

Edición y Visualización del Lenguaje Matemático en la Aplicación Web SHARPO

Ana-Belén Gil¹, Raquel Rodríguez¹, Francisco J. García¹

¹ Departamento de Informática y Automática
Facultad de Ciencias – Universidad de Salamanca
abg@usal.es, raquelrh@gmail.com, fgarcia@usal.es

Abstract. En el presente artículo se aborda la problemática de incluir el lenguaje matemático en soporte Web. Generar digitalmente el conjunto de símbolos o caracteres gráficos que son utilizados en matemáticas para su perfecta definición y presentar cada uno de estos elementos de manera que puedan ser servidos y consumidos por cualquier ordenador en la red, con cualquier navegador y sistema operativo de modo que dicha codificación se pueda editar en cualquier máquina, indexar los contenidos y reutilizar dichas expresiones es la meta. Se realiza un recorrido por las diferentes opciones estudiadas para solventar este problema determinante a la hora del desarrollo de SHARPO, Sistema Hipermedia Adaptativo aplicado a la Resolución de Problemas Online que incorpora un editor de contenidos basado en MathML.

Keywords: Lenguaje Matemático, Web, Entornos de aprendizaje en Web, MathML.

1 Introducción

Son muchas las aplicaciones desarrolladas como una herramienta del uso de las nuevas tecnologías a la rama educativa en la enseñanza de las matemáticas. El uso del ordenador en el área de matemáticas, se fundamenta cognitivamente en el aporte de la herramienta de aprendizaje como potenciadora y motivadora del alumno ([1], [2]). Se han integrado de manera natural con las tecnologías hipermedia con el uso del ordenador para desarrollar juegos matemáticos, bien sobre estrategias de resolución de problemas y toma de decisiones, bien sobre contenidos curriculares específicos, sigue teniendo una amplia vigencia.

Las matemáticas tienen, un lenguaje particular, específico, el cual simplifica, en algunos casos, la comunicación, y por otro lado clarifica y designa de una manera exacta, sin posible confusión, sus contenidos. En este lenguaje, que podemos llamar lenguaje matemático, las afirmaciones son presentadas de una manera propia, siendo tajantes, con demostraciones de su veracidad, y sin permitir ambigüedades. Todos y cada uno de los símbolos de escritura definidos y utilizados tienen una tarea determinada, exacta, sin solapamientos ni posibles equívocos, mientras que también la estructura de su presentación es idónea para su perfecta comprensión. Codificar el lenguaje matemático en formato digital para poder establecer la comunicación Web supone un requisito añadido a cualquier desarrollo de soporte a la docencia on-line de las matemáticas.

El problema de cómo codificar las matemáticas en un formato digital para la comunicación electrónica es más antiguo que la web. Sin embargo, debido al uso de ésta como medio de difusión, este tema ha cobrado una especial importancia como elemento crucial en cualquier desarrollo. La demanda para permitir, de manera efectiva, la comunicación científica electrónica es algo que aumenta día a día. Cada vez más, los investigadores, los científicos, los ingenieros, los educadores, los estudiantes y los técnicos trabajan y colaboran a distancia confiando en la comunicación electrónica. Sin embargo, la manera predominante de transmitir notación científica en la Web se fundamenta en métodos basados en imágenes que son primitivos e inadecuados. La calidad de los documentos es mala, la edición es complicada y a veces imposible, y la información matemática contenida en las imágenes no se presta a búsquedas, registro o aprovechamiento en otras aplicaciones.

Con el fin de dar una solución al problema de la edición y visualización del lenguaje matemático para el desarrollo de nuestra herramienta Web SHARPO[3], Sistema Hipermedia Adaptativo aplicado a la Resolución de Problemas Online, realizamos un estudio de posibles vías que el artículo describe. La sección segunda plantea la historia tecnológica de la edición y visualización web de las matemáticas. La sección tercera detalla los problemas de la inserción de los símbolos matemáticos en forma de imágenes en el HTML de las páginas Web. La sección cuarta pasa a presentar el lenguaje de marcado MathML y su adecuación tecnológica en los navegadores existentes. En la sección quinta revisamos algún software relacionado con los procesos de edición y visualización de lenguaje matemático para la Web. La sexta sección muestra el módulo de edición matemático incorporado en nuestra aplicación Web, SHARPO, resumiendo sus funcionalidades. Por último, en la sección séptima se realizan unas observaciones, a modo de resumen final.

2 El Lenguaje Matemático y la Edición y Visualización Web

La problemática de la edición digital del lenguaje matemático supone ya un tratamiento especial si el formato final es el de un documento impreso. Pero si lo que se quiere es que permita la comunicación electrónica el problema ya viene de antes de la aparición de la Web. La práctica común entre los científicos antes de la web, era la de escribir documentos en alguna forma codificada basada en el conjunto de caracteres ASCII, y enviarlo mediante correo electrónico al destinatario, ver **¡Error! No se encuentra el origen de la referencia.** pero este formato es demasiado limitado.

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Fig. 1. Fórmula en ASCII

En 1984 apareció TeX [4], un lenguaje desarrollado por Donald Knuth que se volvió un método de marcado para las matemáticas, ver Fig. 2, usado ampliamente hasta la actualidad.

$$\left(x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \right)$$

Fig. 2. Fórmula equivalente a la fig 1 en TeX

El problema con TeX es que es un sistema tipográfico, es decir, un sistema que fija un estándar para la calidad visual de la materialización en papel del documento. Además es muy intensivo en procesamiento en el renderizado de los documentos.

Casi a la vez, Leslie Lamport (LamportTeX) desarrolla LaTeX [5] que es un procesador de textos formado por un gran conjunto de macros de TeX, con la intención de facilitar el uso del lenguaje de composición tipográfica creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados con LaTeX1 es comparable a la de una editorial científica de primera línea.

Desde un principio, la Web ha demostrado ser un método muy efectivo para intercambiar información entre grupos de individuos distanciados físicamente. Sin embargo, si bien la World Wide Web fue inicialmente concebida e implementada por científicos y para científicos, las posibilidades para incluir expresiones matemáticas en ella han estado muy limitadas. Actualmente, la mayoría de las matemáticas que hay en la web constan de texto con imágenes para la notación científica que son difíciles de leer o publicar, o de documento enteros en PDF. La *World Wide Web Consortium* (W3C) reconoció que la falta de apoyo para la comunicación científica era un problema serio. Dave Raggett incluyó una propuesta para matemáticas en HTML en el borrador del HTML 3.0 en 1994.

En abril de 1995 se celebró una mesa redonda sobre marcado matemático en la *WWW Conference* en Darmstadt. En noviembre de 1995, representantes de Wolfram Research presentaron una propuesta para incluir matemáticas dentro de HTML para el equipo W3C.

En mayo de 1996, la Reunión del *Digital Library Initiative en Champaign* jugó un papel muy importante reuniendo a muchas partes interesadas. Después de la reunión, se formó el *HTML Math Editorial Review Board*. En los años siguientes, este grupo creció, y ya estaba formalmente reconstituido como el primer *W3C Math Working Group* en marzo de 1997.

En abril de 1998 se aprobó la recomendación MathML 1.0, editada por Patrick Ion y Robert Miner. En julio de 1999 se aprobó la recomendación MathML 1.01 [6], editada por Patrick Ion y Robert Miner, que básicamente corregía errores de la primera versión. En febrero de 2001 se aprobó la recomendación MathML 2.0, editada por Robert Carlisle y otros, en la que se introdujeron y eliminaron algunos elementos. En octubre de 2003 se aprobó la recomendación MathML 2.0 (2º edición), editada por Robert Carlisle y otros, en la que se introdujeron y eliminaron algunos elementos.

En junio de 2006 se creó el grupo de trabajo que debe redactar la recomendación MathML 3.0 en un plazo de dos años. Actualmente, el W3C está preparando las recomendaciones MathML 3.0 [7] y *A MathML for CSS profile* [8].

¹ <http://www.latex-project.org/>

3 Símbolos Matemáticos en Web Basados en Imágenes

La demanda para permitir, de manera efectiva, la comunicación científica electrónica está en aumento. Cada vez más, los investigadores, científicos, ingenieros, educadores, estudiantes y los técnicos trabajan y colaboran a distancia confiando en la comunicación electrónica. Sin embargo, la manera predominante de transmitir notación científica en la Web se basa en los métodos basados en imágenes que son primitivos e inadecuados. Básicamente la calidad de los documentos es mala, la edición resulta complicada y a veces imposible. La información matemática contenida en las imágenes no se presta a búsquedas, edición, registro o aprovechamiento en otras aplicaciones que pudieran consumir esta información. Las ecuaciones basadas en imágenes son generalmente más difíciles de ver, leer y comprender que el texto normal en la ventana del navegador. Además, estos problemas se empeoran cuando el documento se imprime. La resolución de las imágenes de ecuaciones será aproximadamente 70 puntos por pulgada, mientras el texto normal típicamente será 300, 600 o más puntos por pulgada. Esta disparidad en la calidad es inaceptable en la presentación de un documento.

Si se trata de hacer una búsqueda en el documento que incluye las fórmulas o símbolos matemáticos, o si se quiere cortar y pegar una ecuación en otra aplicación o una subexpresión, al tener ecuaciones basadas en imágenes, ninguna de estas necesidades podría satisfacerse de manera adecuada. Aunque el uso del atributo alt en el documento fuente puede ayudar, es evidente que los documentos web que busquen la interactividad deben proveer una interfaz más sofisticada entre navegadores y la notación matemática. Supone por ejemplo una grave dificultad al cumplimiento de la accesibilidad Web para aquellas personas con alguna discapacidad que utilizan ayudas técnicas para el acceso.

Otro problema que surge codificando expresiones matemáticas con imágenes es que se precisa más ancho de banda. Un lenguaje de marcado describiendo una ecuación es mucho más pequeño y más comprensible que una imagen de esa misma ecuación. Además, usando codificación basada en lenguajes de marcado la mayoría del proceso de interpretación se traslada a la máquina cliente. Por ello se precisa el uso de un lenguaje de marcado específico que abordaremos en los siguientes puntos.

4 MathML

MathML (*Mathematical Markup Language*) es un lenguaje de marcado basado en XML, con una fuerte influencia de TeX y cuyo objetivo es expresar notación matemática de forma que distintas máquinas puedan entenderla, para su uso en combinación con XHTML en páginas web, y para intercambio de información entre programas de tipo matemático en general. El soporte de MathML es grande en programas matemáticos (como Maple² o Mathcad³), pero aún escaso en navegadores (sólo vía *plugins* en Internet Explorer, parcial en Mozilla).

² <http://www.maplesoft.com/>

³ www.mathsoft.com/

Se ha puesto un gran esfuerzo para que MathML tenga la misma calidad de representación, pero en materializaciones diversas, que TeX, ya que como se ha mencionado ha sido una de sus mayores influencias. Además de esto, se fijaron múltiples objetivos en la creación de MathML.

El principal usuario de este lenguaje es la comunidad científica, lo que genera una serie de requisitos:

- Codificar material matemático útil para la enseñanza y la comunicación científica a todo nivel.
- Codificar tanto notación matemática como significado matemático.
- Facilitar la conversión desde y hacia otros formatos de presentación (por ejemplo TeX). Lo cual resulta bastante evidente, no tiene sentido inventar una notación nueva si se va a perder todo lo que se tiene actualmente.
- Conveniente para la interacción con software externo. Esto se refiere en particular a generadores de código o posibles intérpretes e inclusive evaluadores de expresiones.
- Ser extensible. No es posible definir toda la matemática, por lo que aquello que no quede definido y sea alguna vez necesario debe de ser posible de definirse.

No importa cuán exitosamente MathML alcance estos objetivos como lenguaje de marcado, está claro que MathML sólo será útil si es implementado correctamente. Con este fin, el W3C *Math Working Group* ha identificado una preselección de objetivos adicionales de implementación. Éstos intentan describir concisamente la funcionalidad mínima que el software de renderizado y procesado de MathML debería satisfacer.

- Las expresiones MathML en páginas de HTML (y XHTML) deberían renderizarse correctamente en la mayoría de navegadores de Internet, conforme a las preferencias de visualización del lector y en la calidad más alta posible dadas las capacidades de la plataforma.
- Documentos que contengan expresiones MathML en HTML (y XHTML) se deben imprimir correctamente en gran resolución en impresoras de alta calidad.
- Las expresiones MathML en páginas web deberían reaccionar a eventos del usuario, por ejemplo con el ratón, y deberían coordinar la comunicación con otras aplicaciones a través del navegador.
- Los editores de expresiones matemáticas y los convertidores deberían ser desarrollados para facilitar la creación de páginas web conteniendo expresiones MathML.

Estos objetivos han comenzado a ser cumplidos mediante elementos incrustados como Applets de Java, plugins y controles ActiveX para renderizar MathML. Sin embargo, el paso final para que estos objetivos sean cumplidos depende de la cooperación y soporte de fabricantes de navegadores y otros desarrolladores de software. El W3C Math Working Group ha continuado trabajando con los grupos de trabajo del Document Object Model (DOM) y el Extensible Style Language (XSL) para asegurar que las necesidades de la comunidad científica serán satisfechas en el futuro, y el fruto de esa colaboración se aprecia en las versiones que van apareciendo.

Este lenguaje de marcado presenta alguna complejidad para los usuarios por lo que MathML no está previsto para ser usado directamente por la mayoría de las personas que deseen publicar material, sino mediante editores de ecuaciones, programas de conversión, etc., que generen código MathML, muy similar a la situación que se

experimenta actualmente con los editores de HTML, donde uno diseña la visualización que desea, y el editor genera el código respectivo.

4.1 Descripción de MathML

Una de las cosas más importantes del lenguaje de marcado es el poder codificar la notación que representa a un objeto matemático y la estructura matemática del mismo. Es más, ambas codificaciones pueden ser usadas juntas para definir presentación y contenido de una idea matemática. Debido a esto es que se puede realizar una clasificación de los elementos de MathML, en estas tres categorías: elementos de presentación, elementos de contenido y elementos de interfaz.

4.1.1 Elementos de presentación

Corresponden a "constructores" de la notación matemática tradicional, es decir, los tipos básicos de símbolos y estructuras para la construcción de expresiones, a partir de los cuales cualquier parte de la notación tradicional de la matemática puede generarse. La descripción de las estructuras notacionales que los elementos representan se dan generalmente de una manera visualmente orientada. Sin embargo, los elementos son independientes de la materialización en el sentido de que han sido diseñados para contener suficiente información. Los elementos de presentación sólo sugieren maneras específicas de representar. Esta especificación describe con algún detalle reglas de representación visual recomendadas, pero un renderizador de MathML es libre de usar sus propias reglas mientras la representación sea comprensible.

4.1.2 Elementos de Contenido

La intención fundamental de la codificación de contenido en MathML es proveer una codificación específica de la estructura matemática subyacente de una expresión, más allá de cualquier representación particular para la expresión.

La principal razón para proveer esta codificación es que aún con el uso sistemático de etiquetas de presentación no se puede capturar la información semántica entregada por este sistema. La dificultad se origina porque existen muchos caminos distintos desde representación hacia la semántica y viceversa. Además la notación cambia con el tiempo y la cultura, por lo que si bien '*' se usa como multiplicación hoy, no quiere decir que sea usado así en unos años, o quizás en otros países.

Al codificar la información de esta forma se puede asegurar un intercambio de información mucho más preciso, incluso permitiendo la evaluación de las expresiones de una forma simple.

Debido a la falta de consenso en la semántica de la notación general, en MathML se optó por hacer explícito un número relativamente pequeño de construcciones matemáticas comunes, proveyendo de un mecanismo para asociar semánticas con nuevas construcciones notacionales. De esta manera el conjunto de construcciones puede ser extendido más allá del originalmente entregado.

El objetivo básico de la codificación de contenido es el establecimiento de relaciones explícitas entre las estructuras matemáticas y sus significados matemáticos. Cada estructura tiene una semántica predeterminadamente asociada y existe un mecanismo para asociar nuevos significados matemáticos con nuevas construcciones.

El uso de etiquetas (tags) específicas de contenido ofrece múltiples beneficios, como son la posibilidad de crear agentes que procesen de manera mucho más precisa el contenido.

Finalmente, las expresiones descritas en términos de elementos de contenido deben ser llevadas a una representación también. Para esto existen visualizaciones claras para expresiones comunes. El resto se obtiene de la claridad de la definición.

La codificación de contenido de MathML está basada en el concepto de árbol de expresión. En este árbol las hojas corresponden a objetos matemáticos básicos como son números (`<cn>15</cn>`), variables (`<ci>q</ci>`), etc. Los nodos intermedios generalmente representan algún tipo de función (`<power/>`) u otra construcción matemática que crea un objeto compuesto. Se puede decir que el elemento `<apply>...</apply>` es quizás uno de los más importantes, ya que es el que se usa para aplicar la función a sus argumentos. El orden de los hijos del elemento si importa, pero no está forzado en la DTD de *MathML*.

Los elementos de contenido de MathML pueden ser agrupados en las siguientes categorías según su uso: contenedores, operadores y funciones, calificadores, relaciones, condiciones, mapeos semánticos, constantes y símbolos

4.1.3 Elementos de interfaz

Los elementos de interfaz son aquellos relacionados en la generación y representación de MathML, siendo particularmente importantes los que respectan a insertar MathML en HTML y XHTML, puesto que la idea es codificar expresiones matemáticas en documentos Web.

Hay tres problemas principales en lo que respecta a insertar MathML en otros documentos XML. El primero es que MathML debe estar semánticamente integrado, es decir, debe ser reconocido como contenido XML válido insertado, y no como un error. Esto se soluciona mediante espacios de nombres, y una etiqueta particular que encapsula toda expresión MathML válida. Otro problema es, en el caso de HTML/XHTML, el renderizado de MathML debe estar integrado en el navegador, y aún son muy pocos los navegadores que soportan MathML. El tercer problema es que las herramientas para generar y procesar MathML deben ser capaces de comunicarse. Hay un número de herramientas que ya han sido o están siendo desarrolladas, y dado que las expresiones suelen ser largas y tienden a ser ingresadas erróneamente a mano, se debe poner especial énfasis en asegurar que el código puede ser generado con herramientas fáciles de usar, que sean independientes de plataforma y marca.

A continuación la Tabla 1 muestra el código para generar la ecuación de la Fig. 3, mediante notación de presentación y de contenido.

$$(x + y)^2$$

Fig. 3. Fórmula matemática

Tabla 1. Elementos de representación (izq.) y de contenido (dcha.)

<pre> <math> <msup> <mrow> <mo> (</mo> <mrow> <mi> x </mi> <mo> + </mo> <mi> y </mi> </mrow> <mo>) </mo> </mrow> <mn> 2 </mn> </msup> </math> </pre>	<pre> <math> <apply> <power/> <apply> <plus/> <ci>x</ci> <ci>y</ci> </apply> <cn>3</cn> </apply> </math> </pre>
--	---

4.2 MathML en Páginas Web

Una página web que contiene elementos MathML es un documento compuesto que contiene tanto elementos XHTML como MathML. El tipo del documento tiene que ser al menos XHTML 1.1 + MathML 2.0 y se debe servir al navegador con el tipo MIME application/xhtml+xml.

Normalmente los servidores sirven los documentos que tienen la extensión .html con el tipo MIME text/html y los documentos que tienen la extensión .xhtml con el tipo MIME application/xhtml+xml, por lo que conviene guardar los documentos que incluyan elementos MathML con la extensión .xhtml.

MathML en Firefox. Firefox es capaz de mostrar páginas con elementos MathML, desde la versión Mozilla 0.9.9 (marzo de 2002), gracias al trabajo realizado por el proyecto MathML de Mozilla. Es necesario instalar ciertas fuentes para poder mostrar correctamente todos los caracteres matemáticos. Si las fuentes no están instaladas, al abrir una página que contengan elementos MathML, se abrirá una ventana de aviso. La forma más sencilla de instalar esas fuentes en Windows es utilizar el instalador MIT MathML Fonts 1.0. Este paquete instala varias fuentes creadas por diferentes organizaciones (por ejemplo, las fuentes CM de TeX y las fuentes de Matemática). Próximamente se podrán utilizar las fuentes STIX, unas fuentes de 8047 caracteres creadas por el consorcio STI Pub, formado por varias organizaciones científicas norteamericanas y la editorial Elsevier.

MathML en Internet Explorer. Internet Explorer no es capaz de mostrar directamente páginas con elementos MathML, pero existen plugins gratuitos para conseguirlo, por ejemplo *MathPlayer* de *DesignScience*. Antes de instalar dicho plugin, cuando se intenta abrir una página con elementos MathML, Internet Explorer no muestra la página sino que muestra una ventana dando la posibilidad de guardar el archivo. Pero una vez instalado este, Internet Explorer es capaz de mostrar páginas con elementos MathML.

4.3 ASCIIMathML

ASCIIMathML convierte la notación matemática de ASCII a MathML. ASCIIMathML es un Javascript, por lo tanto, es necesario disponer de un navegador que sea capaz de interpretar este lenguaje (y que esté activado). Este JavaScript realiza la traducción de fórmulas incrustadas en documentos HTML de pseudo-LaTeX a MathML. Para utilizarlo basta con incluir el script en la cabecera de la página donde se vaya a usar y llamar posteriormente a la función “translate()” en el elemento HTML donde se quiera realizar la traducción.

Funciona de la siguiente manera; realiza un recorrido de la página a partir del elemento donde se haya llamado a la función “translate()” y va aplicando la función de traducción recursivamente a todos los subelementos. Para cada elemento a tratar, busca el carácter que delimita las fórmulas matemáticas y comienza a sustituir cada una de las entidades por su correspondiente en MathML. El proceso de sustitución de elementos de la página mediante funciones de JavaScript es posible gracias a la especificación DOM, como ya se ha dicho antes.

Puesto que MathML es un estándar, cabría esperar que cualquier navegador fuese capaz de interpretar las etiquetas que genera ASCIIMathML. Sin embargo, dada la relativa juventud de MathML, la necesidad de tener intérprete de Javascript y la forma en que evolucionan los navegadores se tienen las opciones: Firefox (con los tipos de letra adecuados en Linux, MacOS y Windows), Mozilla ((v. 1.3a) y Camino en Mac OS X.), e Internet Explorer ((v. 6.x) con el plugin correspondiente (MathPlayer) en Windows).

ASCIIMathML está disponible bajo GNU General Public License. La versión actual (Julio 2008) es la 2.0 cuya actualización es de septiembre del 2007, la versión con la que se ha trabajado es la 1.4.7.

Si se necesita usar símbolos matemáticos, ASCIIMathML tiene funciones estándares predefinidas, tiene una serie de símbolos y letras, operadores, llaves, paréntesis, elementos para generar espacio, alinear fórmulas, centrar, etc.

4.4 Interpretación de MathML en IE Y Mozilla

Una peculiaridad del ASCIIMathML es que genera distinto código para IE (Internet Explorer) y Mozilla, esto implica un problema bastante grave, pues las fórmulas introducidas desde IE no se visualizan correctamente en Mozilla y viceversa. El problema está en la interpretación que hace IE del MathML.

Mozilla interpreta código MathML de manera nativa, sin necesidad de añadir nada al navegador, sin embargo, Internet Explorer necesita el plugin MathPlayer, que veremos en el punto siguiente, para visualizarlo correctamente. Código genérico en MathML no se visualiza de manera estable en el tandem formado por Internet Explorer+MathPlayer, es decir, unas veces funciona y otras no.

Para que arranque el plugin MathPlayer y funcione de manera estable es necesario uno de estos dos métodos. Un primer método consiste en encapsular todo código en MathML como un objeto de tipo MathPlayer mediante el siguiente código:

```
<object id="mathplayer" classid="..."> ... </object>
```

Y el segundo método, que es el usado en ASCIIMathML, es incrustar un objeto MathPlayer vacío, declarar un espacio de nombre que use la implementación de ese objeto y finalmente referenciar todas las entidades de MathML hacia dicho espacio de nombres. Por ejemplo:

```
<object id="mathplayer" classid="..."></object>  
<?import namespace="mml" implementation="#mathplayer"?>  
<mml:math> ... </mml:math>
```

4.5 Math Player

Design Science MathPlayer es un motor de representación de alto rendimiento para el navegador IE 5.5 o posterior. Está disponible gratuitamente con el objetivo de acelerar la adopción de MathML por la comunidad científica. Se hace indispensable para los usuarios de Internet Explorer mientras éste no incluya un soporte completo de MathML. Actualmente solo Netscape, Mozilla y Amaya pueden representar MathML de forma directa.

MathML es una recomendación del W3C que permite la inclusión de una forma uniforme de símbolos y fórmulas matemáticas en páginas web. Si hablamos de navegadores Mozilla tiene soporte para parte del estándar desde hace tiempo. En cambio, IE no tiene soporte alguno para MathML, siendo necesario instalar programas de terceros, uno de los más conocidos es MathPlayer. El problema radica en que Mozilla sigue muy de cerca los estándares del W3C y para interpretar correctamente el código MathML necesita que éste esté dentro de una página en formato XHTML, lo que es bastante lógico si tenemos en cuenta que tanto XHTML como MathML son XML (no así HTML). En cambio MathPlayer (y probablemente otros programas) soporta el uso de MathML tanto dentro de XHTML como dentro de HTML.

5 Software Relacionado

A continuación se va a presentar el software existente (el que se ha analizado) que da soporte para MathML, buscando una posible utilidad para el desarrollo de nuestro proyecto. Se distinguirá entre software para edición de ecuaciones matemáticas y software de traducción a MathML.

5.1 Editores Web de Ecuaciones Matemáticas

Amaya W3C's web browser. Amaya es el navegador web del la W3C, que incluye un editor web que permite leer y modificar contenidos en la web de forma directa. Comenzó como un simple editor de HTML y hojas de estilo CSS, evolucionando para dar soporte para XML (incluido MathML). Se trata de un proyecto de código libre, escrito en C y disponible para las plataformas Windows, Unix y MacOS X. Podría plantearse hacer una reconversión de Amaya para utilizar su editor matemático a MathML. El código fuente de Amaya está escrito en C, un lenguaje no soportado por la web por lo que se debería hacer una traducción de sus métodos en otro lenguaje que sí tenga aplicaciones para la web. Los lenguajes que únicamente se pueden usar para esta traducción son lenguajes de no-scripting, por lo que las opciones serían Java y C#. C# es un lenguaje exclusivo de la plataforma .NET de Microsoft. Su uso implica ser propietario de una licencia de Visual Studio .NET y además su uso, a nivel de servidores, está restringido a la plataforma Windows. Por lo que la única opción sería realizar la traducción del código C de Amaya a Java. Hemos descartado esta opción. En principio, se desconoce si se podría realizar exitosamente por las librerías de las que podría estar haciendo uso el código. Pero creemos que podría ser una opción futura.

Design Science WebEQ . WebEQ ⁴(WebEQ™ Developers Suite) es una colección de herramientas y componentes Java para el procesado de MathML. Los componentes, cuyo uso no es gratuito, más importantes son los siguientes:

- WebEQ Editor: Es un editor gráfico de ecuaciones en MathML realizado íntegramente en Java.
- WebEQ Publisher: Es un procesador de documentos en numerosos formatos para la conversión de éstos en páginas web para facilitar su publicación. Publisher acepta tanto MathML como WebTex (similar a la parte matemática de LaTeX), y genera los mismos formatos que WebEQ Editor: MathML, imágenes y applets.
- WebEQ Equation Server: Realiza las mismas funciones que WebEQ Publisher pero a nivel de servidor. Estos procesos pueden ser realizados de distintas formas: desde la línea de comandos, desde un servidor web usando scripts CGI, ASP o un servlet Java.
- WebEQ Controls: Son applets para ver, editar, evaluar y comparar formulas matemáticas en MathML y WebTeX desde páginas web. Incluye los siguientes módulos: Input Control, Viewer Control, Graph Control, Evaluation Control

⁴ www.dessci.com/en/products/webeq/interactive/

Meditor⁵. Es un proyecto alojado en el repositorio de proyectos de código abierto SourceForge. Trata de proveer una interfaz de edición simbólica de ecuaciones, no de forma gráfica. Además, usa un lenguaje matemático propio, no es LaTeX, que es capaz de exportar a MathML. Lo único interesante de este proyecto es el motor de parsing hacia MathML implementado en Java.

Swift⁶. Editor de ecuaciones matemáticas hecho en java. Aunque las utilidades que presentaba podían servir a la aplicación; un editor de ecuaciones y un visor, se descartó utilizarlo primero porque no fue posible ponerse en contacto con el autor y segundo porque no hubo manera de adaptar ambos elementos al sistema.

MimeTeX. Parsea una expresión matemática escrita en LaTeX generando su correspondiente imagen gif. MimeTeX tiene un programa pequeño que no utiliza TeX ni sus fuentes, se trata de un CGI (Common Gateway Interface) que se puede colocar en el directorio cgi-bin/ de la aplicación, sin más dependencias. MimeTeX es muy fácil de instalar y de usar. Basta con colocar la etiqueta html en el documento donde se quiera ver la correspondiente expresión LaTeX. Por ejemplo:

```

```

5.2 Conversores de Latex a MathML

LaTeX2HTML⁷. Este software es un conversor de LaTeX a MathML realizado en Perl. El resultado generado es una página en HTML que contiene MathML, y no únicamente el código MathML resultante, por lo que haría falta un proceso posterior para aislar esta información.

TeX4ht⁸. Es un conversor de TeX/LaTeX a documentos XML para su publicación en Web. Es un ejecutable de código abierto. Su integración Web es difícil, al tratarse de un programa externo que debería usarse a través de CGI's. Su uso es altamente complicado, y a través de su documentación y tutoriales se duda si realmente realiza una conversión de LaTeX a MathML o simplemente renderiza GIFs o JPEGs que permitan la visualización de las fórmulas.

TtM⁹. Se trata de un conversor comercial de TeX a MathML escrito en C y disponible en Windows y Unix. Se trata de una herramienta de pago y no es de libre distribución.

⁵ [/jscl-meditor.sourceforge.net/](http://jscl-meditor.sourceforge.net/)

⁶ www.geocities.com/SiliconValley/Heights/5445/swift.html

⁷ <http://www.latex2html.org/>

⁸ <http://www.cse.ohio-state.edu/~gurari/TeX4ht/mn.html>

⁹ <http://hutchinson.belmont.ma.us/tth/mml/>

6 SHARPO: Sistema Hipermedia Adaptativo aplicado a la Resolución de Problemas Online

Tanto la resolución de problemas como la transferencia de conocimientos del educador al alumno, constituyen dos núcleos relacionados, esenciales en la investigación cognitiva y en la educación matemática, que han sido un laboratorio en la generación de desarrollos y aplicaciones. No es por ello casual que en las investigaciones sobre la aplicación de la informática a la enseñanza de las matemáticas se desarrollaran modelos específicos para su tratamiento.

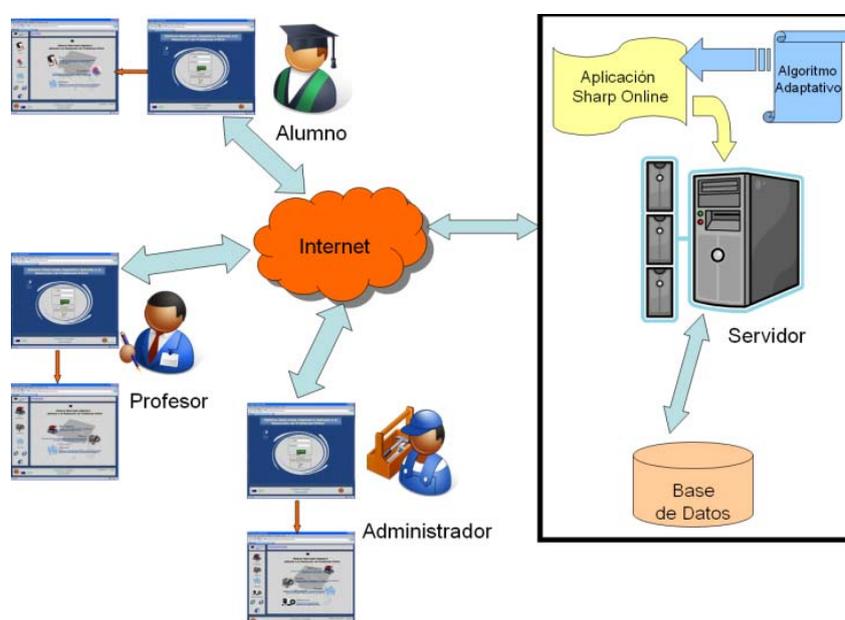


Fig. 4. Arquitectura de SHARPO

SHARPO, Sistema Hipermedia Adaptativo aplicado a la Resolución de Problemas Online, es una aplicación que hemos desarrollado [3], basada en el aprendizaje por entrenamiento en el área de la didáctica de las matemáticas basada en el modelo de Shöenfeld ([9],[10]). La arquitectura Web de dicha aplicación desarrollada, ver Figura 4, sustenta todas las tareas on-line. Asigna la funcionalidad mediante tres roles: profesor, alumno y administrador. Integra pautas de diseño del sistema hipermedia adaptativo, pero incrementando sus funcionalidades hacia tecnologías que permitan la formación online. De este modo el alumno puede reforzar sus conocimientos en el campo de las matemáticas a través de Internet, lo que permite implementar nuevas estrategias de aprendizaje basadas en el e-learning.

En el desarrollo de dicha aplicación aparecen serios problemas a la hora de la edición de símbolos matemáticos, en modo profesor, así como la visualización de

varias zonas, ver Fig. 5. Zona de edición y zona de visualización donde el texto de la pregunta, se muestra en el recuadro inferior tal y como será mostrado al alumno. El profesor dispone de ayuda contextual a la edición de fórmulas.

6.2 Envío y almacenamiento de datos

Una vez editada la pregunta por el profesor generador del contenido en su navegador el proceso de almacenamiento en la base de datos del sistema para guardar la pregunta es el que indica la Fig. 6.

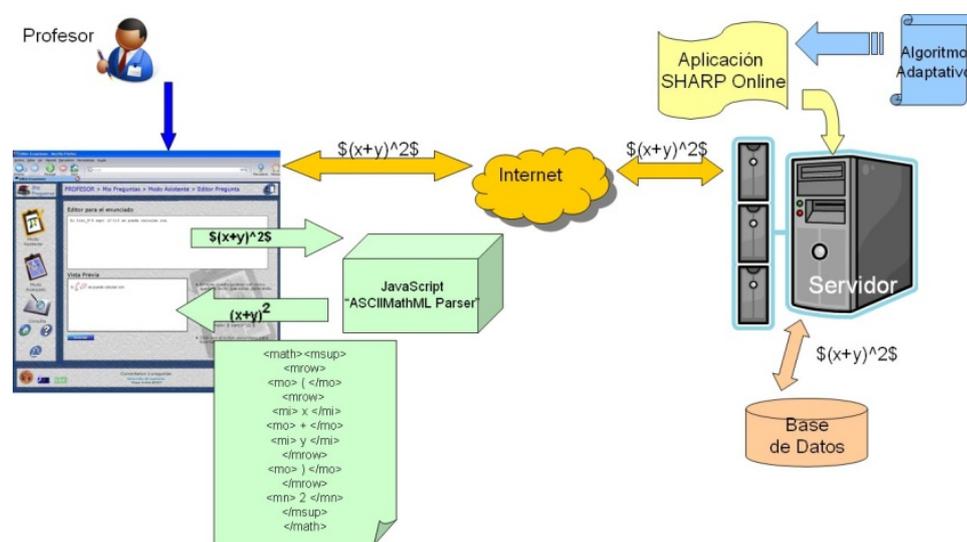


Fig. 6. Capas de acceso y visualización para la notación matemática Web

Cada fórmula generada es enviada a través de la red en el formato con el código ASCII en el que se ha escrito. La aplicación alojada en el servidor marcará dicha fórmula como “elemento matemático” y lo asociará con la opción de la pregunta donde haya sido creada. Toda la información será almacenada en la base de datos con el código ASCII específico. De este modo cuando se realice la tarea inversa, la de visualización en la generación del test al alumno, el sistema obtendrá los datos de la pregunta para que se carguen en el navegador del cliente y el parser implementado mostrará dichos elementos en su formato en modo visualización.

7 Conclusiones

A lo largo del artículo se han descrito las limitaciones de la edición y visualización de símbolos matemáticos en Web. El sistema de ecuaciones en imágenes que prolifera en Web es inadecuado debido a que la información matemática contenida en las imágenes no es útil para realizar búsquedas, indexar o reutilizar en otras aplicaciones.

Realizamos una revisión de los lenguajes y estándares existentes para la edición digital de fórmulas matemáticas y software relacionado que permita su soporte en Web.

Finalmente presentamos nuestro sistema de soporte a la docencia en línea, SHARPO, que permite a los alumnos entrenarse en los métodos resolutivos de problemas matemáticos. La arquitectura Web de la aplicación es modular y abarca funcionalidades para la gestión docente y la creación de contenidos. Mostramos los módulos de edición y visualización de SHARPO basados en ASCIIMathML, un JavaScript que realiza la conversión de ASCII, a MathML. MathML como lenguaje de marcado para expresiones matemáticas permite que las matemáticas sean servidas, recibidas y procesadas en la Web al igual que lo es el texto HTML, de manera que es el elegido en el desarrollo. La aplicación, debido a su diseño modular, nos permitirá en un futuro próximo seguir incrementando su desarrollo. Se plantea su adaptación a las herramientas basadas en utilidades de escritorio de acuerdo a la Web 2.0 y su adaptación a contenidos basados en los estándares de objetos de aprendizaje.

8 Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencias así como por el proyecto FEDER Keops (TSI2005-00960).

References

1. Mandler, G.: "Affect and learning: Causes and consequences of emotional interaction." En D. B. McLeod & V. M. Adams (Eds.), *Affect and mathematical problem solving: A new perspective* (pp. 3 19). New York: Springer (1989).
2. Lepper, M. R., y Malone, T W.: "Intrinsic motivation and instructional effectiveness in computer-based education." En R. E. Snow & M. J. Farr (Eds.), *Aptitude, learning and instruction: Conative and affective process analysis* (Vol. 3, pp. 255 286). Hillsdale, NJ: Lawrence Erlbaum Associates (1987).
3. Raquel Rodríguez, Ana-Belén Gil, Francisco J. García, Ricardo López. "SHARP Online: Sistema Hipermedia Adaptativo Aplicado a la Resolución de Problemas Matemáticos". *Actas del IX Congreso Internacional Interacción'2008*, pp. 271-284. ISBN 978-84-691-3871-7. Albacete, 9-11 de Junio de 2008. (2008)
4. Knuth, D.: "The TeXbook." Addison-Wesley, ISBN 0-201-13447-0, 1984
5. Lamport, L.: "LaTeX: A document preparation system", Addison-Wesley, ISBN 0-201-52983-1 (1994).
6. MathML 1.0: <http://www.w3.org/TR/1998/REC-MathML-19980407/>
7. MathML 3.0: <http://www.w3.org/TR/MathML3/>
8. MathML y Css: <http://www.w3.org/TR/mathml-for-css/>
9. Schoenfeld, A.: "A. Mathematical problem solving", Academic Press, New York (1985).
10. Schoenfeld, A.: "Metacognitive and epistemological issues in mathematical understanding", en *Teaching and learning mathematical problem solving: Multiple research perspectives* (pp. 361 380). Hillsdale, New Jersey (1985).