

LENGUAJES DE MARCA Y SISTEMAS DE GESTIÓN DE LA INFORMACIÓN.

Actividad 1.

Introducción

De todos los lenguajes que he buscado y probado he decidido para esta actividad resaltar SMIL(Synchronized Multimedia Integration Language) y GML(Geography Markup Language),a continuación expongo sus principales características,elementos,ventajas y ficheros ejemplos de cada uno.

SMIL

Características

Es un lenguaje de marcado que en XML que permite realizar presentaciones combinando distintos contenidos multimedia interactivos (imágenes, audio, video, texto...)

Aunque la utilidad más conocida es la creación de presentaciones multimedia o transparencias, también se puede utilizar para:

- Subtitulado de películas.
- Apoyo a disminuidos sensoriales.
- Karaoke.
- Noticiarios teletipo.

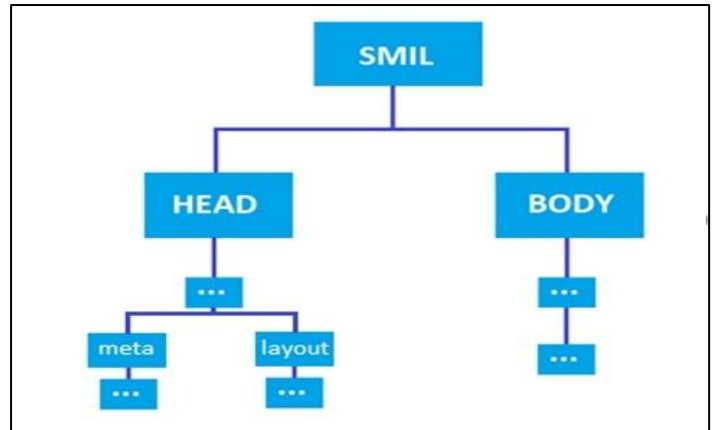
Y en las presentaciones multimedia se puede hacer:

- Describir el comportamiento temporal de la presentación.
- Describir la distribución de los elementos de la presentación en la pantalla.
- Crear hiperenlaces a otros objetos multimedia, otras presentaciones SMIL o a otras páginas.
- Utilizar botones para controlar el flujo de la presentación.
- Utilizar funciones que permiten definir diferentes secuencias y duración de los elementos de la presentación

Elementos

Al igual que sucede con todos los lenguajes XML, los documentos SMIL deberán poseer una estructura determinada que será común en todos los documentos. Los elementos básicos de SMIL son:

- SMIL: Es el elemento básico y contiene a los demás elementos, sobre él se construye la presentación y se colocan las regiones y objetos multimedia.
- Head: Describe la estructura general de la presentación. (tamaño, posicionamiento...)
- Body: En este se define qué clase de elementos va a contener la presentación SMIL y su comportamiento en el tiempo.



Además los anteriores, otros elementos importantes son:

- Text. Indica atributos obvios como la parte de la pantalla en donde va a aparecer, Text emplea un descriptor, MIME, similar al utilizado por los sistemas de correo electrónico, y que indica si se trata de texto plano o de texto con estilo (color, negrillas, fuentes diversas, etc.).
- Img. Se utiliza para mostrar una imagen estática indicando su nombre, el URL (la dirección en donde se encuentra que puede ser una carpeta o un sitio web).
- Audio. Se utiliza para reproducir secuencias de audio. Los tipos de formato de audio que se pueden utilizar dependen de la capacidad para reproducirlos que tenga el sistema operativo.
- Video. Se utiliza para reproducir secuencias de video indicando el tamaño y ubicación de la pantalla que los va a mostrar, así como su duración. Pueden utilizar son aquellos soportados por el reproductor instalado.
- Par. Significa paralelo e indica qué elementos se van a reproducir de manera paralela o a la vez.
- Seq. Su significado es secuencia y se emplea para reproducir elementos multimedia uno seguido de otro.

A continuación se muestra una foto de todos los elementos del lenguaje SMIL.

• a SMIL 3.0 Linking	• paramGroup SMIL 3.0 Media Object
• anchor SMIL 3.0 Linking	• prefetch SMIL 3.0 Content Control
• animate SMIL 3.0 Animation	• priorityClass SMIL 3.0 Timing and Synchronization
• animateColor SMIL 3.0 Animation	• ref SMIL 3.0 Media Object
• animateMotion SMIL 3.0 Animation	• region SMIL 3.0 Layout
• animation SMIL 3.0 Media Object	• regPoint SMIL 3.0 Layout
• area SMIL 3.0 Linking	• root-layout SMIL 3.0 Layout
• audio SMIL 3.0 Media Object	• send (**) SMIL 3.0 State
• body SMIL 3.0 Structure	• seq SMIL 3.0 Timing and Synchronization
• br (**) SMIL 3.0 smilText	• set SMIL 3.0 Animation
• brush SMIL 3.0 Media Object	• setValue (**) SMIL 3.0 State
• clear (**) SMIL 3.0 smilText	• smil SMIL 3.0 Structure
• customAttributes SMIL 3.0 Content Control	• smilText (**) SMIL 3.0 smilText
• customTest SMIL 3.0 Content Control	• span (**) SMIL 3.0 smilText
• delvalue (**) SMIL 3.0 State	• state (**) SMIL 3.0 State
• div (**) SMIL 3.0 smilText	• submission (**) SMIL 3.0 State
• excl SMIL 3.0 Timing and Synchronization	• switch SMIL 3.0 Content Control
• head SMIL 3.0 Structure	• tev (**) SMIL 3.0 smilText
• img SMIL 3.0 Media Object	• text SMIL 3.0 Media Object
• layout SMIL 3.0 Layout	• textStyle (**) SMIL 3.0 smilText
• meta SMIL 3.0 Metainformation	• textStyling (**) SMIL 3.0 smilText
• metadata SMIL 3.0 Metainformation	• textstream SMIL 3.0 Media Object
• newValue (**) SMIL 3.0 State	• topLayout SMIL 3.0 Layout
• p (**) SMIL 3.0 smilText	• transition SMIL 3.0 Transition Effects
• par SMIL 3.0 Timing and Synchronization	• transitionFilter SMIL 3.0 Transition Effects
• param SMIL 3.0 Media Object	• video SMIL 3.0 Media Object Chapter

Ventajas

- Su gran flexibilidad.
- La independencia que mantiene entre los diferentes elementos multimedia que contiene.
- La gran adaptabilidad a las preferencias del usuario.

Fichero ejemplo

A continuación se puede observar un sencillo ejemplo (ejemplo1) del uso de SMIL. En él se definen dos regiones una para el audio y otra para la imagen. Además también se le añade un texto explicativo a la imagen. El código fuente sería:

```
<?xml version="1.0" encoding="UTF-8"?>
<!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>
    <root-layout title=" RADIO IMAGEN " id="layout1" width="300" height="400"/>
    <region id="R1" left="0%" right="0%" width="300" height="300" fit="fill"/>
    <region id="R2" left="0%" right="0%" width="300" height="100" top="300" fit="fill"/>
    <region id="R3" left="0%" right="0%" fit="fill"/>
  </layout>
</head>
<body>
  <par>
    
    <textstream src="./texto1.rt" region="R2" />
    <audio src="./musica.mp3" region="R3" />
  </par>
</body>
</smil>
```

Editores

Para este lenguaje me ha parecido mejor utilizar el editor que uso normalmente que es el visual studio code ya que este editor te permite ver en vivo lo que estas editando en el navegador en tiempo real gracias a una extensión que te lo permite ver, por lo que si estamos editando elementos multimedia lo más práctico es ver como está quedando visualmente.

DOCBOOK

Características

DocBook es una aplicación del estándar SGML/XML e incluye una estructura de documento (DTD) propia y que se utiliza de manera más destacada en el área de la documentación técnica, especialmente para documentar todo tipo de material y programas informáticos.

DocBook inicialmente comenzó como una DTD de SGML, pero a partir de la versión 4 existe un equivalente para XML.

DocBook es un lenguaje semántico que nos permite crear documentos en un formato neutro, independiente de la presentación, es decir, que se recogen tanto el contenido como la estructura lógica del mismo, permitiendo así que pueda ser publicado (presentado) automáticamente en multitud de formatos: **HTML**, **XHTML**, **EPUB**, **PDF**, **man pages**, **HTML Help**, etc., simplemente aplicando "plantillas" de presentación, sin que sea necesario ningún cambio sobre el documento original.

Elementos

Dentro de estos lenguajes podemos encontrar varios elementos categorizados por etiquetas y son las siguientes:

Etiquetas estructurales

Especifican características generales de sus contenidos. Por ejemplo, el elemento *book* especifica que sus elementos hijo serán partes de un libro: *títulos*, *capítulos*, *glosarios*, *apéndices*, etc. Algunas etiquetas estructurales son:

- set:
- book:
- part:
- article:
- chapter:
- appendix:
- dedication:

Los elementos estructurales pueden contener a otros elementos estructurales, pero han de ser siempre elementos de primer nivel dentro de un documento DocBook.

Etiquetas de bloque

Representan elementos tales como *párrafos*, *listas*, etc., y no todos ellos han de contener necesariamente texto en su interior.

Estos elementos de bloque suelen ir distribuidos secuencialmente, y serán visualizados uno "debajo" de otro.

Etiquetas de línea

Representan elementos tales como *letras enfatizadas*, *hyper-enlaces*, etc, y se suelen aplicar a porciones de texto en el interior de un elemento de bloque, provocando habitualmente que la herramienta procesadora de presentación aplique algún tipo de tratamiento tipográfico especial a dichas porciones.

Ventajas

-Estructuración de los datos

Docbook nos permite que nuestros documentos posean una estructura de forma que podamos mantener orden dentro del mismo y habilitar características como tablas de contenidos o índices generados automáticamente.

-Similitud con HTML

Crear documentos en Docbook es parecido a escribir documentos en HTML. Si ya has trabajado con páginas web, no te costará habituarte a la sintaxis y la forma de trabajo de Docbook.

-Versatilidad

Con Docbook puedes crear casi cualquier tipo de documento que necesites. Además, dadas sus características podrás comprimirlo o enviarlo a través de red con facilidad. Por otro lado, Docbook es fácil de analizar o extraer la información semántica que aporta.

Fichero de ejemplo

```
<?xml version="1.0" standalone="no"?>
  <!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN"
    "http://www.oasis-open.org/docbook/xml/4.2/docbookx.dtd">
```

DTD PROPIA ARRIBA

```
<article lang="es">
  <title>Creando Documentación Técnica con
  DocBook</title>
```

```
<articleinfo>
  <author>
    <firstname>Leonardo</firstname>
    <surname>Caballero</surname>
  </author>
</articleinfo>
```

```
<section id="art-intro">
```

```
<title>Introducción</title>
```

```
<para>DocBook es un aplicación del estándar  
SGML/XML e incluye una DTD propia y que se utiliza de manera  
más destacada el área de la documentación  
técnica, especialmente para documentar todo tipo de  
material y programas informáticos, para mayor  
información consultar aquí <ulink  
url="http://www.oasis-open.org/docbook/" />.</para>
```

```
</section>
```

```
</article>
```

En este fichero podemos encontrarnos la estructura enfocada a la documentacion técnica,ya que viendo las etiquetas y los atributos nos damos cuenta de qué tipo de aplicación se le da al lenguaje.

Si nos fijamos podemos encontrar también la declaración de DTD propia mencionada en el fichero.

Editores

Para este tipo de lenguaje mi opinión es que el mejor editor que hay para mi gusto es VIM, ya que al estar acostumbrado a la colorizacion de elementos en HTML este editor nos lo ofrece con la ventaja de ser más basico que los demás,ya que los demás se basan en editores de java como jedit,emacs y xxe.