

# **GENERADOR DE SIGNOESCRITURA**

**Marta Herrador Iradier, José G. Zato Recellado**

Escuela Universitaria de Informática de la Universidad Politécnica de Madrid

Campus Sur, Ctra. de Valencia km 7. 28031 Madrid, España

{mherrador,jzato}@eui.upm.es

**Palabras clave:** Signoescritura, Lengua de Signos, Deficiencia Auditiva, Discapacidad, Comunicación, XML, SVG

Campo temático: Accesibilidad para la comunicación

CAMPO TEMÁTICO Bloque II. Accesibilidad para la Comunicación

## **RESUMEN**

En una sociedad mayoritariamente oyente, las personas sordas deben hacer frente no sólo a los problemas derivados por su discapacidad, sino también a aquellos que la sociedad impone en forma de barreras de comunicación.

Hace algunos años se introdujo la Signoescritura en España como método de escritura de la lengua de signos española. Desde entonces, ha sido necesaria la creación de un sistema que permitiera fácil e intuitivamente la generación de estos signos.

El objetivo de este trabajo es desarrollar un sistema que permita la generación de contenidos escritos en lengua de signos, se centra fundamentalmente en la signoescritura pero no limita la posibilidad de generar los mismos contenidos en otros formatos. Con la tecnología actual, se puede llevar a cabo esta generalización, de modo que se puede partir de un documento que representa un mensaje en lengua de signos de forma genérica y que puede ser presentado gráficamente en cualquier notación. Este enfoque se basa en separar los contenidos en lengua de signos de la representación en notaciones específicas utilizando la tecnología XML y el lenguaje basado en XML para gráficos vectoriales SVG.

Como implementación a estas ideas se está desarrollando un prototipo que permite la edición de contenidos en signoescritura.

## 1.- INTRODUCCIÓN

Durante décadas, las personas que sufren algún tipo de discapacidad, se han encontrado con continuas discriminaciones y limitaciones que les han impedido llevar una vida dentro de la normalidad impuesta por los no discapacitados.

Actualmente, el papel que puede desempeñar la tecnología es indiscutible. La tecnología puede ser utilizada durante el proceso educativo, en un proceso de rehabilitación, en la comunicación, o simplemente en el acceso a la información.

El principal objetivo del trabajo presentado en este artículo es mejorar la calidad de vida de un colectivo específico: las personas sordas signantes<sup>1</sup>, ofreciéndoles la posibilidad de escribir su propia lengua (en cualquiera de las notaciones existentes) de una forma cómoda y útil.

Actualmente, en la mayoría de las ocasiones, se ven forzados a adaptarse a lenguas escritas como el castellano o el inglés, reduciendo así sus posibilidades de acceso a las facilidades de las que disfruta la comunidad normo-oyente.

Se propone por tanto proporcionar a este colectivo un sistema completo que permita la edición, publicación e intercambio de contenidos escritos en lengua de signos mediante la generación de Signoescritura, una notación para escribir lenguas de signos que utiliza símbolos gráficos para representar los componentes que intervienen en la lengua de signos, como la configuración de las manos, los movimientos y las expresiones faciales.



Fig.1.- Ejemplo de Signoescritura

## 2.- DESCRIPCIÓN DEL ENTORNO DE TRABAJO

### 2.1.- DIFICULTADES INICIALES

La naturaleza visual (no textual) de las notaciones existentes para escribir las lenguas de signos requiere un enfoque diferente al de los editores de texto convencionales. El teclado no puede seguir siendo la entrada estándar al no existir relación directa entre las teclas del teclado y los signos que componen la lengua de signos.

---

<sup>1</sup> Personas sordas que utilizan la Lengua de Signos

La Lengua de Signos es el sistema natural de comunicación entre las personas sordas, es un sistema visual y gestual, no oral.

El primer enfoque para representar la lengua de signos en formato electrónico se centró obviamente en los formatos estándares gráficos de mapas de bits (bmp, gif, jpg..) Un punto de partida lógico, pero lleno de dificultades ya que cada signo es considerado una unidad difícilmente tratable. Sólo hay que imaginar lo que puede suponer tratar de editar texto en un fichero bmp, o incluso peor, tratar de determinar si dos ficheros gráficos contienen el mismo signo sin utilizar complejas técnicas de reconocimiento de formas.

La aparición de un nuevo formato SVG<sup>2</sup> (Recomendación del W3C de sept. 2001) provocó el desarrollo de un nuevo enfoque basado en este formato.

## 2.2.- ENFOQUE ACTUAL DEL GENERADOR DE SIGNOESCRITURA

El usuario elige a través del interfaz de usuario, la configuración de todos los componentes, manuales y no manuales del signo que quiere generar. Internamente, y de forma transparente al usuario, se genera un documento XML que sigue el esquema de SIGML<sup>3</sup> y especifica los componentes del signo seleccionado.

```
<hamgestural_sign>
  <sign_manual>
    <handconfiguration handshape="finger2345" />
  </sign_manual>
  <sign_nonmanual>
    <facialexpr_tier>
      <eye_brows movement="RB_both_eyebrows_raised" />
      <eye_lids movement="WB_wide_open_eyelids" />
    </facialexpr_tier>
  </sign_nonmanual>
</hamgestural_sign>
```

Fig.2.- Extracto de un documento XML

El eje fundamental de la aplicación debe ser por tanto, un documento XML que especifica un mensaje en lengua de signos. Partiendo del documento generado, se pueden aplicar distintos tratamientos para obtener distintos formatos de salida, en este caso, interesa representar gráficamente el signo.

Para ello, se utiliza la herramienta XSLT, aplicación XML que define un lenguaje para modificar documentos XML mediante plantillas XSL.

<sup>2</sup> SVG (Scalable Vector Graphics) es un lenguaje para describir gráficos vectoriales bidimensionales, tanto estáticos como animados en XML

<sup>3</sup> SIGML (Sign Gesture Markup Language) es una descripción de signos genérica basada en un documento XML desarrollado dentro del proyecto Visicast (Elliot et al. 2000).

Se transforma el documento XML en un documento SVG, estableciendo correspondencia entre cada elemento del documento XML y su representación gráfica en SVG.

```
<xsl:when test="@movement='WB_wide_open_eyelids'">
  <circle cx="165" cy="92" r="5" id="eyelids" style="stroke:black;stroke-
width:2;fill:none" />
  <circle cx="185" cy="92" r="5" id="eyelids" style="stroke:black;stroke-
width:2;fill:none" />
</xsl:when>
```

Fig.3.- Fragmento de plantilla de traducción XSL

De esta forma, partiendo del documento XML anterior, y realizando sobre él la transformación oportuna, se obtiene un documento SVG.

El resultado se muestra en el componente SVGCanvas del interfaz de la aplicación y además se crea un documento SVG independiente.

Para el usuario el resultado es el siguiente:

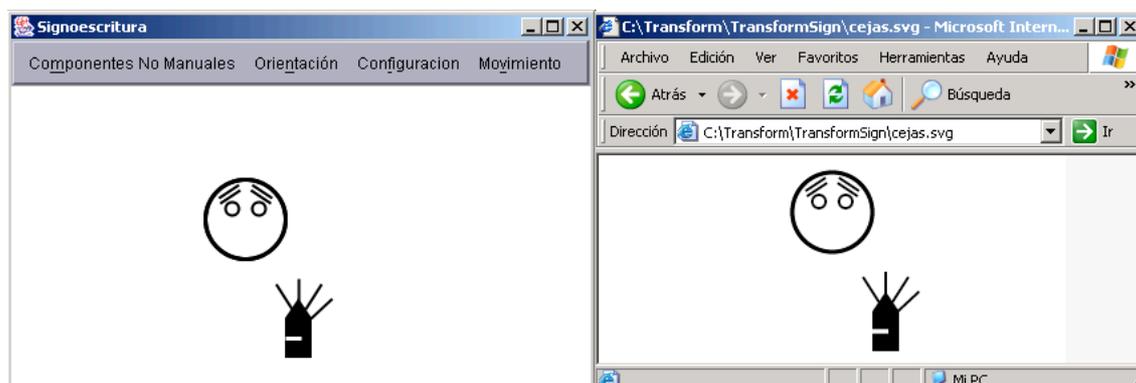


Fig.4.- Resultado en Signoescritura

La parte izquierda de la figura corresponde al resultado mostrado en el interfaz de la aplicación, en la parte derecha puede observarse el documento generado de forma independiente. En nuestro caso, interesa que el objeto de salida sea un documento SVG, pero si modificamos la plantilla XSL se puede obtener cualquier otro resultado.

El usuario, puede decidir qué tipo de representación para ese mensaje quiere, puede interesarle la Signoescritura, en cuyo caso aplicaremos la plantilla XSL correspondiente que transforma dicho documento XML en un documento gráfico SVG en Signoescritura (como puede observarse en la imagen). Puede interesarle recibir dicha información en HamNoSys<sup>4</sup>, para lo que seguiremos el mismo proceso con la plantilla correspondiente de HamNoSys, y así sucesivamente con los formatos oportunos, HTML si quiere la información en WEB, WML si

<sup>4</sup> HamNoSys (Hamburg Sign Language Notation System) Sistema de escritura de lenguas de signos desarrollado como herramienta científica en el campo de la investigación

prefiere que le sea presentada en formato WAP o SIGML si quiere que sirva de entrada para un signante virtual.

### **3.- FUNCIONALIDAD DE LA APLICACIÓN**

No existe el concepto de Accesibilidad Universal. Las necesidades de cada discapacidad son muy diferentes, por lo que los criterios de accesibilidad son igualmente diferentes.

Si centramos la atención en un usuario con discapacidad visual, se debe intentar que la información que se presente al usuario sea principalmente texto (para poder ser reconocido por un lector de pantalla), en cambio, si se trata de un usuario con discapacidad auditiva, la información deberá presentarse en sentido opuesto, es decir, lo más gráfica y visual posible, evitando grandes párrafos que pueden resultar ilegibles. ¿Cuál es el criterio de accesibilidad correcto? Evidentemente, los dos son igualmente correctos, cada uno aplicado a una limitación concreta.

Destacamos por ello dos aspectos relevantes:

#### **3.1.- SEPARAR EL CONTENIDO DE LA PRESENTACIÓN.**

El contenido debe ser presentado en un lenguaje interpretable por la máquina aunque no por el usuario final. Es éste quien debe decidir que tipo de representación quiere para ese contenido. En nuestro caso, esto se traduce en que la aplicación especifica un mensaje en lengua de signos, pero no se puede determinar a priori si el usuario preferirá leer el mensaje en Signoescritura, HamNoSys o una explicación textual del signo.

#### **3.2.- GENERACIÓN AUTOMÁTICA DE SIGNOS**

Como consecuencia de lo anterior, ya que nuestros signos se pueden generar en distintos formatos, necesitaríamos una gran capacidad de almacenamiento para poder almacenar todos los signos en todos los posibles formatos.

Por ello, destacamos también la importancia de que los signos se generen automáticamente, es decir, no se precisa de una base de datos que almacene todos los posibles signos. A través de las hojas de estilo XSL, definimos las instrucciones necesarias para crear un signo, y cuando el usuario selecciona un signo, se crea automáticamente siguiendo estas instrucciones.

## 4.- CONCLUSIONES Y FUTUROS PROYECTOS

Con este enfoque de representación de signos se pretende dar una solución global a las necesidades existentes de un grupo concreto de usuarios. En este sentido se requiere una fase de pruebas por parte de los usuarios a los que está destinado este trabajo para poder definir un interfaz amigable y eficiente.

Se espera que el entorno de trabajo crezca para incluir distintas funcionalidades como la creación de diccionarios específicos dependientes de una lengua concreta<sup>5</sup> y su integración en el editor para permitir algún tipo de corrección “ortográfica”

Otra característica interesante es la creación de páginas y sitios web en lengua de signos, que podría conseguirse mediante una extensión del editor.

## REFERENCIAS

- Elliot, R. et al. The development of language processing support for the Visicast Project. 4<sup>th</sup> International ACM SIG-CAPH Conference on Assitive Technologies. New York, 2000.
- Parkhurst, S., Parkhurst, D. “Un sistema completo para escribir y leer las Lenguas de Signos”. PROEL. 2000
- Proyecto VISICAST. [on line] <http://www.visicast.co.uk/>
- Signwriting, read, write, type sign languages [on line] <http://www.signwriting.org>
- Mc Laughlin, Brett. “Java & XML“. Ed. O’Reilly, Sept.2001

---

<sup>5</sup> La Lengua de Signos no es Universal