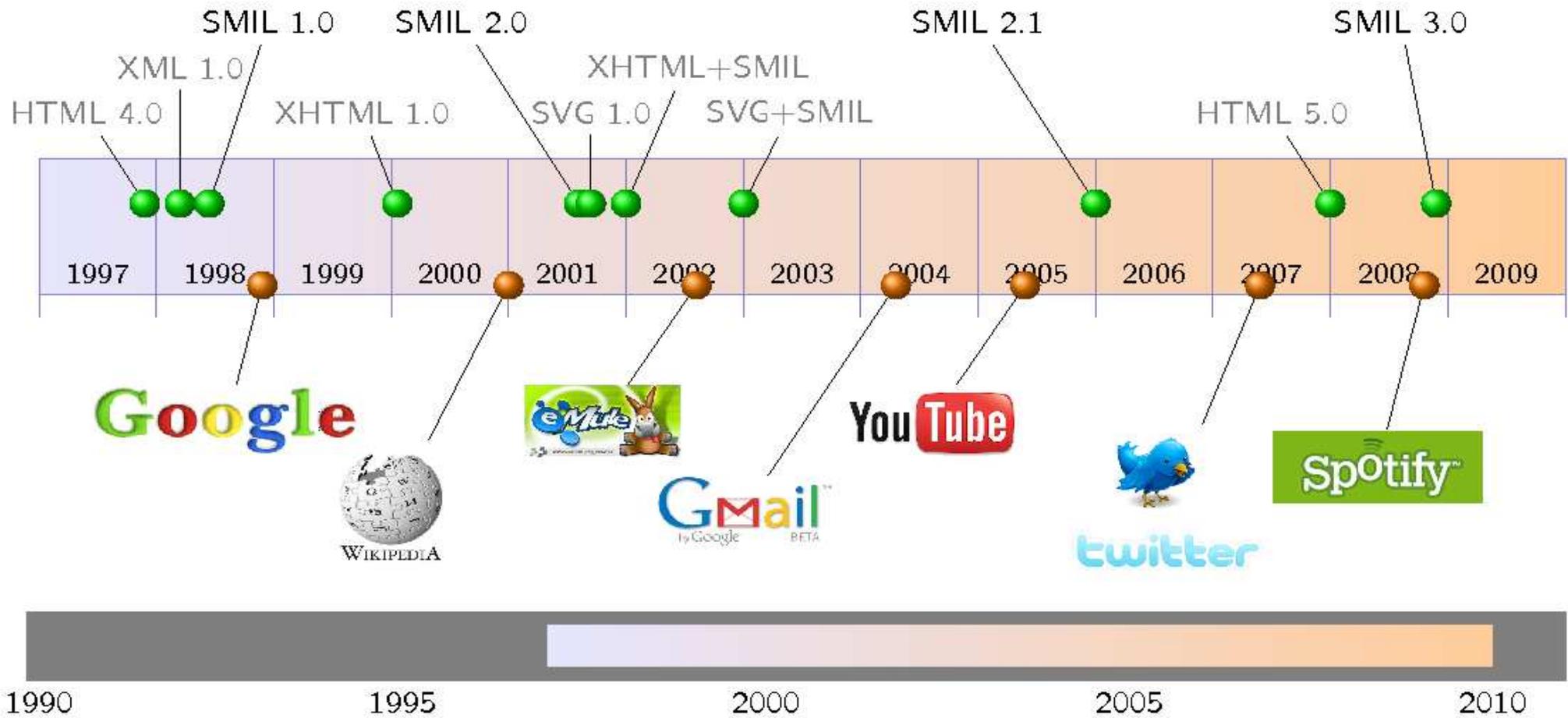
Elementos

Atributos

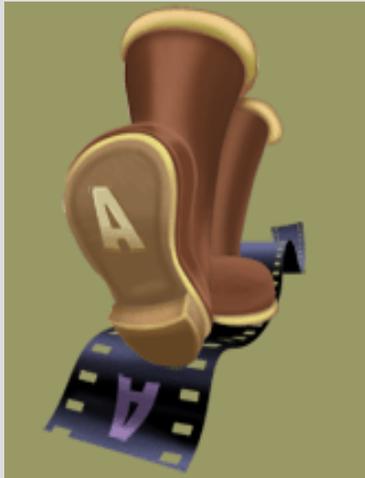


- Definido con DTDs y Schema de XML:
<http://www.w3.org/2005/SMIL21/SMIL21.dtd>
- Elementos de medios referenciados, no incluidos
- Puede ser extendido a medida
- Integrable en entornos XML
- Disposición espacial replicable con CSS
- Comparte construcciones con (X)HTML

SMIL: Versiones



Implementaciones de SMIL



Ambulant

Código abierto
Soporte completo
Único con soporte
SMIL 3.0



Real Player

Líder en
reproductores SMIL
Soporte SMIL 2.0
Language y SMIL 2.1



Apple Quicktime

Soporta SMIL 1.0



Adobe SVG Viewer

Soporta SVG+SMIL



▪ **Lenguaje de Integración de Multimedia Sincronizada**

Utilizando SMIL, un autor puede:



- describir la disposición espacial de la presentación en una pantalla



- asociar hiperenlaces con objetos de medios



- describir el comportamiento temporal de una presentación multimedia

- La sintaxis SMIL 2.1 se puede reutilizar en otros lenguajes XML (p.e. SVG y XHTML)



- La sintaxis de SMIL 2.1 se "reparte" en 50 módulos, organizados en 10 grupos

... Animation, Content Control,
Layout, Linking, Media Object,
Metainformation, Structure,
Timing and Synchronization,
Time Manipulations, y Transition
Effects, ...

- El grupo *Timing and Synchronization* es el núcleo de la especificación
- Lenguajes basados en XML (p.ej: SVG) pueden soportar sólo algunos módulos



- Un perfil proporciona una colección fija de elementos y atributos, tomados de uno o más módulos
- Es el nivel de la especificación usado por el diseñador
- Algunos perfiles importantes:
 - SMIL 2.1 Language Profile
 - SMIL 2.1 Mobile Profile
 - SMIL 2.1 Extended Mobile Profile
 - XHTML+SMIL Profile (basado en SMIL 2.0)
 - SMIL 3.0 Tiny Profile

¿Qué especifica SMIL?



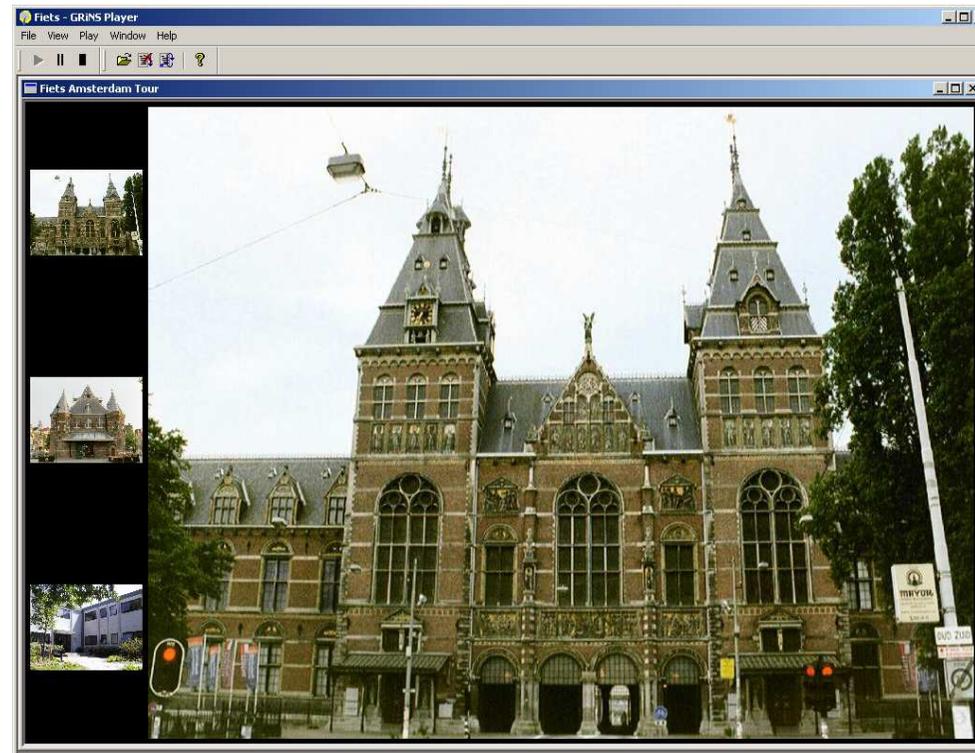
MEDIOS



ADAPTABILIDAD



DISPOSICIÓN



INTERACCIÓN



TEMPORIZACIÓN



¿Qué especifica SMIL?



MEDIOS



Texto, audio, vídeo, imágenes, animaciones

DISPOSICIÓN



Distribución espacial de los elementos de la presentación

ADAPTABILIDAD



A características del usuario
A distintos anchos de banda
A meta-datos

INTERACCIÓN



Posibilidad de navegar por los contenidos multimedia

TEMPORIZACIÓN



Distribución temporal de los elementos de la presentación. Sincronización de los múltiples medios



Soluciones propietarias de RealNetworks

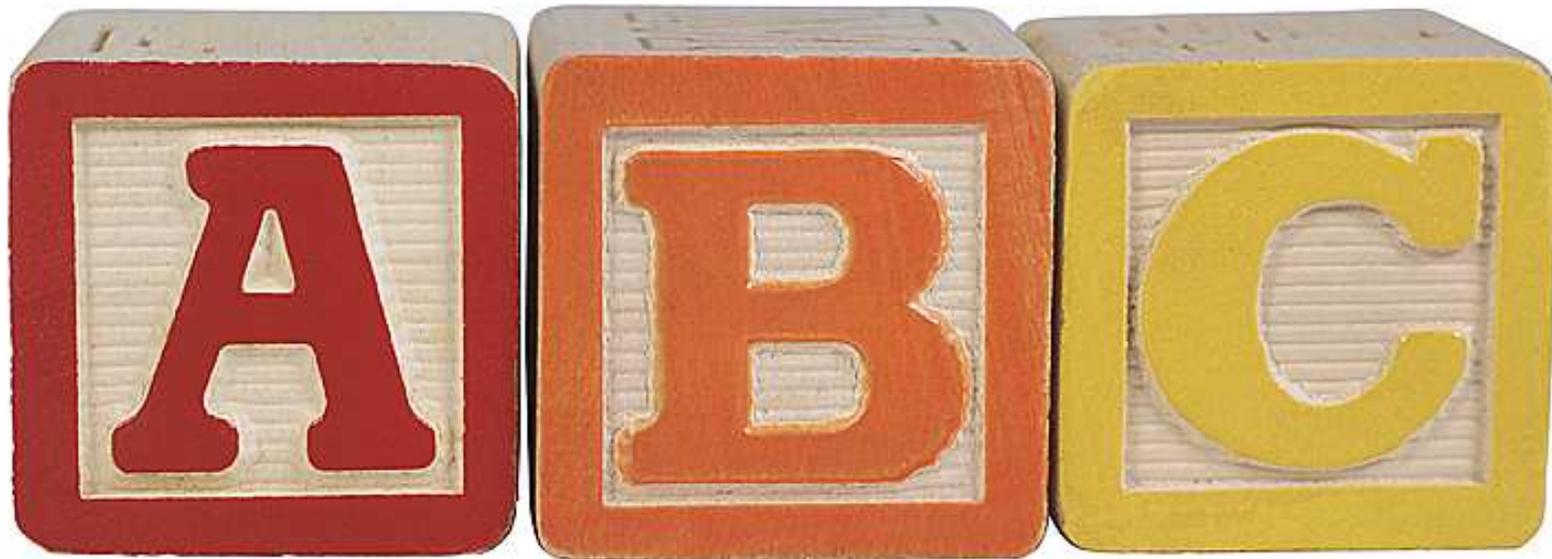
- Además de RealAudio/RealVideo...
- RealPix: formato gráfico
- RealText: formato de representación de texto
- Herramientas de autoría: RealPresenter, RealSlideShow

En qué nos centraremos



- Estudio general de SMIL
- SMIL 2.0 Language Profile
 - Sobre RealPlayer
 - Con algunas extensiones propietarias

Conceptos básicos



■ Clases de Documentos SMIL

```
<!DOCTYPE smil PUBLIC "-//W3C//DTD SMIL 2.1//EN"
    "http://www.w3.org/2005/SMIL21/SMIL21.dtd">
<smil xmlns="http://www.w3.org/2005/SMIL21/Language">
```

SMIL 2.1
Language Profile

```
<!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">
```

SMIL 2.0
Language Profile

```
<!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">
```

SMIL 2.0
XHTML+SMIL profile



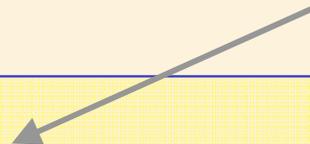
- Elemento `<smil>`
 - Contiene todos los elementos de un doc SMIL
- Elemento `<head>`
 - Disposición y meta-información como en HTML
- Elemento `<body>`
 - Contenido y enlaces (como en HTML)
+ temporización

Ejemplo 1



```
<smil>
<head>
  <meta name="title" content=" MI PRIMER SMIL " />
  <meta name="generator" content="notepad" />
  <meta name="author" content="Nacho" />
  <meta name="copyright" content="(c) 2004 Nacho" />
  <layout>
    <root-layout title=" MI PRIMER LAYOUT " id="layout1"
      width="1000" height="1000" />
  </layout>
</head>
<body>
  
  <text src="ejemplo1/saludo.txt" dur="5s" />
  <video src="ejemplo1/videotest.rm" />
</body>
</smil>
```

Contenido





- URI: Uniform Resource Identifier
 - Ejemplo:
`http://www.uniovi.es/SMIL/presentacion.smi`
- Atributo `src=`
 - Almacena una URI para acceder a un medio
- Path del sistema local de ficheros
 - Relativos y absolutos
- `xml:base=`
 - Atributo que define la base para URIs relativas
 - Árbol de herencias descendente

- `<ref>`, `<text>`, `<textstream>`, ``,
`<audio>`, `<video>` y `<animation>`

`<ref src="cualquiercosa.???" ... />`

`<text src="texto.html" ... />`

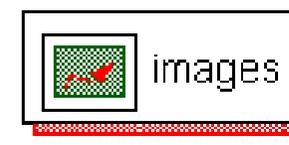
`<textstream src="creditos.rtx" ... />`

``

`<audio src="voz.wav" ... />`

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

`<animation src="rotulos.anim" ... />`



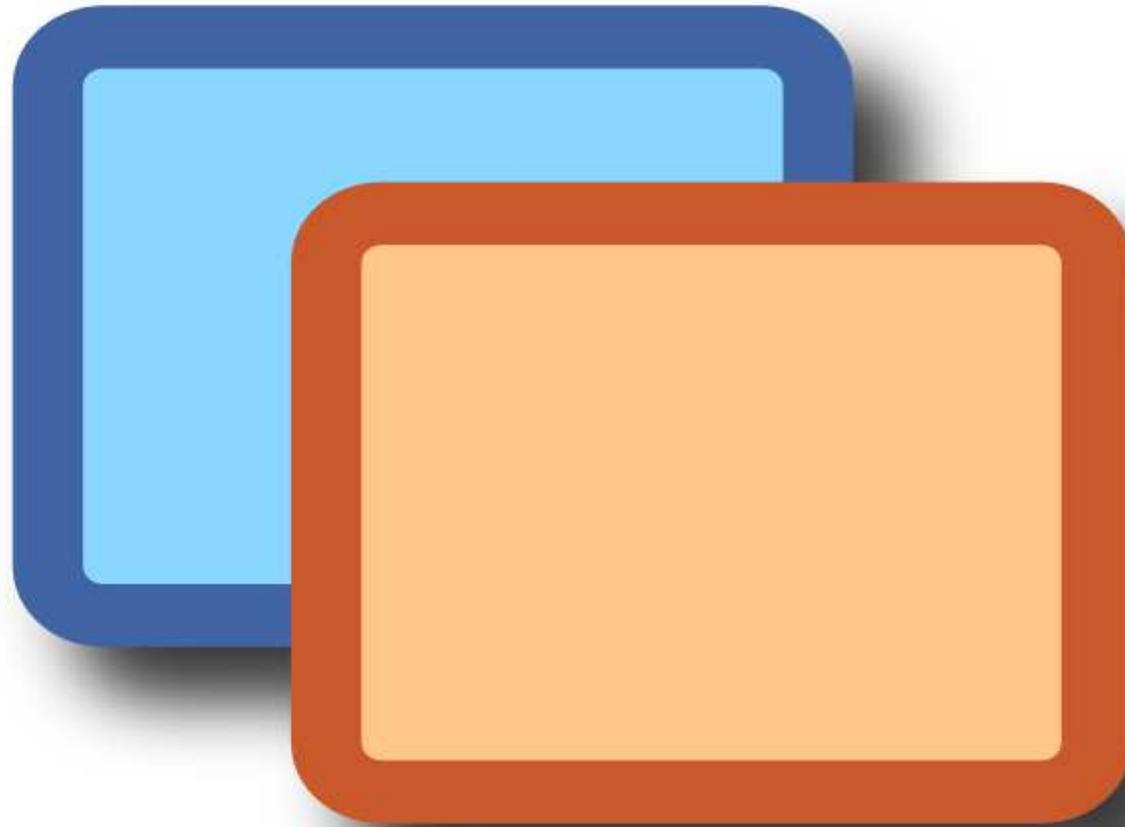


- Tipos generales
 - Imágenes (JPEG, GIF, RealPix, PNG, BMP,...)
 - Video (MPEG, RealVideo, QuickTime, ...)
 - Audio (WAV, RealAudio, MP3, ...)
 - Animaciones (GIF, SVG, Flash, ...)
 - Texto (HTML, texto plano, ...)
- Tipado de medios
 - Construcción mimetype (Atributo **type=**)
 - Extensión de archivo en caso de omisión del tipo



- Atributo **src=**
URI que apunta al medio
- Atributo **type=**
Tipo (MIME) del medio
- Atributos **clipBegin=** y **clipend=**
Instante de comienzo y fin de reproducción
- Atributo **color=**
Color de fondo para el medio

Disposición espacial



Disposición espacial en SMIL 2.0



Cuatro módulos:

- **BasicLayout:**

La disposición se controla mediante los elementos `<layout>` `<root-layout>` y `<region>`

- **AudioLayout:**

define nivel de salida relativo con el atributo `soundLevel=` para el elemento `<region>`

- **MultiWindowLayout:**

extiende `BasicLayout` para definir múltiples ventanas raíz

- **HierarchicalLayout:**

extiende `BasicLayout` y `MultiWindowLayout` para definir solapamientos de disposiciones, regiones de forma dinámica, etc.



- Ventana del reproductor
 - Contiene los elementos que el reproductor quiera colocar (botones, etc...)
 - Y una zona rectangular que es el `<root-layout>` (ventana de presentación)
 - SMIL puede definir regiones dentro del `<root-layout>`
- Regiones
 - Son divisiones rectangulares del `<root-layout>`
 - No se pueden anidar
 - A cada una se le asigna un nombre (atributo `id=`)
 - Y una posición y tamaño (atributos)
 - Los medios especifican en qué región aparecerán (atributo `region=`)

Ejemplo



```
<smil>
<head>
  <!--Definicion de metadatos -->
  <layout>
    <root-layout background-color="blue"
      width="250" height="230"/>
    <region id="video" top="5" left="5" width="240"
      height="180" background-color="yellow"/>
  </layout>
</head>
<body>
  <video src="Ejemplo/pelicula.rm" region="video" />
</body>
</smil>
```

Definición del layout

Ventana de presentación

Contiene una región

El medio especifica en qué región debe salir

Layouts complejos. Modelo de cajas



Regiones

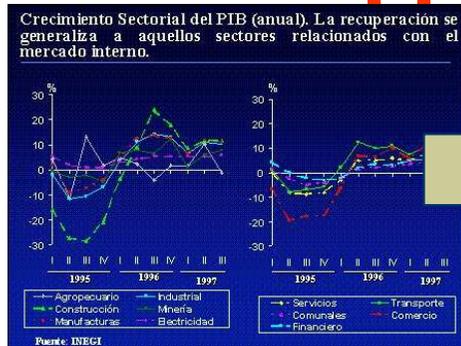
INFORME DE PROGRESO

ECONÓMICO

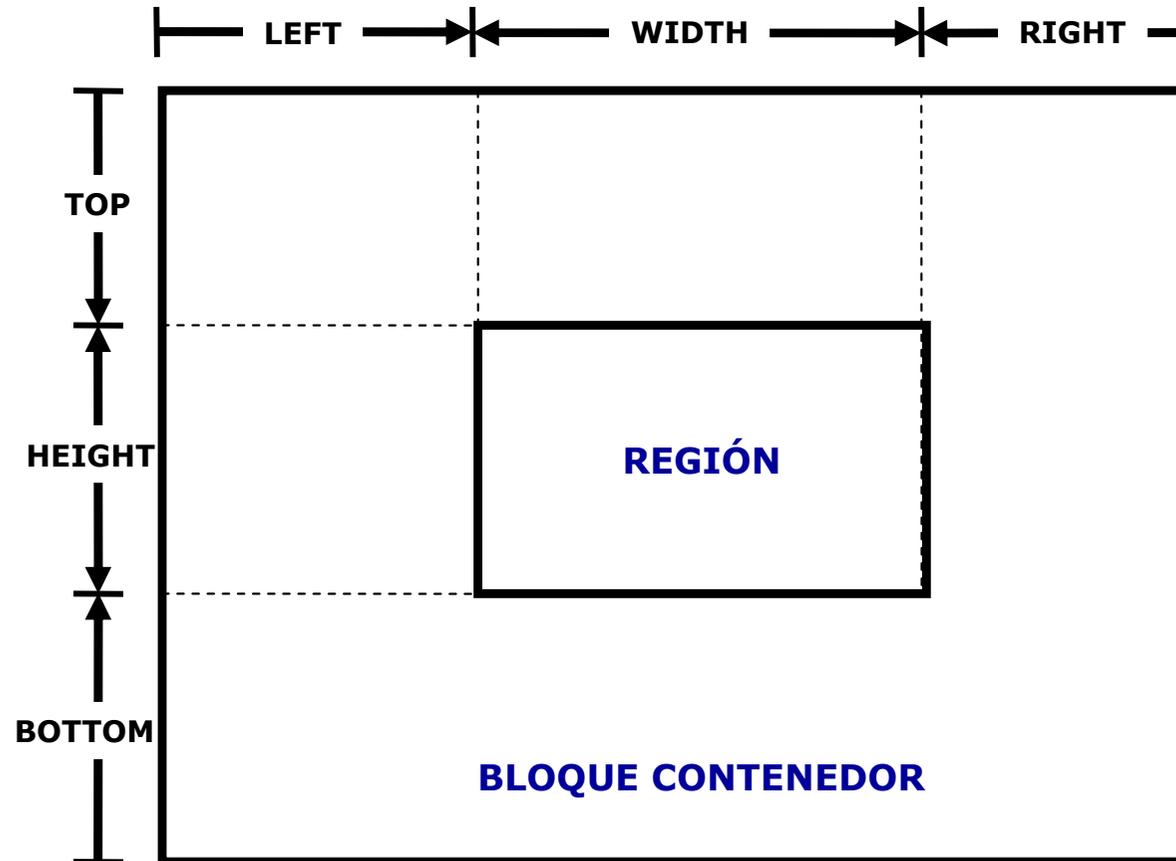
según un informe del
Ministerio de Economía



Joe Perkins / Lolo Gómez



Modelo de cajas



ATRIBUTOS DE POSICIONAMIENTO DE REGIÓN

Cajas contenedoras: alineamiento



- Sub-regiones: Posicionamiento preciso de imagen dentro de regiones

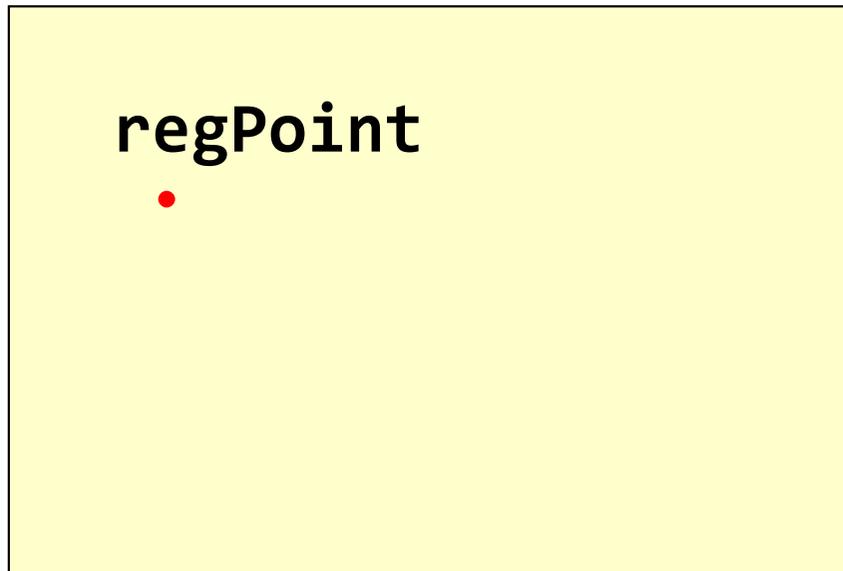
- Alineamiento:

regPoint= Punto de referencia en el layout

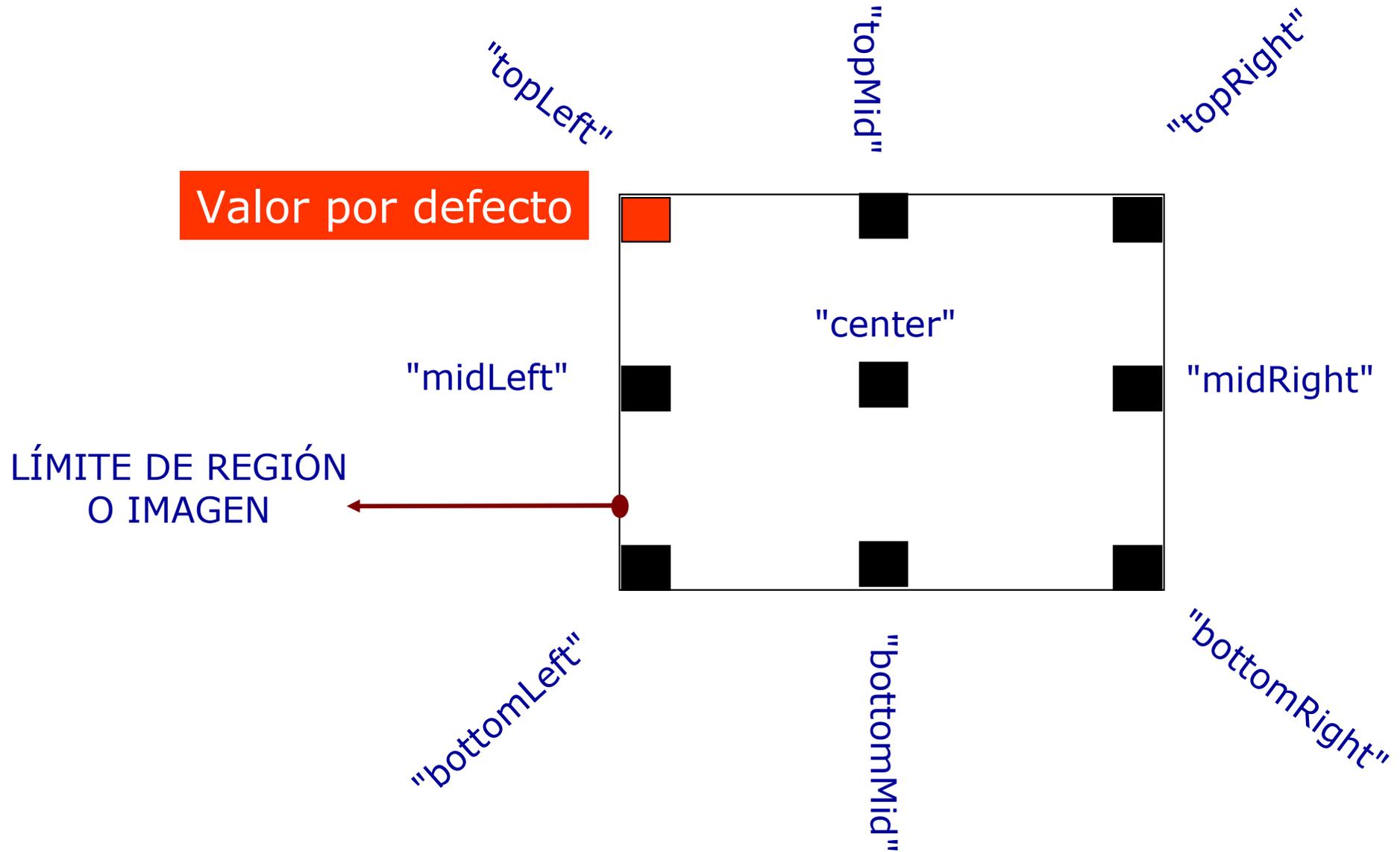
Es un elemento del layout

regAlign= Punto del objeto que se alineará con regPoint

Es un atributo del objeto



Alineamiento. Valores de regAlign y regPoint

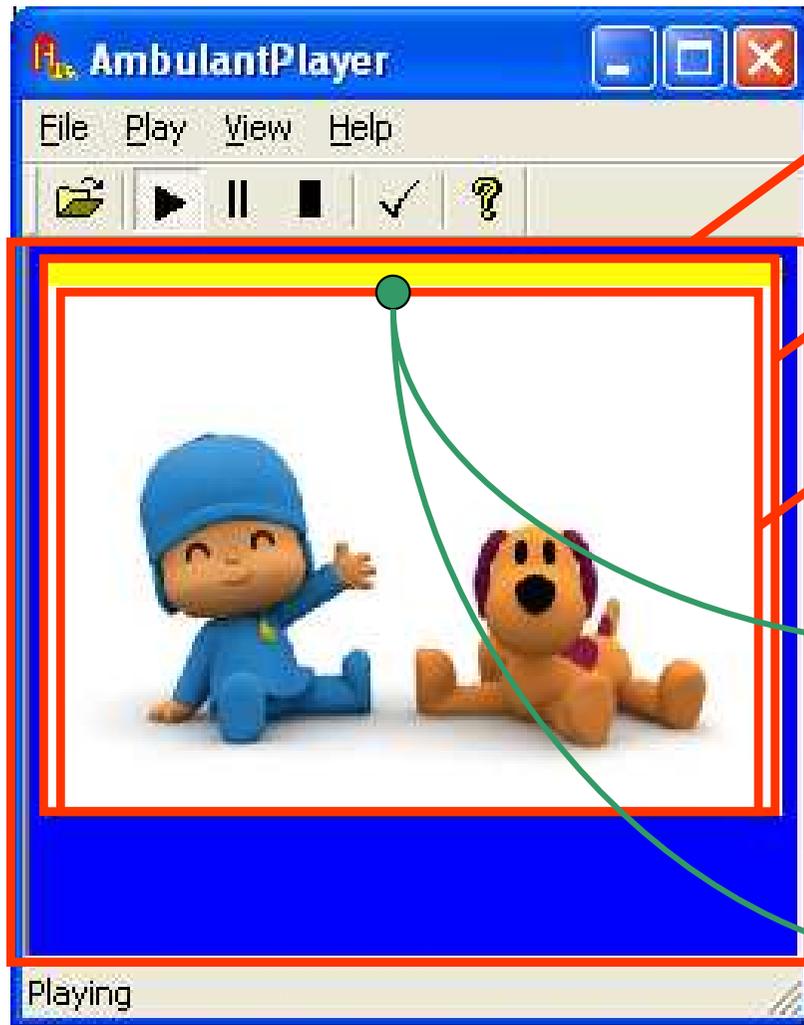


Creando nuevos valores de regPoint/regAlign

El elemento `<regpoint>` permite definir nuevos nombres, tipo "topLeft", para utilizar en los atributos `regPoint=` ó `regAlign=`

```
<smil><head>
  <layout>
    <root-layout background-color="blue"
      width="250" height="230"/>
    <region id="foto" top="5" left="5" width="240"
      height="180" background-color="yellow"/>
    <regPoint id="casiArriba" top="5%" left="50%" />
  </layout></head>
<body>
  
</body>
</smil>
```

Resultado



root-layout (azul)

region "foto" (amarillo)

regPoint "casiArriba"
colocado a 5px del borde superior de "foto" y al 50% de su ancho

regAlign "topMid"
colocado en el centro del borde superior del

Cajas contenedoras: alineamiento



Ubicación del Clip	Valores de puntos de registro	Ejemplo
Arriba Izquierda (por defecto)	<code>left="0%" top="0%" regAlign="topLeft"</code>	
Arriba Centro	<code>left="50%" top="0%" regAlign="topMid"</code>	
Arriba Derecha	<code>left="100%" top="0%" regAlign="topRight"</code>	
Mitad Izquierda	<code>left="0%" top="50%" regAlign="midLeft"</code>	
Centro	<code>left="50%" top="50%" regAlign="center"</code>	
Mitad Derecha	<code>left="100%" top="50%" regAlign="midRight"</code>	
Abajo Izquierda	<code>left="0%" top="100%" regAlign="bottomLeft"</code>	
Abajo Centro	<code>left="50%" top="100%" regAlign="bottomMid"</code>	
Abajo Derecha	<code>left="100%" top="100%" regAlign="bottomRight"</code>	
Cuadrante Superior Izquierdo	<code>left="50%" top="50%" regAlign="bottomRight"</code>	
Cuadrante Superior Derecho	<code>left="50%" top="50%" regAlign="bottomLeft"</code>	
Cuadrante Inferior Izquierdo	<code>left="50%" top="50%" regAlign="topRight"</code>	
Cuadrante Inferior Derecho	<code>left="50%" top="50%" regAlign="topLeft"</code>	

Escalado de objetos en regiones



- ¿Qué hacer si el tamaño del objeto no coincide con el de la región?
 - El atributo `fit=` en la región lo especifica
 - La interacción de `fit=` con `regAlign=` y `regPoint` puede ser compleja

Escalado en regiones (fit=)



"hidden" (por defecto)
OBJETO NO ESCALADO



"hidden" (por defecto)
OBJETO NO ESCALADO



"scroll"
OBJETO NO ESCALADO



"meet"
RAZÓN DE ASPECTO MANTENIDA



"slice"
RAZÓN DE ASPECTO MANTENIDA



"fill"
RAZÓN DE ASPECTO NO MANTENIDA

Ejemplo



```
<smil>
<head>
  <meta name="title" content=" MI SEGUNDO SMIL  "/>
  <layout>
    <root-layout background-color="blue"
      width="250" height="230"/>
    <region id="foto"
      top="5" left="5" width="240" height="180"
      background-color="yellow" fit="hidden"/>
  </layout>
</head>
<body>
  
</body>
</smil>
```

Al no especificar `regAlign`, la imagen alineará su "topLeft" con el "topLeft" de la región

Regiones solapadas (z-index)



z-index determina el orden de apilado

- El entero más alto supone el tope de la pila
- Por defecto, el elemento iniciado más recientemente aparece en el tope de la pila

```
<smil><head>
  <layout>
    <root-layout background-color="blue" height="400" width="400" />
    <region id="R1" top="50" left="50" height="300" width="300" z-index="0" />
    <region id="R2" top="125" left="50" height="150" width="300" z-index="2" />
    <region id="R3" top="100" left="75" height="200" width="250" z-index="1" />
    <region id="R4" top="125" left="125" height="150" width="150" z-index="3" />
    <region id="R5" top="50" left="140" height="300" width="120" z-index="4" />
  </layout></head>
<body>
  
  . . .
  
</body>
</smil>
```

Temporización



Duración de un medio



Intrínseca

Derivada del contenido del medio

```
<video src="v.mpg" region="rV" />
```

1

Intrínseca medios discretos (texto, imagen) = cero

Explícita

Atributo **dur=** valor de reloj o “**indefinite**”

Puede ser menor que su duración intrínseca

```
<video src="v.mpg" region="rV" dur="4s" />
```

2

O mayor

```
<video src="v.mpg" region="rV" dur="6.5s" />
```

3

1



Audio (o vídeo) dura 5.3 segundos

2



El medio para después de 4 segundos

3



El medio para después de 6.5 segundos

El audio se detiene a los 5.3 y se mantiene visible el último frame del vídeo

Más parámetros de temporización



begin=

instante de comienzo

end=

instante de fin

dur=

(ya visto) duración explícita

repeatDur=

iterar durante un tiempo especificado

repeatCount=

iterar un n^o de veces especificado

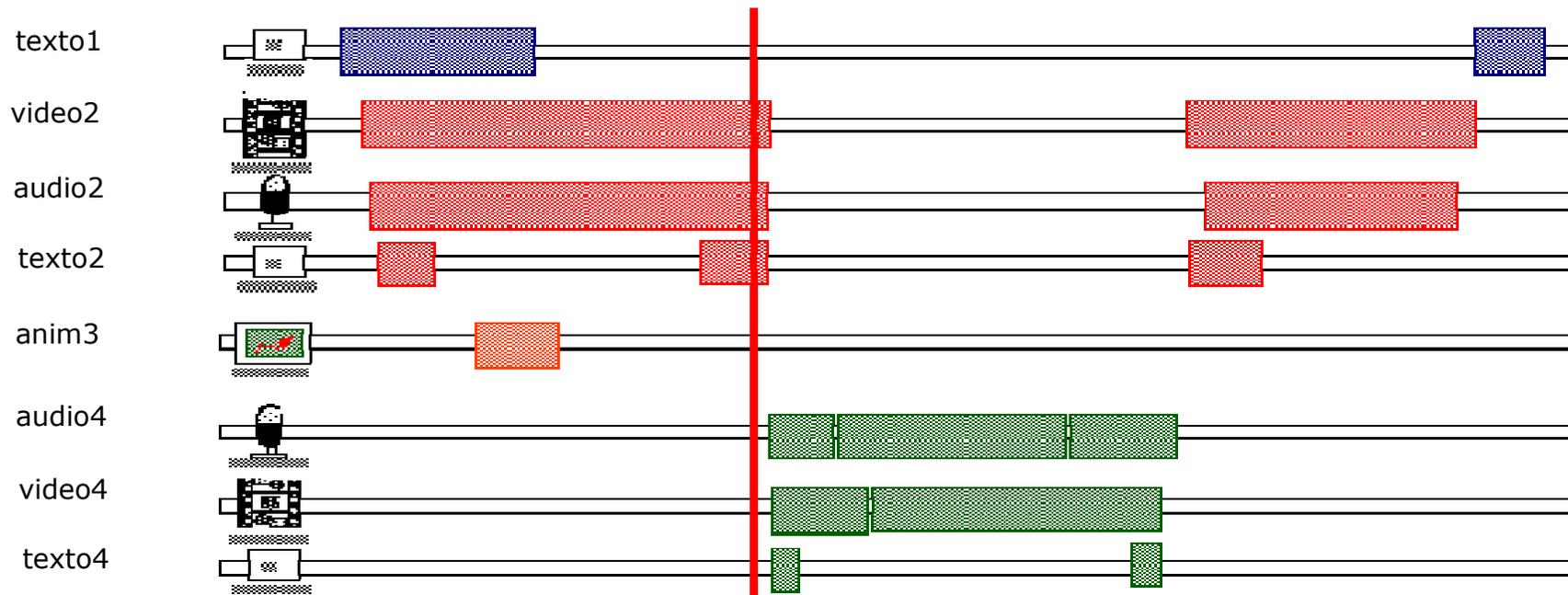


- Los siguientes elementos pueden contener medios e indican cuándo se reproducirían
 - `<body>`, `<seq>` antes o después que sus hermanos
 - `<par>` a la vez que sus hermanos
 - `<switch>`, `<excl>`, `<priorityClass>` en lugar de sus hermanos
 - `<a>` en respuesta a una interacción

Secuenciación



Se pueden crear secuenciaciones complejas anidando los elementos anteriores



Ejemplo



```
<smil><head>
  <layout>
    <root-layout width="380" height="300" />
    <region id="R1" top="10" left="40" width="320" height="256" />
    <region id="R2" bottom="5" right="20" width="320" height="18" />
  </layout></head>
```

```
<body>
```

```
<par>
```

Los elementos dentro de <par> evolucionan simultáneamente

1

```
<video src="ejemplo2/BTT1.rm" region="R1" />
<text src="ejemplo6/saludo.txt" dur="7s" region="R2"
      regAlign="center" regPoint="center" />
```

```
</par>
```

```
<par>
```

2

```
<video src="ejemplo6/22 BTT2.rm" region="R1" />
<text src="ejemplo6/seguimos.txt" dur="7s" region="R2"
      regAlign="center" regPoint="center" />
```

```
</par>
```

```
</body></smil>
```

Los elementos dentro de <body> se muestran en secuencia

Secuenciación ¿Cuándo acaba un bloque <par>



```
<par>
```

```
1 <text src="titulo.html" region="m_titulo" dur="5s"/>
```

```
2 <video id="v1" src="noticia.mpg" region="V-Main" begin="1.4s"/>
```

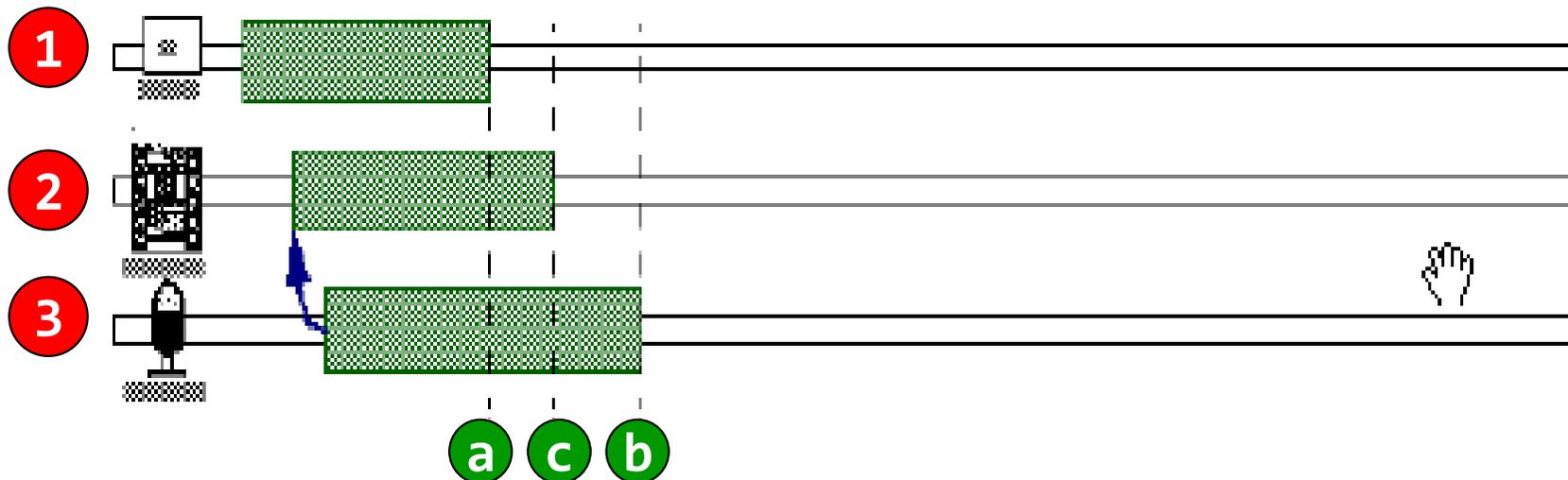
```
3 <audio src="noticia.aiff" region="music" begin="id(v1)(0.5s)"/>
```

```
</par>
```

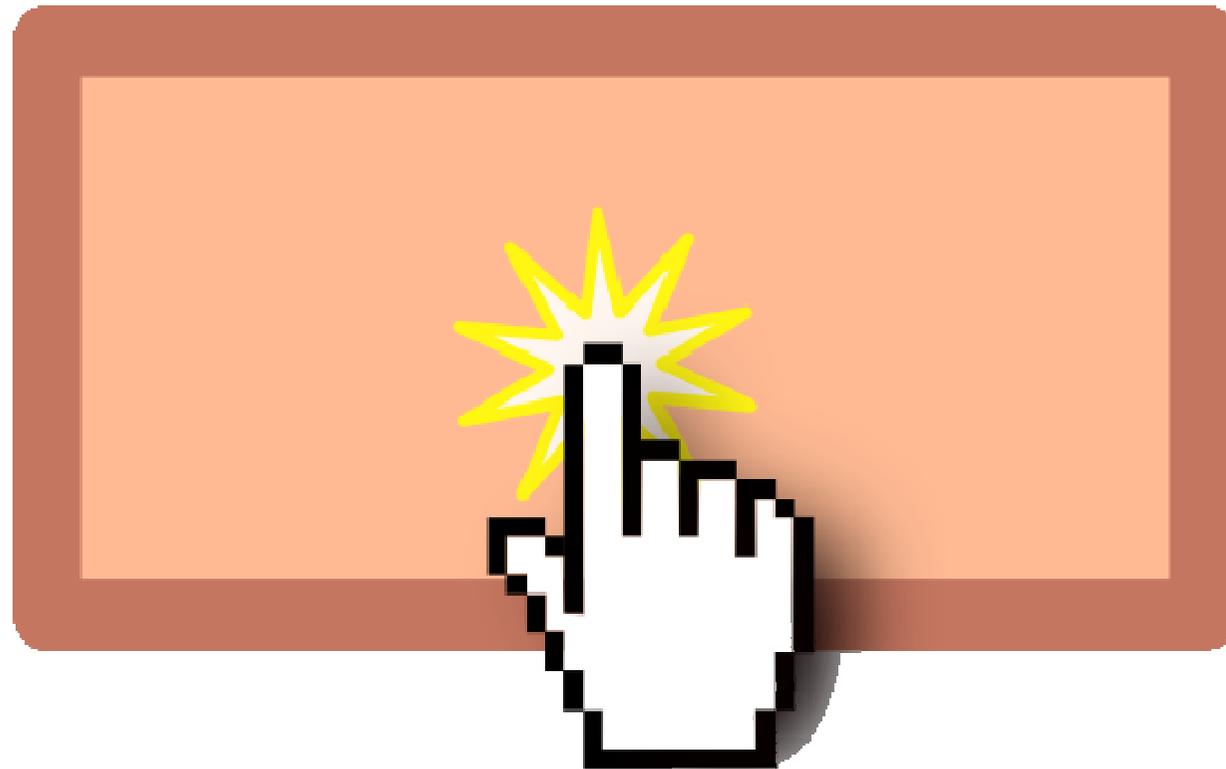
El elemento <par> puede llevar un atributo **endsync=**

- a "first" Cuando acabe el primero
- b "last" Cuando acabe el último (valor por defecto)
- c "id(valorID)" Cuando acabe el elemento con el **id** dado

Ej: "id(v1)"



Interacción



Interacción: enlaces



- Elemento **<a>**
 - Contiene referencias a medios cuya reproducción es disparada por un enlace (como en HTML)
- Atributo **href=**
 - Valor URI que indica el destino del enlace
- Elemento **<area>**
 - Define un área "clicable" y una dirección de enlace
- Enlazado con otra parte de la presentación
 - Asignación del atributo **href="#idref"**
 - Causa movimiento en la línea de tiempo

Ejemplo



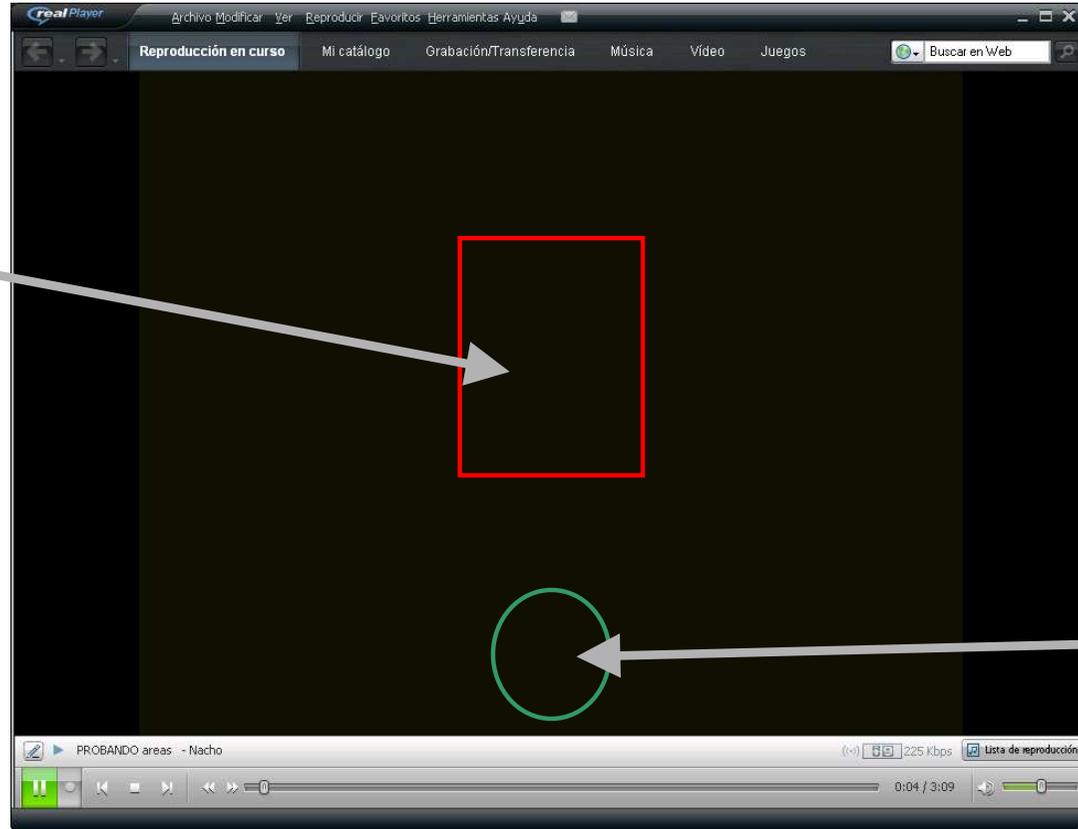
```
<smil>
<head>
  <meta name="title" content="  PROBANDO areas  " />
</head>
<body>
  El área puede tener diferentes formas
  <video src="ejemplos/video.rm">
    <area id="a1" href="ejemplos/Texto1.txt"
      shape="rect" coords="110,10,180,100" />
    <area id="a2" href="ejemplos/Texto2.txt"
      shape="circle" coords="160,177,20"
      begin="10s" dur="5s" />
  </video>
</body>
</smil>
```

Y atributos temporales (aparece/desaparece)

Resultado



Lleva a Texto1



Lleva a Texto2

(En realidad no son visibles)

Moverse en el tiempo

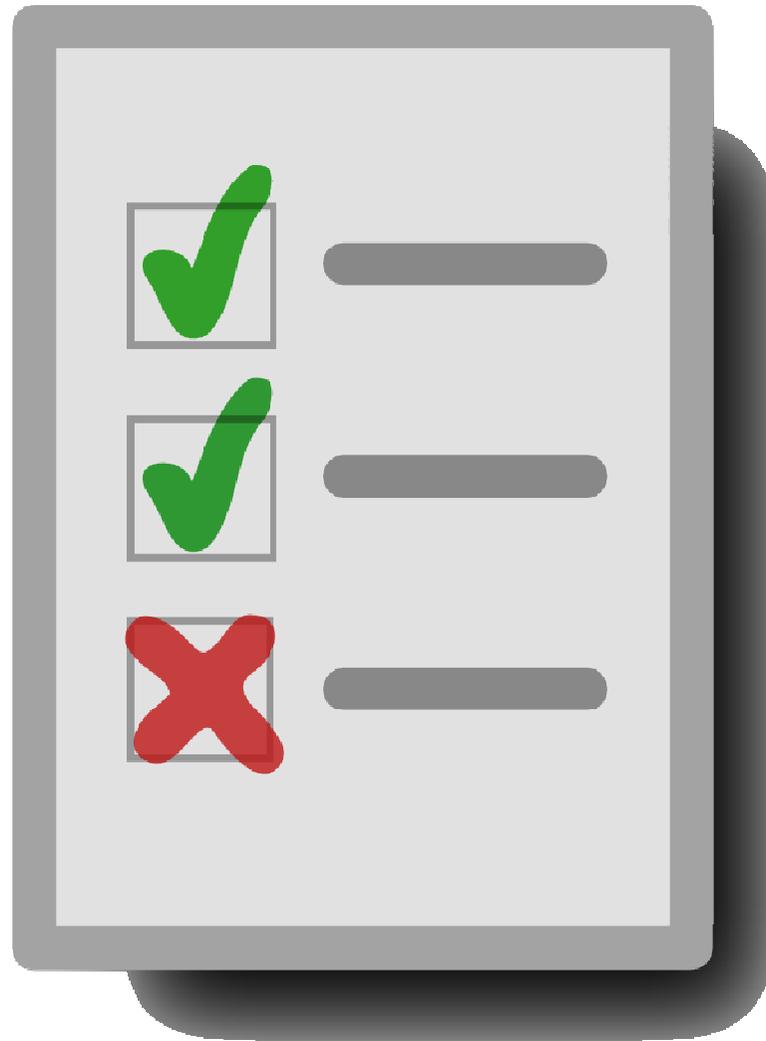


El destino de un enlace puede causar un salto en el tiempo

```
<smil><head>
<layout>
  <region id="R1" left="0%" top="0%" right="0%" bottom="0%" />
</layout></head>
<body>
  <video src="ejemplos/pelicula.rm" region="R1" >
    <area id="S1" href="#S2" shape="rect"
      coords="0%,0%,100%,100%" begin="0s" dur="10s" />
    <area id="S2" href="#S3" shape="rect"
      coords="0%,0%,100%,100%" begin="38s" dur="10s" />
    <area id="S3" href="#S1" shape="rect"
      coords="0%,0%,100%,100%" begin="0:02:45" dur="10s" />
  </video>
</body>
</smil>
```

El area S1 lleva al área S2, por tanto al instante 38s

Selectividad



Principios



- En cada elemento se puede indicar una prueba de “reproducibilidad”
- Aquellos elementos no seleccionados (no pasan la prueba) se ignoran y no son reproducidos
- Atributos de prueba del sistema:
 - Colección estándar de condiciones a comprobar
 - Pueden ser combinados

Atributos de prueba del sistema



Adaptación al usuario

```
systemLanguage= ("es", "en", etc.)  
systemCaptions= ("on", "off")  
systemOverdubOrSubtitle= ("overdub", "subtitle")  
systemAudioDesc= ("on", "off")
```

Adaptación al hardware

```
systemBitrate= ("128000", "256000", "512000", ...)  
systemCPU= ("alpha", "x86", "ppc", "mips", "arm", ...)  
systemScreenSize= ("800x600", "1024x768", ...)  
systemScreenDepth= ("1", "4", "8", "16", "32")
```

Adaptación al software

```
systemOperatingSystem= ("winnt", "wince", "win9x",  
"win32", "win16", "unix", "qnx", "linux", "solaris", ...)
```

Ejemplos de selectividad



```
<par>
  <video src="peli.rm">
  <text systemCaptions="on" src="subtitulos.rt" />
</par>
```

Reproduce un vídeo y muestra subtítulos en caso de estar activada la opción correspondiente en el reproductor

```
<par>
  <video src="peli.rm">
  <par systemCaption="on">
    <audio systemOverdubOrSubtitle="overdub" src="doblaje.mp3" />
    <textstream systemOverdubOrSubtitle="subtitle" src="subtit.rt" />
  </par>
</par>
```

Comprueba si se prefiere la opción de subtítulos o la de versión doblada

Selectividad: <switch>



- El elemento <switch> contiene varios elementos con medios, y cada uno con un atributo de selectividad
- La primera entrada aceptable es reproducida
 - Listado por orden de preferencia
 - Última entrada aceptada por defecto

Ejemplos con <switch>



```
<switch>
  <audio src="pele_version_francesa.rm" systemLanguage="fr"/>
  <audio src="pele_version_alemana.rm" systemLanguage="de"/>
  <audio src="pele_version_espanola.rm" systemLanguage="es"/>
  <audio src="pele_version_japonesa.rm" systemLanguage="ja"/>
  <audio src="pele_version_coreana.rm" systemLanguage="ko"/>
  <audio src="pele_version_inglesa.rm"/>
</switch>
```

Reproduce una de las 5 primeras versiones según el idioma. Por defecto la inglesa

```
<switch>
  <ref src="pele_calidad_optima.rm" systemBitrate="1000000" />
  <ref src="pele_calidad_alta.rm" systemBitrate="512000" />
  <ref src="pele_calidad_normal.rm" />
</switch>
```

Reproduce el vídeo con la calidad más alta posible según el ancho de banda de la red de acceso del cliente

Accesibilidad



Accesibilidad: Atributos principales



- Atributos de elementos de objetos de medios
 - alt=** Contiene texto alternativo
 - longdesc=** Suplementario a alt, pero más largo; debería incluir descripciones de áreas
 - readIndex=** Orden de lectura en voz alta para reproductores con esta capacidad (para discapacitados visuales)

Adaptación del layout



- Adaptación a diferentes tamaños de pantalla
 - Coordenadas relativas a las dimensiones de la ventana del reproductor
 - Estrategias alternativas de disposición espacial
- Selección en layout y region
 - Diferentes layouts o regiones pueden usar tests de selectividad para que se elija el más adecuado
- Ejemplos
 - Hacer espacio a subtítulos
 - Reorganización para tamaños de pantalla que varían

Otros aspectos



SMIL permite aún muchas más cosas



- Transiciones entre elementos (como PowerPoint)
- Animaciones (cambio del valor de un atributo en el tiempo, p.e. posición)
- Definición de atributos de selectividad a medida (p.e. edad de la audiencia ...)
- Precarga de medios (prefetch)
- Cambiar la velocidad en el paso del tiempo: acelerar y decelerar
- Reproducción de medios "marcha atrás"
- Uso del teclado para "navegación rápida" (tecla TAB)
- etc. (¿dijimos ya que son 50 módulos?)

Futuro



Desafíos de SMIL



- Mercado multimedia y móvil emergente
- Futuros formatos de SMIL
- Futuro desarrollo de SMIL
- **SOPORTE EN REPRODUCTORES REALES**