Extensible Markup Language (XML): Una solución para modelar documentos y sus interrelaciones basada en la semántica de la información

María Mercedes Martínez González
Universidad de Valladolid

0.1 Resumen

Se discuten las características y ventajas del estándar de formateo de documentos conocido como *Extended Markup Language* (XML). Entre las mismas destacan la sencillez y legibilidad de los documentos, la separación entre formato sintáctico y presentación gráfica, y la potencia de sus hiperenlaces. Las características comentadas se ilustran mediante una aplicación XML para el tratamiento y recuperación de información jurídica.

Palabras clave: Lenguajes de etiquetado de documentos. Extended Markup Language (XML). Hiperenlaces. Documentación jurídica.

0.2 Abstract

The characteristics and advantages of the emerging document formatting standard acknowledged as Extended Markup Language (XML) are discussed. Among them, the legibility and clarity of the documents, the separation between the syntactic structure and the graphical presentation and the powerfulness of its hyperlink system are analyzed. The characteristics are illustrated with a XML application oriented to the treatment and retrieval of juridical documentation.

Keywords: Document markup languages. Extended Markup Language (XML). Hyperlinks. Juridical documentation.

1. De HTML a XML

Los documentos digitales se pueden encontrar en numerosos formatos, dependientes habitualmente de la aplicación de edición utilizada para crearlos. Estos formatos se caracterizan en general por la utilización de códigos de control que sirven para marcar los apartados de los documentos y caracterizar los atributos de formato del texto.

Con la Web se popularizó una nueva filosofía a la hora de almacenar los documentos, cuyo principio básico es que toda la información concerniente al documento es textual (incluidos los atributos relacionados con el formato, divisiones en apartados, etc.). De este modo, los documentos son independientes de las herramientas de edición usadas para crearlos y visualizarlos. Resultado de esta orientación fue la definición de un lenguaje para la descripción de documentos, conocido como *Hypertext Markup Language* (HTML).

Se pueden crear documentos HTML con el más sencillo o el más avanzado de los editores de texto; además, los documentos HTML implementan el concepto de hipertexto (Conklin, 1987), permitiendo así a los usuarios navegar de un documento a otro a través de los hiperenlaces.

La independencia de la herramienta con la se crean los documentos unida a las capacidades hipertexto han hecho de HTML el "formato" más aceptado actualmente para difundir documentos.

Sin embargo, esta popularidad dio lugar a la aparición de documentos HTML "de mala calidad" (1), cuyo procesamiento automático es complicado; en algunos casos las aplicaciones informáticas son incapaces de procesar algunos de estos documentos, degenerando en errores que abortan la ejecución. La figura 1 muestra uno de estos documentos, donde el cierre de un final de párrafo que no está abierto previamente (la penúltima línea del fichero contiene un cierre de párrafo, </p>, que no se corresponde con ninguna apertura anterior) da lugar a errores irrecuperables en algunas herramientas destinadas a la indexación de documentos. Como da a entender este ejemplo, en gran medida, la causa de la

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<doc>
<articulo id="a1"><title>Artículo Primero. </title>
<p>El referendum en sus distintas modalidades, se celebrará de
acuerdo con las condiciones y procedimientos regulados en la
presente Ley Orgánica.</p>
</articulo>
<articulo id="a2"><title>Artículo Segundo. </title>
<p>Uno. La autorización para la convocatoria de consultas populares
por vía de referendum en cualquiera de sus modalidades, es competencia
exclusiva del Estado.</p>
</articulo>
</doc>
```

Fig. 1. Etiquetado incorrecto en un documento —cierre de una etiqueta que no fue abierta previamente (último </p>)—, que provoca errores en la ejecución de las aplicaciones informáticas.

existencia de estos documentos de mala calidad es la flexibilidad sintáctica de HTML.

A la vez que HTML se difunde, sus limitaciones como formato "general" para manipular la información se hacen más evidentes. Los usuarios exigen cada vez más de las herramientas que manipulan información. Tal es el caso de las herramientas de búsqueda, que compiten por mejorar las funcionalidades que ofrecen a sus usuarios. Ya no es suficiente con localizar los documentos que contienen una cierta cadena; se requieren búsquedas que tengan en cuenta la estructura (divisiones internas e inclusiones entre ellas) de los documentos. Como hemos visto, con documentos mal etiquetados, esto no es posible.

También se demanda la posibilidad de anotar documentos remotos, creados por otros autores (sobre los cuales no se tiene permiso de escritura), construir catálogos de información en base a búsquedas efectuadas sobre documentos propios y de otros autores, etc.

En este trabajo se presentan algunas de las soluciones que ofrece el Extended Markup Language (XML) para resolver las limitaciones de HTML. La exposición se centra en aquellos aspectos de XML que han sido de utilidad en el supuesto de trabajo presentado en el apartado 7.2. Existen más estándares y características de XML que, por no haber sido utilizados en esta aplicación, no se contemplan en este trabajo. Por otro lado, tampoco se pretende hacer una descripción exhaustiva de estos aspectos, sino más bien una breve introducción, suficiente para justificar la elección de este estándar. Explicaciones más detalladas sobre XML y sus estándares asociados se pueden encontrar en las especificaciones que se referencian a lo largo del artículo, en obras sobre XML —como (Rusty-Harold, 1999)—, o en las propias páginas que el W3C (2) dedica a XML.

Este artículo se divide en dos bloques principales:

1. *Una descripción de XML y los estándares asociados* (secciones 2 a 6). A su vez, esta parte se organiza del siguiente modo:
   a) *Presentación de XML*: cómo son los documentos XML, cuáles so las principales características de este estándar, qué es y para qué sirve la DTD, cómo dar formato a los documentos XML y cómo transformar y crear nuevos documentos utilizando el lenguaje de transformación XSLT.
   b) *XML y la manipulación de relaciones*. La sección 6 marca el comienzo de esta parte. Las relaciones se modelan como enlaces; los enlaces XML se conocen como xlinks, y para aprovechar toda la potencia de estos enlaces es necesario utilizar los estándares asociados XPointer y XPath.

2. En la segunda parte, se muestra cómo estos estándares han facilitado la obtención de servicios avanzados en el tratamiento de documentos (sección 7).

2. Modelado de datos con XML

XML (Extensible Markup Language) (XML, 2000) es un lenguaje de etiquetado de documentos (se introducen etiquetas entre el texto). XML fue publicado por el W3C como una recomendación estable (3) en 1998. Se trata de una simplificación de SGML (Standard Generalised Markup Language) (SGML, 1986), cuya aplicación más conocida es HTML.

XML modela los documentos como conjuntos de elementos que contienen cadenas de caracteres. El principio y el final de cada elemento se delimita con etiquetas.

La idea fundamental detrás de XML consiste en marcar o etiquetar la información, de modo que cada porción (elemento) del documento se delimita por una etiqueta de comienzo seguida de la correspondiente etiqueta de cierre que indica el final de ese elemento (similar, pero con una barra inclinada -'/'- que la diferencia de la marca de comienzo de elemento).

Las etiquetas son los textos delimitados entre ángulos (<...>); el resto es el contenido del documento. Cada elemento tiene un tipo (nombre de la etiqueta) y un valor (lo que hay entre las etiquetas). Cada elemento puede contener una combinación de texto y otros elementos.

2.1. Cómo es un documento XML

En el ejemplo de la figura 2 se muestra un documento, que contiene datos acerca de un libro. El comienzo del documento se marca con la etiqueta <libro>; el elemento libro es el más externo, y contiene todos los demás (el texto que le precede son processing-instructions, útiles para las aplicaciones informáticas, pero no significativas para los usuarios). El principio del elemento libro se delimita con la marca <libro>; todo lo que se encuentra desde esta marca hasta la marca </libro> forma parte del elemento. El texto de este elemento es “Referencia para la asignatura de Bases de Datos”. Pero además incluye en su interior el elemento ‘titulo’, dos elementos ‘autor’, un elemento ‘edicion’, ‘editorial’ y ‘despacho’. De forma similar, el segundo elemento ‘autor’ contiene el texto “Ramez” y un elemento ‘apellido’, que a su vez contiene el texto “Elmasri”. Dos elementos pueden ser del mismo tipo (en el ejemplo, hay dos elementos ‘autor’). Los elementos pueden tener atributos que los caracterizan; el elemento ‘despacho’ tiene un atributo (catalogado) que indica si el libro está o no catalogado.

2.2. Propiedades relevantes de XML

Delimitar claramente los elementos es fundamental para que sea posible construir herramientas capaces de analizar los documentos (*parsers*), seleccionar elementos del documento, indexar los documentos, etc., que de otro modo serían incapaces de detectar los límites de cada elemento. Por eso, una de las bases de XML es la definición de unas reglas sintácticas simples pero estrictas, que garantizan que cualquier documento XML puede ser procesado por herramientas sencillas, construidas para trabajar con documentos XML genéricos. A los documentos que cumplen estas reglas sintácticas se les dice documentos "bien formados" (*well-formed*). Estas características son especialmente relevantes —y se espera que lo sean aún más— en un entorno tan popular y en auge como Internet.

Pero lo realmente relevante de XML es que es permite etiquetar los documentos teniendo en cuenta la semántica de la información. Dicho de otro modo, cada diseñador de información puede crear las etiquetas o conjuntos de etiquetas que considere que mejor describen los elementos que componen sus documentos. A esta propiedad consistente en permitir la creación de nuevas etiquetas se la denomina *extensibilidad*. De esta propiedad se deriva otra propiedad adicional de los documentos XML: su *legibilidad*. Dado un documento XML etiquetado con marcas suficientemente representativas, cualquier usuario —con o sin conocimientos de XML —es capaz de reconocer los distintos elementos del documento, las reglas de inclusión entre esos elementos, y qué representa cada uno de los elementos. En el ejemplo del libro que hemos visto, es sencillo deducir que la información que se está modelando para cada libro son datos tales como su autor o autores, precio, y otros.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- libro.xml -->
<?xml:stylesheet type="text/css" href="libro.css" ?>

<!DOCTYPE libro SYSTEM "libro.dtd">
<libro> Referencia para la asignatura de Bases de Datos.
  <titulo>Sistemas de Bases de Datos. Conceptos Fundamentales.</titulo>
  <autor>Ramez<lastname>Elmasri</lastname></autor>
  <autor>Shamkant B.<lastname>Navathe</lastname></autor>
  <precio moneda="ESP">?</precio>
  <edicion>Segunda</edicion>
  <editorial>Addison-Wesley Iberoamericana</editorial>
  <despacho catalogado="si">Mercedes</despacho>
</libro>
```

Fig. 2. Datos para los libros modelados con XML.

2.3. XML y HTML

La importancia de estas propiedades resalta si comparamos XML con HTML (Tabla 1). HTML también se basa en la idea de etiquetar los elementos. Sin embargo, existen dos diferencias importantes entre ambos. La primera es que en HTML las reglas sintácticas son menos estrictas. Por ejemplo, en un documento HTML puede haber etiquetas de apertura para elementos que nunca estén cerrados (un caso muy habitual son las páginas que contienen una etiqueta <html>, pero no incluyen en ningún punto la etiqueta (</html>). Esta primera diferencia provoca que el tratamiento automático de los documentos HTML sea mucho más difícil que el de los documentos XML (4), y es la culpable de que existan pocas herramientas que lo permitan, o de que los motores de búsqueda en Internet ofrezcan muy limitadas posibilidades de búsqueda por campos en las páginas. Este problema se evita con XML, ya que cualquier documento que se diga XML debe cumplir unas reglas sintácticas mínimas; es decir, debe estar "bien formado".

La segunda diferencia reside precisamente en la propiedad de extensibilidad de XML: en HTML el conjunto de etiquetas que pueden aparecer en el documento está restringido al que define el propio estándar y se trata de un conjunto de etiquetas que modelan características relativas al formato del documento (por ejemplo, la etiqueta <h1> indica que el estilo que se debe utilizar para visualizar el fragmento de texto marcado es Heading 1). Sin embargo, los documentos XML —si bien pueden estar etiquetados en base a criterios de formato o presentación— suelen disociar la semántica de los documentos de su presentación, de modo que el etiquetado dice qué contienen los elementos en vez de cómo se presentan dichos elementos o qué hacer con ellos. Esto es lo que ocurre en el ejemplo del libro de la figura 2: sabemos qué información guardamos, pero nos despreocupamos de cómo se visualizará, imprimirá, etc. En realidad, la calidad del etiquetado depende del creador o autor del documento.

HTML	XML
Texto etiquetado	Texto etiquetado
Etiquetas predefinidas	Extensible (etiquetas definidas por los usuarios)
Las etiquetas dicen "cómo" formatear el elemento	Etiquetado descriptivo: Las etiquetas dicen "qué" contiene el elemento
Sintaxis relajada: difícil de tratar por las aplicaciones informáticas	Sintaxis estricta: mejor para las aplicaciones informáticas
Enlaces que permiten navegar por los documentos	Enlaces más potentes

Tabla 1. Tabla comparativa de HTML y XML.

```
<html>
  <p>Referencia para la asignatura de Bases de Datos.
   <h2>Sistemas de Bases de Datos. Conceptos Fundamentales.</h2>
   <p><h3>Ramez <em>Elmasri</em>, Shamkant B. <em>Navathe</em>.</h3></p>
   <p>Segunda edición.</p>
   <p><b>Despacho: </b>Mercedes. Catalogado.</p>
  </p>
</html>
```

Fig. 3. Código de la propuesta en HTML para el libro de la figura 2 (libro1.html)

Las figuras 3 y 4 muestran dos versiones HTML de los mismos datos del libro cuya codificación con XML aparece en la figura 2. En este caso se aprecian varios problemas. Primero, sin una indicación sobre la correspondencia entre el nombre de la etiqueta y el significado de cada uno es imposible saber qué elemento modela la información sobre el título, cuál sobre los autores, etc. (5). Segundo, aún habiendo superado este obstáculo, cualquier decisión —por pequeña que sea— que suponga cambiar el aspecto de alguno de los elementos, implica modificar todos los programas que tratan estos documentos. Por ejemplo, cada vez que se modificase la etiqueta que delimita el título, se debería modificar la herramienta que realiza búsquedas en el campo título. Normalmente, este tipo de implicaciones suponen finalmente, bien renunciar a algunas funcionalidades de las aplicaciones que tratan los documentos, bien restringir las modificaciones referentes al formato con el objetivo de no dificultar el mantenimiento de las demás aplicaciones.

Separar la información sobre semántica de los aspectos de presentación puede resultar mucho más útil de lo que en principio puede parecer. En nuestro ejemplo, si se quieren visualizar los datos acerca del libro que hemos visto en un navegador, pero también se quieren imprimir y mostrarlos en el monitor de un teléfono móvil, se puede asociar para cada caso un conjunto de reglas que permita presentarlos de modo distinto en cada uno de estos medios, adaptando el formato de salida en cada uno de los casos a las características del dispositivo de salida (navegador, impresora, móvil).

```
<html>
  <p>Referencia para la asignatura de Bases de Datos.
   <h2>Sistemas de Bases de Datos. Conceptos Fundamentales.</h2>
   <p><h3>Ramez <em>Elmasri</em>, Shamkant B. <em>Navathe</em>.</h3></p>
   <p>Segunda edición.</p>
   <p><b>Despacho: </b>Mercedes. Catalogado.</p>
  </p>
</html>
```

Fig. 4. Código de la propuesta en HTML para el libro de la figura 2 (libro2.html)

Finalmente, como resultado de todas estas propiedades, es posible afirmar que XML aporta *interoperabilidad* en los datos. La interoperabilidad sintáctica está garantizada por las propias restricciones sintácticas que definen los documentos XML. Los documentos XML creados por una empresa o proveedor pueden ser procesados por cualquier parser XML (6), que informará de cualquier error sintáctico presente en el documento. En cuanto a la interoperabilidad semántica, cada comunidad de usuarios puede definir sus propios dominios de nombres para las etiquetas. Dadas las definiciones que cada comunidad de usuarios adopta en sus documentos, estos analizadores genéricos son capaces de validar la conformidad del documento respecto a estas reglas. La única restricción es que se le proporcione la definición de las reglas, además del documento. Resumiendo, las propiedades relevantes de XML son:

1. Sencillez
2. Legibilidad
3. Extensibilidad
4. Interoperabilidad

3. La DTD

Los documentos XML pueden agruparse en clases. Cada clase se caracteriza por el conjunto de elementos (etiquetas) que pueden aparecer en los documentos de la clase y las reglas de inclusión entre ellos. Por ejemplo, todos los documentos que contengan datos sobre los libros de nuestro departamento constarán de elementos ‘book, que podrán contener la información sobre autores, etc.; un elemento ‘autor’ puede aparecer en el interior de un elemento ‘book’, pero no en el interior de un elemento ‘title’, etc.

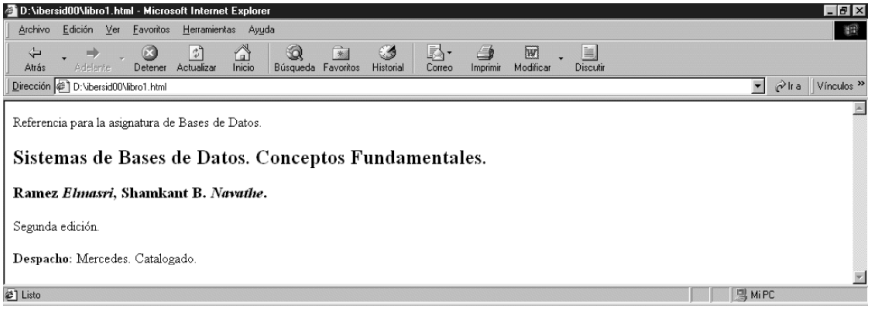

Fig. 5. Visualización en un navegador de libro2.html (Fig.3)

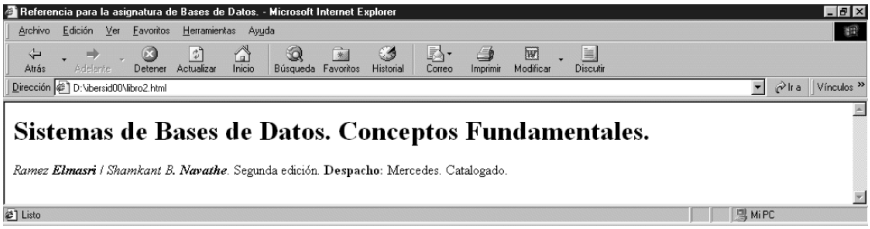

Fig. 6. Visualización en un navegador de libro2.html (Fig.4)

La forma de expresar las reglas aplicables a todos los documentos de una clase es definiendo una DTD (*Document Type Definition*). Un documento que se ajusta a las convenciones expresadas en una DTD es un documento válido respecto a esa DTD. La directiva DOCTYPE al principio de un documento XML indica cuál es la DTD a la cual se ajusta el documento. Esta directiva indica al procesador XML que el documento no solo debe estar conforme a las reglas sintácticas inherentes a XML (bien formado), sino que también debe ajustarse a las reglas expresadas en la DTD (válido). Así, el elemento 'book' que aparece en la figura 6 da lugar a un documento que está bien formado, pero que no es válido al contrastarlo con la DTD de la figura 7 (el elemento titulo contiene un elemento autor, lo cual, según indica la DTD, no está permitido).

3.1. Sintaxis

Las DTD se expresan con una sintaxis particular, que se describe brevemente sobre la DTD asociada a los documentos de tipo libro (figura 7). La declaración de reglas comienza en la tercera línea. Se trata de una declaración de elemento, que establece restricciones sobre el contenido de los elementos del tipo 'libro'. Las convenciones sintácticas establecen que las comas se usan para las

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- libro.dtd -->
<!ELEMENT libro (titulo, autor+,  edicion ?,  editorial ?, despacho?)>
<!ELEMENT titulo (CDATA)>
<!ELEMENT autor (CDATA|nombre|apellido)*>
<!ELEMENT edicion (CDATA)>
<!ELEMENT editorial (CDATA)>
<!ELEMENT despacho (CDATA)>
<!ATTLIST despacho catalogado CDATA "sí">
```

Fig. 7. DTD para la clase ``libro''.

enumeraciones, los paréntesis para agrupar, la barra vertical es una disyunción y los operadores '?', '*' y '+' indican cero o una, cero o varias, y una o más apariciones del elemento o grupo que precede inmediatamente al operador. Según esto, un elemento 'libro' contiene una secuencia formada por un elemento del tipo 'título', seguida de uno o más 'autores', el 'precio', y opcionalmente, la 'edición' y 'editorial'. En la segunda línea se indica que el 'título' está formado por una secuencia de caracteres (CDATA). El campo 'autor' está formado por cualquier combinación de texto y elementos nombre, apellido, en cualquier orden. Los valores del 'precio', 'edición' y 'editorial' serán cadenas de caracteres. Los elementos de tipo 'despacho' tienen un atributo que indica si el libro ya ha sido catalogado —las declaraciones de atributos de cada elemento se hacen con <!ATTLIST ...>. En este caso el atributo tendrá como valor una cadena de caracteres y —si no indica lo contrario en el documento— se considera que el libro sí está catalogado.

3.2. Ventajas y limitaciones de la DTD

La DTD es muy útil. La primera utilidad es obvia: permite verificar la corrección de un documento de modo automático; por ejemplo, se pueden validar los catálogos provenientes de varias fuentes para comprobar que todos han sido creados con los elementos (etiquetas) adecuados. Pero además aporta otras ventajas:

1. Disponer de la DTD posibilita crear plantillas de formato aplicables a todos los documentos de la misma clase. No es necesario dar formato a cada documento, es suficiente con disponer de la DTD asociada y una hoja de estilo que contiene las reglas de formato para todos los documentos de esa clase. De este modo, las modificaciones sobre los aspectos

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- libro.xml -->
<?xml:stylesheet type="text/css" href="libro.css" ?>

<!DOCTYPE libro SYSTEM "libro.dtd">
<libro> Referencia para la asignatura de Bases de Datos.
  <titulo>Sistemas de Bases de Datos. Conceptos Fundamentales.
  <autor>Ramez<lastname>Elmasri</lastname></autor>
  <autor>Shamkant B.<lastname>Navathe</lastname></autor></titulo>
  <precio moneda="ESP">?</precio>
  <edicion>Segunda</edicion>
  <editorial>Addison-Wesley Iberoamericana</editorial>
  <despacho catalogado="si">Mercedes</despacho>
</libro>
```

Fig. 8. Documento no válido respecto a la DTD de la figura 7.

relacionados con la presentación de los documentos se simplifican al máximo, ya que cada modificación es aplicable a todos los documentos de una clase.

Un ejemplo para los documentos de la clase libro aparece en el apartado 4.1.

2. Conocida una DTD es posible definir un conjunto de reglas que permitan transformar todos los documentos que se ajustan a ella en los documentos equivalentes, conformes a otra DTD. Un caso muy sencillo lo ilustra sobre el ejemplo de los libros. Partiendo de la DTD de la figura 5 se puede definir un conjunto de reglas que permita obtener los catálogos equivalentes, pero con las etiquetas expresadas en inglés. También es posible crear nuevos documentos. Este proceso recibe el nombre de transformación y se comenta en el apartado 4.2.

A pesar de su valía, en su estado actual, la DTD tiene limitada su capacidad expresiva. Por ejemplo, no se puede restringir los posibles valores de un elemento o atributo a un cierto rango (en nuestro ejemplo de libro, no es posible indicar que el precio siempre debe ser mayor que 0). Esta limitación y otras que se han detectado con las DTD son las que se intenta resolver con nuevas propuestas que sustituyan a la DTD. Todas estas propuestas son aún objeto de debate y se pueden encontrar agrupadas bajo el epígrafe XSchema en la página correspondiente del W3C (7).

3.3. Algunas DTD relevantes

Cada usuario o comunidad de usuarios puede definir sus propias DTD, adaptadas a la información que manipula. Existen algunas DTD que son más o menos populares y que en algunos casos se han convertido prácticamente en estándares dentro de un cierto contexto. La más popular de todas es HTML, que conocen todos los creadores y usuarios de páginas Web —incluso aunque nunca hayan visto la DTD. Esta DTD es la que impone, por ejemplo, que un campo meta no puede aparecer en el interior del body de una página HTML. Esta información que proporciona la DTD la utilizan a su vez los navegadores, que solo visualizan el contenido del body de la página, utilizando el resto de campos como información adicional útil para otros fines.

La TEI (*Text Encoding Intitiative*) (Sperber-McQueen, 1994) es una gran DTD concebida para representar documentos de carácter general, aunque se ha utilizado sobre todo con documentos literarios (colecciones de manuscritos y textos literarios).

Otra DTD concebida para ser utilizada con documentos, de carácter técnico en este caso, es la DocBook (Walsh, 1999).

4. Hojas de estilo. Formato y transformaciones

Disponer de la DTD de una clase permite asociar reglas para dar formato a todos los documentos de esa clase. Las hojas de estilo son conjuntos de reglas que se adjuntan a los documentos XML, y que se aplican a éstos en el momento de su presentación al usuario.

Hay dos tipos de hojas de estilo:

1. Las que contienen reglas sobre el formato o presentación del documento. Dentro de las primeras se incluyen las hojas CSS (*Cascading Style Sheets*).
2. Existen otras hojas de estilo, escritas con el lenguaje de transformación XSLT (*XSL Transformations*) (XSLT, 1999) que permite obtener transformar documentos XML en otros documentos XML o HTML.

En consecuencia, a la hora de presentar un documento XML en un Navegador (8) es posible adjuntar a éste una hoja CSS que el navegador interpreta para darle formato, o transformarlo en una página HTML antes de enviarlo al navegador.

4.1. Hojas de estilo CSS

En una hoja de estilo CSS encontraremos, para cada elemento de una DTD, un bloque donde se especifican las características de presentación para ese tipo de elemento, tales como tamaño de letra, tipo de fuente, etc.

En el ejemplo de la figura 9 se cualifican los elementos que hemos visto en la DTD de la figura 7. A cada elemento le siguen sus propiedades, encerradas entre llaves. Para cada atributo se expresa el nombre y su valor separados por dos puntos. Por ejemplo, los títulos deben mostrarse en negrita (bold).

El resultado que se obtiene cuando el navegador aplica esta hoja de estilo al documento de la figura 2 es el que se muestra en la figura 10. Para que el navegador sepa cuál es la hoja de estilo que debe aplicar, el documento debe incluir una directiva similar a la que se utiliza para la DTD. En el ejemplo de la figura 2, se trata de la directiva <?xml:stylesheet type="text/css" href="libro.css" ?>.

4.2. Transformando documentos XML

En algunos casos no es suficiente con disponer de una serie de información y ser capaz de presentarla de varios modos posibles. Por ejemplo, a partir de la colección de datos sobre los libros disponibles se puede crear un nuevo documento donde aparezcan únicamente los títulos y autores listados por fecha de publicación. También es posible realizar transformaciones más sencillas sobre los datos como, por ejemplo, variaciones en el orden de colocación: los datos sobre un libro se pueden visualizar con el título en primer lugar seguido del autor y demás campos, o bien que sea el autor el primer campo que veamos ya que es

```
libro
{   display : block ;  font–family:Verdana;   font–size:12pt ; }

titulo
{  display : block ;  margin–top:1em; font_weight:bold; }

autor
{   display : block ;  background–color:teal ;
   font–style :  italic  ;   color : white ; }

edicion
{   display : inline ; }

editorial
{   display : none; }

despacho
{   display : inline ; }
```

Fig. 9. Hoja de estilo CSS para documentos de la clase libro.

aquél en el que estamos interesados. Por último, se pueden realizar transformaciones entre DTD, de modo que el mismo texto se encuentre etiquetado de forma distinta. Por ejemplo, para intercambiar nuestro catálogo de libros con algún colega que no conozca el español puede ser interesante "traducir" las etiquetas del documento a una DTD conocida por ambos. En estos casos se trata de "generar" nuevos documentos a partir de los documentos disponibles, lo cual no es posible con hojas CSS.

Es necesario realizar una transformación, tal que, a partir de un documento XML, se obtiene un nuevo documento. Normalmente, éste también es un documento XML, aunque también es posible obtener otros formatos, como texto o HTML.

Fig. 10. Visualización en un navegador de un documento XML (figura 2) con una hoja

4.3. Transformaciones de documentos en la Web

En el momento actual, una de las aplicaciones más importantes de las transformaciones de documentos se encuentra en la propia Web. Dado que actualmente la mayor parte de los navegadores no soportan aún XML, para ofrecer datos XML a través de Internet de forma que cualquier usuario sea capaz de visualizar con su navegador, los proveedores de información han de encargarse de generar páginas HTML a partir de la fuente en XML. Esto se puede conseguir de dos maneras:

1. Transformando los documentos XML en HTML antes de depositarlos en el servidor Web.
2. Instalando un servidor Web capaz de manipular XML (a estos servidores se les denomina servidores XML o servidores *XML-enabled*).

La solución de los servidores XML está en franco desarrollo y es de esperar que en un futuro próximo la mayor parte de los servidores Web sean servidores de este tipo. Algunos ejemplos de servidores cuyos prototipos ya están disponibles son Cocoon —implementado sobre Apache (9)— y XML-enabler de IBM. Vamos a centrarnos en cómo consiguen estos servidores generar las páginas HTML que mandan a los clientes.

El paso de XML a HTML consiste en una transformación, como ya hemos dicho. Las reglas que indican al servidor cómo realizar la transformación vienen dadas en una hoja de estilo, si bien su aspecto tiene poco que ver con el de las hojas CSS. Al lenguaje en el cual están escritas estas hojas de estilo se le denomina XSLT (*XSL Transformations*), y es un subconjunto de un lenguaje pensado para hojas de estilo XML conocido como XSL (*XML Stylesheets Language*). Así pues, a estas hojas de estilo se les dice hojas XSL, diferenciándolas así de las hojas CSS. En el momento de atender una solicitud de una página HTML, el servidor tomará el documento XML, la hoja XSL y aplicará la transformación que dará como resultado la página HTML.

4.4 Cómo describir transformaciones con XSLT

Las hojas de estilo XSLT consisten en un conjunto de reglas o plantillas (*templates*). Cada regla contiene los siguientes componentes básicos:

1. Un camino que selecciona un elemento dentro del documento XML. Representa el recorrido a través de las etiquetas de los elementos que contienen al que nos interesa, desde el más externo, hacia el interior. Dicho camino se expresa utilizando la sintaxis de Xpath (XML Path Language) (XPath, 1999) (10). En el ejemplo de la figura 11, la cadena XPath libro/titulo selecciona el título de un libro. Es decir, encontraremos el titulo dentro del elemento libro.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<xsl: stylesheet  xmlns: xsl="http://www.w3.org/1999/XSL/Transform"
                  xmlns: xlink ="http://www.w3.org/1999/xlink"
                  version ="1.0">



<!-- REGLA 1. Aplicar al nodo raíz. Continuar procesando los  hijos. -->
<xsl:template match="/">
        <html><head><title>Nuestro pequeño catálogo.</title></head>
          <body>
            <xsl:apply-templates/>
          </body>
        </html>
</xsl:template>

<!-- REGLA 2. Aplicar a todos los elementos libro. -->
<xsl:template match="libro">
            Título :   <xsl:apply-templates select="titulo"/>
            <xsl:apply-templates select="autor"/>
</xsl:template>



<!-- REGLA 2. Aplicar a los elementos 'titulo ' cuyo padre es ' libro '.
              -->
<xsl:template match="libro/titulo">
        <b><xsl:apply-templates/></b>
</xsl:template>

<!-- REGLA 3. Aplicar a los elementos 'autor' cuyo padre es ' libro '.
              Procesar recursivamente sus hijos, que aparecerán en negrita ('b') -->
<xsl:template match="libro/autor">
        <br></br> Autor: <em><xsl:apply-templates/></em>
</xsl:template>


</xsl: stylesheet >
```

Fig. 11. Hoja de estilo XSL para obtener un listado HTML de los libros, aplicable a documentos como el de la figura 2.

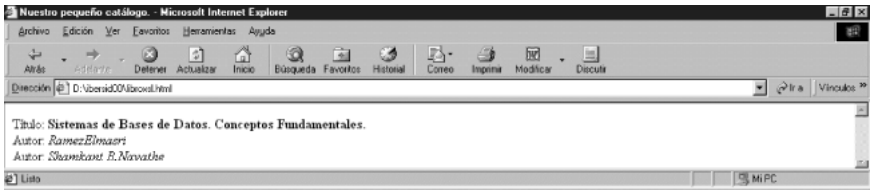

Fig. 12. Página HTML resultado de aplicar la transformación de la figura 11 al documento XML de la figura 2.

2. El texto que se va a escribir en la salida en el lugar del elemento seleccionado.
3. Indicaciones para seguir procesando el documento.

La hoja de estilo de la figura 11 genera un listado de los libros en el cual para cada libro se muestran el título y autores. La primera regla se aplica al comenzar a trabajar con el documento (match='/'), y crea la estructura básica del documento HTML: <html><head>...</head>. Es interesante fijarse en que se crea un elemento cabecera (head), que no existía en el documento XML de entrada. Además, se indica que se debe seguir con la transformación del resto del documento (<xsl:apply-templates/>).

La segunda regla se aplica a los títulos (match="book/title") y escribe el título en negrita (). La tercera regla escribe los autores (match="book/author") en itálica (). La página HTML visualizada en un navegador se puede ver en la figura 12.

5. XML e Internet

XML está considerado por muchos como el sucesor de HTML en Internet. Aunque ya empieza a haber implementaciones de servidores Web y navegadores que soportan XML aún hay mucho trabajo por hacer hasta que estas aplicaciones sean las más usuales. Ahora mismo gran parte del esfuerzo se concentra en conseguir que los datos XML sean accesibles desde el máximo de aplicaciones, para así difundir su utilización. Esta es la labor que realizan las herramientas que se pueden acoplar a los servidores Web, que permiten transformar los datos XML en HTML. Estas herramientas son procesadores de XML y XSL (procesadores XSLT). Este es el caso por ejemplo, de XML-enabler de IBM (11). Otro ejemplo es Cocoon (12), que capacita al servidor Web de Apache para manipular XML.

En lo que a los navegadores respecta, éstos también prometen manipular XML en un futuro muy cercano. Sin embargo, en este momento el único navegador que soporta XML es Internet Explorer 5 y, aún así, soporta un conjunto limitado de XML: hojas de estilo CSS y una versión reducida de XSL correspondiente a un Working Draft anterior a la recomendación estable (es decir, una versión obsoleta). En las secciones siguientes nos centraremos en uno de los aspectos más importantes de Internet: los enlaces. Además, estudiaremos hasta qué punta son adecuadamente soportados e implementados en las actuales herramientas de Internet.

6. Enlaces con XML

Los documentos pueden estar relacionados de muy diversos modos: porque sean del mismo autor, traten del mismo tema, tengan títulos similares, o cualquier

otra causa que un usuario considere suficiente para establecer una relación entre dos documentos. Los enlaces permiten expresar las relaciones entre datos y ampliar así la información del lector con otros datos relacionados con el documento al que accede en ese momento. Por ejemplo, un enlace en un documento desde la aparición de un término hasta otro punto donde se encuentra una definición de dicho término facilita la labor de comprensión del lector, que obtiene directamente dicha definición sin necesidad de realizar una búsqueda en un diccionario. Los enlaces entre un documento y las anotaciones que el usuario (u otros usuarios) haya realizado sobre dicho documento aportan una información complementaria que también puede ser muy útil en algunos casos. Como éstos, se podrían citar otros muchos ejemplos donde la disponibilidad de documentos relacionados, de un modo tan simple como pulsar un botón del ratón en un ordenador, cuando menos, simplifica la tarea de extracción de información que se está realizando. El ejemplo más obvio actualmente es el de Internet, que debe su popularidad no tanto a la gran cantidad de información disponible como a los enlaces que permiten "navegar" de un documento a otro.

6.1. ¿Qué tienen los enlaces XML que no tengan los enlaces HTML?

HTML ha popularizado los enlaces. Las páginas HTML contienen enlaces (caracterizados por la marca <a>...), que establecen un vínculo entre el fragmento de texto etiquetado y la URL a la que apuntan. Un enlace como el de la figura 13 nos llevaría desde este documento hasta la especificación de XML; a su vez, desde esta página podríamos seguir otros enlaces y así navegar de una página a otra.

Sin embargo, para retroceder al documento donde iniciamos el recorrido es necesario recurrir al menú de la aplicación que utilizamos (navegador en el caso de la Web), buscar la historia de la trayectoria que hemos seguido, y seleccionar el punto original. Dependemos del navegador, ya que los enlaces HTML son unidireccionales: se recorren desde el origen (documento donde se encuentra) hacia el destino (URL a la que apunta), pero es imposible hacerlo en sentido inverso.

Otra limitación que en principio pudiera no parecerlo, es que los enlaces HTML siempre están incluidos en el interior del documento origen. Esto no es ningún problema cuando somos los creadores del documento, ya que únicamente tenemos que insertar los enlaces en los puntos que consideremos apropiados, como ocurre en el ejemplo de la figura 11, donde el enlace se encuentra embebido en el texto del documento. Pero en otros casos, como el de las "anotaciones", es posible que queramos comentar documentos de los cuales no somos autores, y en los que, por tanto, no tenemos permiso para escribir. Esta situación es aún más evidente cuando varios usuarios comparten un documento. A todos ellos se les permite "leer" el documento, pero no se les permite "escribir", de esta manera se

preserva la integridad del texto original. Así pues, no queda más remedio que hacer las anotaciones "fuera" del documento.

Estas dos limitaciones por si solas son suficientes para justificar la necesidad de enlaces más potentes y flexibles que los que ofrece HTML.

Estos enlaces son los enlaces XLink (*XML Linking Language*) (XLink, 2000).

6.1.1. Características relevantes de los enlaces XML

Las características más relevantes de los enlaces XML, que los hacen más potentes que los enlaces HTML son las que se enumeran a continuación:

```
<html>
 ....
<h1>Referencias</h1>
<!-- enlace simple desde el documento donde se encuentra,
     que apunta hacia la Recomendación XML -->
<li>
<a href = "http://www.w3.org/TR/REC-xml">
  Extensible Markup Language (XML) 1.0 (Second Edition)</ax>.
  W3C Recommendation 6 October 2000. http://www.w3.org/TR/REC-xml.
</li>
 ...
</html>
```

Fig. 13. Un enlace HTML. Siguiendo este enlace, el usuario accede a la especificación de XML, cuya URL es http://www.w3.org/TR/REC-xml.

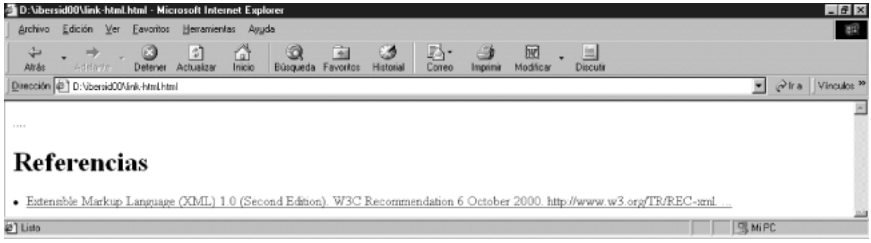

Fig. 14. Visualización del enlace HTML de la figura 11 en un navegador.

1. *Extensibilidad*: Los enlaces XLink —al igual que cualquier dato XML— no están limitados a una etiqueta o conjunto de etiquetas predefinidas. Las etiquetas de un XLink las define el autor de los enlaces, cuya única obligación es indicar de algún modo (que explicamos en esta misma sección) que ese elemento es un enlace.
2. *Inclusión de información acerca de la semántica de la relación*: Con XLink es posible incluir información adicional sobre la relación que liga los recursos, como puede ser la fecha en que dicha relación comenzó o finalizó, el tipo de vínculo existente, etc. Los elementos y atributos que forman parte de un enlace HTML están predefinidos por la especificación HTML.

   Los enlaces XLink pueden incluir información que indique a la aplicación que los manipula qué debe hacer cuando se encuentre con el enlace. Por ejemplo, si debe esperar o no a que el usuario se posicione sobre el enlace para ejecutar una determinada acción.
3. *Enlaces bidireccionales*: Los enlaces XLink pueden ser unidireccionales o bidireccionales.
4. *Enlaces múltiples*: Los enlaces XLink pueden tener múltiples orígenes y múltiples destinos. Por ejemplo, un mismo enlace puede agrupar todas las anotaciones hechas por distintos autores a un cierto texto. Así, una aplicación podría presentar el texto y a su lado todas las anotaciones disponibles sobre dicho texto. También es posible el caso contrario; por ejemplo, desde un texto se pueden "enlazar" varias definiciones de un mismo término provenientes de distintas fuentes, que aparecerían junto a éste para que el usuario pudiera compararlas.
5. *Enlaces fuera de los documentos:* Los enlaces no tienen porqué estar obligatoriamente dentro de los documentos (enlaces out-of-line). Esto permite, por ejemplo, hacer "anotaciones" a documentos sobre los cuales no se tiene permiso de escritura.
6. *Ligar fragmentos de documentos*: XLink es capaz de direccionar documentos enteros (como HTML), pero también es capaz de direccionar fragmentos concretos de un documento. Nuevamente es importante recordar que no es necesario editar el texto al que apunta el enlace para insertar ningún tipo de marca; es posible hacerlo con solo conocer la estructura del documento.
7. *Manipulación como cualquier otro dato XML*: Dado que los enlaces XLink también son datos XML, es posible explotar todas las capacidades de consulta de XML. Sobre estos enlaces se pueden realizar consultas al igual que con cualquier documento XML.

6.2. Tipos de enlaces XML

Se distinguen dos clases de enlaces en XML:

1. *Simples*: Vinculan dos recursos. Son similares a los enlaces HTML.
2. *Extendidos*: No tienen equivalente en HTML. Engloban los enlaces múltiples (que vinculan más de dos recursos) y *out-of-line*.

En la figura 15 se puede ver un enlace Xlink simple, cuya funcionalidad es similar a la de los enlaces HTML. La etiqueta o nombre del elemento es elegida por el usuario y es irrelevante para que la aplicación sepa que se trata de un enlace. De hecho, las aplicaciones lo reconocen como tal gracias a la presencia del indicador xmlns:xlink="http://www.w3.org/1999/xlink" dentro de la etiqueta de apertura del elemento.

Todos los atributos de un enlace que provienen de la especificación XLink deben llevar el prefijo xlink delante del nombre de atributo y separado de éste último por dos puntos, ':'. En el ejemplo, se sabe que se trata de un enlace simple (embebido en el documento y unidireccional) porque así nos lo indica el valor del atributo xlink:type. Los valores que puede tomar este atributo están especificados

```
<!-- el elemento REFERENCIA es un enlace simple desde el documento
     donde se encuentra, que apunta hacia la Recomendación XML -->
<REFERENCIA xmlns:xlink = "http://www.w3.org/1999/xlink"
          xlink:type = "simple"
          xlink:href = "http://www.w3.org/TR/REC-xml">
  Extensible Markup Language (XML) 1.0 (Second Edition)</REFERENCIA>
```

Fig. 15. Enlace XML simple. Vincula el documento donde se encuentra con la especificación XML.

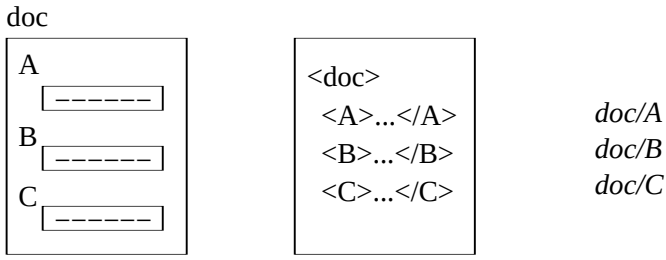

Fig. 16. Enlace XML extendido. Vincula el índice de un documento (index.xml) con las cuatro secciones que lo componen (s1.xml, s2.xml, s3.xml, s4.xml).

en la norma XLink y cualquier otro valor será reconocido como erróneo por una aplicación informática.

El enlace de la figura 16 es un enlace múltiple que sirve para crear una tabla de contenidos de este artículo. En primer lugar, se listan todos los recursos que forman parte del enlace (elementos SECCION). Seguidamente, los elementos de tipo ARCO establecen las conexiones entre el documento que contiene el enlace (tabla de contenidos) y las distintas secciones (véase la figura 17).

6.3. Cómo reconocer un xlink

XLink especifica la sintaxis de los elementos XML que son enlaces, y cómo caracterizar estos elementos para que las aplicaciones los reconozcan como enlaces. Además, XLink define algunos atributos que aportan información adicional a las aplicaciones sobre de qué tipo de enlace se trata y cómo manipular estos elementos.

Los elementos XLink se declaran en la DTD al igual que los restantes elementos de un documento XML; se debe especificar en esta declaración todos los atributos del elemento enlace —provengan de la especificación o sean creación del autor. El fragmento de DTD donde se declaran los elementos ARTICULO y sus descendientes SECCION —como el de la figura 16— es el que se encuentra en la figura 17; los elementos SECCION tienen 3 atributos (xlink:type, xlink:href, xlink:role), todos ellos definidos en la especificación XLink.

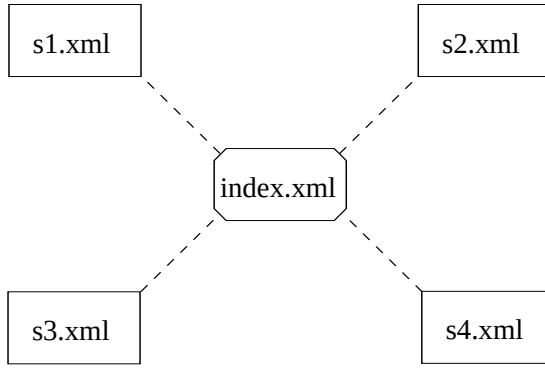

Fig. 17. Recursos que componen el enlace que define una tabla de contenidos, según se especifica en el xlink de la figura 14.

6.4. Granularidad en los enlaces: XPointer y XPath

Existen casos en los que un enlace afecta a fragmentos de documentos en vez de documentos completos. Si se está comentando una pieza de teatro, lo normal es que haya anotaciones específicas para cada escena. Igualmente, si se trata de citar la normativa reguladora de los contratos de los profesores universitarios, ésta se encuentra en unos artículos concretos dentro de una ley que abarca más aspectos relacionados con la universidad. El enlace afectará únicamente a los artículos referentes a los profesores.

6.4.1. Limitaciones de HTML

Con HTML, conseguir esta granularidad en los enlaces supone editar el documento —obra de teatro, LRU— e insertar marcas en las porciones de los documentos que van a formar parte de los enlaces —escenas, artículos de la ley—. En el momento de crear el enlace se usaría cada una de estas marcas como "destino" (href) del enlace. Sin embargo, si no es posible editar los documentos —lo más probable en los casos de las piezas teatrales y textos jurídicos— resulta imposible crear este tipo de enlace. Así pues, si no se dispone de permisos de edición sobre todos los documentos implicados en el enlace, no es posible conseguir enlaces a un nivel más fino de especificidad que el del documento. Este problema se solventa con XML.

6.4.2. La solución XML: XPointer

XPointer (*XML Pointer Language*) (Xpointer, 2001) es un estándar asociado a XLink que aborda el aspecto de direccionamiento de fragmentos internos de los documentos. XPointer está basado en XPath (*XML Path Language*) (XPath,

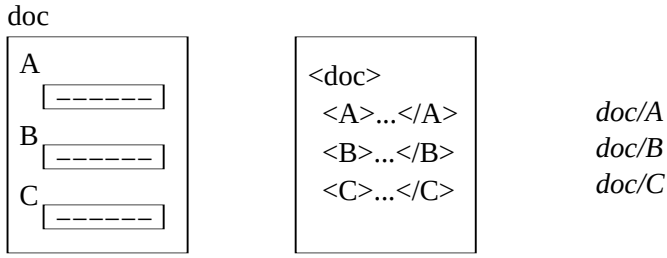

Fig 18. Con XPath, una porción de un documento se direcciona en base al camino, que comienza el elemento más externo. A la derecha de la figura se muestran los xpath que permiten acceder respectivamente a los elementos (divisiones internas) A, B y C del documento XML representado a su izquierda.

1999). Con XPath, una porción de un documento se direcciona en base al camino que hay que seguir para llegar a un elemento desde el elemento raíz (más externo) del documento. Para cada elemento se indica la secuencia de elementos (etiquetas de comienzo de elemento) por las que se debe pasar hasta llegar a él (figura 18). En el ejemplo del libro de la figura 2, la cadena XPath libro/autor/apellido selecciona todos los apellidos de los autores que estén dentro de algún elemento libro (en el ejemplo, 'Elmasri' y 'Navathe').

El enlace de la figura 19 —cuya representación gráfica se puede apreciar en la figura 18— es similar al enlace extendido de las figuras 16 y 17, aunque en este caso los enlaces no apuntan hacia documentos completos, sino hacia fragmentos de documentos. Concretamente, las secciones 1 y 2 del artículo representado por el enlace provienen del documento d1.xml; a su vez, las secciones 3 y 4 corresponden respectivamente a las secciones 1 y 2 del documento d2.xml.

6.5. XLink, XPointer e Internet

Los enlaces son fundamentales en Internet y, dadas las ventajas que Xlink ofrece, parece claro que en breve estos enlaces pasarán a formar parte de Internet.

```
<!-- el elemento ARTICULO es un enlace múltiple -->
<ARTICULO xmlns:xlink = "http://www.w3.org/1999/xlink"
          xlink:type = "extended">

  <!-- Recursos que forman parte del enlace ARTICULO -->
  <TS xlink:type = "locator" xlink:href = "index.xml" xlink:role="index"/>
  <SECCION xlink:type = "locator"
           xlink:href = "d1.xml#xpointer(doc/seccion[1])" xlink:role="seccion"/>
  <SECCION xlink:type = "locator"
           xlink:href = "d1.xml#xpointer(doc/seccion[2])" xlink:role="seccion"/>
  <SECCION xlink:type = "locator"
           xlink:href = "d2.xml#xpointer(doc/seccion[1])" xlink:role="seccion"/>
  <SECCION xlink:type = "locator"
           xlink:href = "d2.xml#xpointer(doc/seccion[2])" xlink:role="seccion"/>

  <!-- Arcos entre los recursos -->
  <!-- Hay enlaces desde el índice hacia todas las secciones -->
  <ARCO xlink:type = "arc" xlink:from = "index" xlink:to="seccion"/>
  <!-- y desde las secciones hacia el índice -->
  <ARCO xlink:type = "arc" xlink:from = "seccion" xlink:to="index"/>

</ARTICULO>
```

Fig. 19. xlink extendido con accesos a fragmentos de documentos. Se liga el índice (index.xml) con las 4 secciones que componen el documento, las cuales se extraen de los documentos d1.xml y d2.xml.

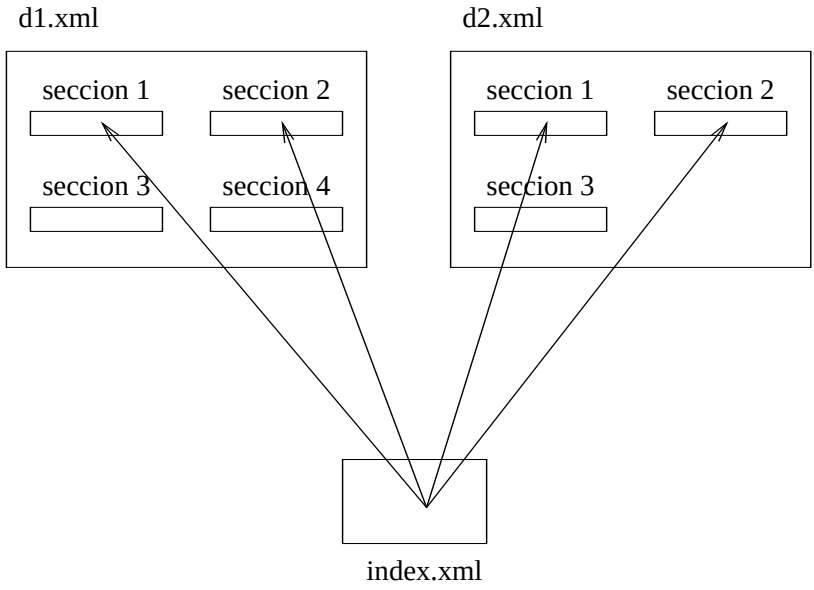

Fig. 20. Representación gráfica del enlace extendido de la figura 19.

Por tanto, la pregunta que parece lógica es "¿Soportan los servidores y navegadores XLink?". La respuesta a esta pregunta es que en este momento ninguna de estas herramientas lo soporta. La razón es que en este momento XLink y Xpointer son aún borradores (*working drafts*), en vez de recomendaciones estables. Sin embargo, dado que ya existen sendas recomendaciones candidatas, es de esperar que en el momento en que dichas recomendaciones sean aceptadas por el W3C como estables, tanto los navegadores como los servidores soportarán este tipo de enlaces en sus próximas versiones. De hecho, se asegura su inclusión tanto en Internet Explorer 6 como Netscape Communicator 6.

7. Ejemplos de aplicación

Existen múltiples circunstancias en las que se puede explotar la potencia de XML y de sus estándares asociados. En este apartado se comentan algunos contextos donde ya se han experimentado, y han quedado claras las ventajas conseguidas con esta decisión.

7.1. Revistas on-line

La difusión de publicaciones a través de Internet es cada vez mayor. Son ejemplos de esta tendencia revistas como Dlib on-line Magazine (13) —dedicada a las bibliotecas digitales—, Library & Information Management On Line (14), etc. La expansión del fenómeno ha animado a un colectivo de editores —el Reference Linking Working Group (15)— a promover reuniones y proyectos destinados a favorecer la creación, difusión y compartimiento de recursos entre revistas. Dentro de esta filosofía colaborativa, los enlaces ocupan, lógicamente, un lugar fundamental.

Si un determinado artículo o fragmento de artículo es reutilizado en más de una publicación, el objetivo es mantener enlaces a una copia del recurso en vez de hacer copias locales. Los sumarios son en realidad colecciones de enlaces a los artículos; parece por tanto que el modo más conveniente de implementarlos sea utilizando enlaces. Los artículos contienen en su interior referencias a otros artículos, que son asimismo enlaces a dichos trabajos; nuevamente, disponer de enlaces que posibiliten acceder directamente a los artículos referenciados es un valor añadido para la revista. Las referencias bibliográficas son objeto de múltiples estudios: "¿cuántas veces aparece referenciado un artículo?", "¿cuántas veces aparece referenciado un autor?", "¿en qué artículos se cita un trabajo concreto?". Estas preguntas y otras similares son en realidad preguntas sobre los enlaces (las referencias bibliográficas en este caso) que —como se vio al enumerar las propiedades relevantes de los xlink la sección 6— se pueden contestar usando los XLink.

7.2. Los documentos jurídicos

Los documentos jurídicos son otro buen ejemplo de que la utilización de XML aporta grandes ventajas. El tratamiento automático de la información legislativa ha atraído el interés de aquellos que investigan o producen software para gestionar documentos debido a la demanda de este tipo de servicios por parte de la comunidad jurídica. Esto hace que sea un tipo de información sobre la cual están muy claros cuáles son los servicios que se quieren obtener en su tratamiento. Además la información jurídica tiene una serie de peculiaridades, como su rigidez y la enorme cantidad de interrelaciones entre los documentos (Wilson, 1990; Di Giorgi, 1992) que la hace especialmente adecuada para ser modelada como documentos estructurados —con XML— e hipertextuales.

Dadas estas circunstancias no es de extrañar que existan bastantes proyectos en los que se aborda el modelado de este tipo de documentos y de sus interrelaciones, que se localizan en su mayoría en la década de los 90. Wilson (Wilson, 1990) enumeraba en 1990 las principales características y servicios que se podían esperar de un sistema que manipule información jurídica. Hacía especial hin-

capié en los relaciones bidireccionales, ilustrándo su importancia del siguiente modo: es tan interesante para un especialista tener acceso a toda la jurisprudencia relacionada con una ley (documentos que citan esta ley) como ser capaz de acceder, —dada una sentencia— a todas las normas referenciadas en la sentencia (normas que cita este documento).

SGML supuso para los sistemas de información jurídica la solución ideal. Los documentos jurídicos tienen una estructura muy estable, que es importante mantener; delimitar la extensión de un artículo concreto de una norma puede ser tan importante como el acceso a la norma. La solución consistente en etiquetar los fragmentos de los documentos facilita el almacenamiento de esta estructura, evitando complejas estructuras de enlaces que eran la solución utilizada previamente (Agosti, 1991). Surgen así propuestas de DTD creadas para los documentos jurídicos, como el proyecto Legis (Haider, 1996; Magnusson-Sjöberg, 1997, Arnold-Moore, 1997, Eulegis). En otros casos la propuesta consiste en aplicar una DTD estándar —la TEI— a los documentos jurídicos (Finke, 1997); sin embargo, cuando se trata de información con una estructura tan bien definida como la de los documentos jurídicos es preferible aplicar una DTD que refleje dicha estructura en vez de utilizar subterfugios para adaptar la DTD estándar a las peculiaridades jurídicas.

Existen servidores públicos de información legislativa, que contienen los documentos en formato HTML. El Estado español dispone de un servicio de este tipo accesibles desde las páginas ministeriales. Algunos servidores similares son Leggifrance (16) del Estado francés y el servicio del Legal Information Institute (17) en la universidad de Cornell (USA).

7.3. Cómo aprovechar XML para explotar las relaciones en el entorno jurídico

En nuestro caso pretendíamos obtener un modelo de documentos que nos permitiese modelar la estructura semántica inherente a cada tipo de documento, acceder a los fragmentos de los documentos, reflejar las interrelaciones entre documentos derivadas de las citas entre documentos con la mayor precisión posible en cuanto a la especificidad del enlace. También era deseable ser capaz de responder a preguntas que tienen que ver con las relaciones entre los documentos más que con los propios documentos, del tipo "¿en qué documentos se cita esta ley?".

7.3.1 Documentos etiquetados en base a su semántica

Teniendo en cuenta estos requisitos y las características de XML descritas, parecía claro que el estándar adecuado era XML, ya que nos permitía crear las etiquetas a medida de la legislación española (semántica), definir nuestras pro-

pias DTD, y postergar los aspectos relacionados con el formato para las fases finales de presentación de información al usuario. Las figuras 21 y 22 muestran dos fragmentos de la Constitución española. El fragmento de la subfigura 21 es un documento etiquetado con HTML, mientras que el de la subfigura 22 es un documento XML etiquetado con nuestras propias etiquetas. Las etiquetas en el primer caso no aportan ninguna información útil para saber con qué tipo de elemento (artículo, sección, título) estamos trabajando; en el segundo caso, permiten a una aplicación informática reconocen sin ambigüedad el tipo de elemento de que se trata y los límites (principio y final) de éste.

```
<P>
<H3>Art&iacute;culo 1</H3>
<P>
1. España se constituye en un Estado social y democr&aacute;tico de Derecho, que
propugna como valores superiores de su ordenamiento jur&iacute;dico la
libertad , la justicia , la igualdad y el pluralismo pol&iacute;tico .
<P>
2. La soberan&iacute;a nacional reside en el pueblo español , del que emanan los
poderes del Estado.
<P>
3. La forma pol&iacute;tica del Estado español es la Monarqu&iacute;a
parlamentaria .
<P>
<P>
<H3>Art&iacute;culo 2</H3>
```

Fig. 21. Fragmento de la Constitución con HTML.

```
<articulo>Artículo 1.
<P>1. España se constituye en un Estado social
 y democrático de Derecho, que propugna como
 valores superiores de su ordenamiento jurídico
 la libertad , la justicia , la igualdad y el
 pluralismo político .</P>
<P>2. La soberanía nacional reside en el pueblo
 español , del que emanan los poderes del Estado.</P>
<P>3. La forma política del Estado español es la
 Monarquía parlamentaria.</P></articulo>
<articulo>Artículo 2.
```

Fig. 22. El mismo fragmento de la Constitución con XML.

7.3.2 Tratamiento en dos fases

La manipulación de la información se separa en dos fases (figura 23), lo cual simplifica los tratamientos de la primera etapa :

1. *Tratamiento de la información*: indexación, composición de nuevos documentos, etc.
2. *Presentación al usuario*. Consiste en la obtención de documentos HTML, aplicando hojas de estilo CSS o XSLT.

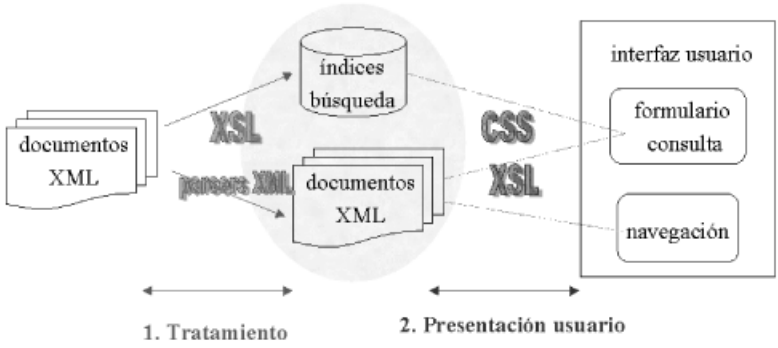

Fig. 23. Manipulación de la información jurídica en dos fases temporales: tratamiento y presentación al usuario.

7.3.3 Hipertexto más potente

En cuanto a los enlaces, hemos conseguido un sistema donde éstos tienen la máxima especificidad. Una referencia a un artículo específico de un documento se puede representar como un enlace hacia dicho artículo dentro del documento (en vez del documento); por ejemplo, las citas a los artículos 3º y 4º de la Constitución se almacenan en enlaces que nos pueden dirigir desde el texto citado hasta exactamente los artículos 3 y 4 de la Constitución, en vez de al principio de la Constitución. Esto permite obtener un hipertexto mucho más rico de lo que se puede conseguir en los sistemas basados en HTML:

1. Se pueden crear índices por categoría, fecha, etc.
2. No es necesario editar ningún documento para crear enlaces.

3. Se puede "navegar" en cualquier sentido ("¿en qué documentos se cita esta ley?","¿qué documentos cita esta ley?").

8. Conclusiones

Se ha presentado XML como una solución para modelar los documentos comparándola con HTML. Ambos derivan de SGML y, dado que gran parte de los problemas de SGML que XML viene a resolver se han detectado a través de la experiencia con HTML, se ha enfocado esta presentación en base a dicha comparación. No obstante, se quiere resaltar que ni XML pretender sustituir a HTML en la creación de documentos no estructurados en Internet, ni todas las aplicaciones de XML están relacionadas con Internet.

Alrededor de XML existen multitud de estándares, de DTD, etc. que no se han contemplado aquí. Esta presentación se ha centrado en la utilidad de XML para modelar información que tiene por si misma una semántica propia —en muchos casos proveniente de su estructura conceptual—, como el caso de la información jurídica, o en la cual se desea realizar una división del documento en fragmentos totalmente independiente de cualquier aspecto relacionado con su formato o presentación. El otro aspecto que se ha destacado es la utilización de XML para modelar las relaciones entre los documentos. Otras utilidades de XML, como el modelado de metainformación, deberían ser objeto de un estudio diferente.

Algunos de los estándares que se han comentado no han alcanzado aún el nivel de recomendaciones estables. Esto hace que XML apenas haya comenzado a convertirse en estándar en un entorno tan importante como es Internet. No cabe duda, sin embargo, de que sus capacidades son prometedoras, y así lo reconocen las empresas que comercializan aplicaciones Internet, con el cercano seguimiento de la evolución de todos los estándares relacionados con XML y su participación en los equipos editoriales de algunas de estas especificaciones. Es previsible, por tanto, que en breve se extenderán las aplicaciones que incluyen entre sus virtudes el soporte para XML.

9. Notas

(1) Por "mala calidad" se entiende aquí la dificultad o imposibilidad para someter la información a un tratamiento automático por parte de una aplicación informática. En absoluto se consideran aspectos relacionados con la calidad de la información que contiene el documento, su organización o presentación.

(2) Su URL es: http://www.w3c.org

(3) Los estándares W3C pasan por varias revisiones hasta ser aceptados como estándares estables. En este momento se les denomina recomendaciones, para diferenciarlos de

las versiones anteriores, susceptibles de ser modificadas en breve tiempo, que se dicen working drafts.

(4) Para resolver este problema se ha propuesto una versión de HTML conforme a las normas XML, denominada XHTML (XHTML, 2000).

(5) Si bien una persona sería capaz de inferirlo, una herramienta software no tiene esta capacidad.

(6) Los parsers XML son herramientas software capaces de procesar un documento XML, verificar su corrección sintáctica, extraer la información de sus elementos y atributos, y pasar esta información a otras aplicaciones.

(7) Su URL es: http://www.w3c.org/TR/xmlschema-1

(8) Si bien nos centramos en los navegadores por ser la situación más conocida, es importante reseñar que las hojas de estilo también se usan para imprimir documentos y transformar a otros formatos (Postscript, PDF, etc.).

(9) Apache es el servidor Web más popular actualmente. Para más información, consultar http://www.apache.org.

(10) XPath es también un elemento fundamental en los enlaces XML.

(11) Su URL es: http://alphaworks.ibm.com/tech/xmlenabler

(12) Su URL es: http://xml.apache.org/cocoon/

(13) Su URL es: http://www.dlib.org

(14) Su URL es: http://www.liblink.co.uk/limo/

(15) Su URL es: http://www.niso.org/reflink.html

(16) Su URL es: http://www.leggifrance.fr

(17) Su URL es: http://www.law.cornell.edu

10. Referencias

Agosti, M. ; Colotti, R. ; Gradenigo, G. (1991). A two-level hypertext retrieval model for legal data. // Proceedings of the 14th ACM-SIGIR International Conference on Research and Development in Information Retrieval : Chicago, IL USA, Oct. 1991.

Arnold-Moore, T. ; Anderson, P.; Sacks-Davis, R. (1997). Managing a digital library of legislation. Proceedings of the 2nd ACM International Conference on Digital Libraries, ACM DL 1997 : Philadephia, PA USA, Jul. 1997, ACM Press. 175-183.

Conklin, J. (1987). Hypertext: An introduction and survey. // IEEE Computer. 20 : 9 (1987) 17-41.

Eulegis. URL: <http://www.eulegis.net>.

Finke, N. (1997). TEI Extensions for Legal Text. Proceedings of the Text Encoding Initiative Tenth Anniversary User Conference, Providence, Rode Island USA, Nov. 1997.

Di Giorgi, R.M. ; Nannucci, R. (1992). Hypertext systems for the law. // Actas de Informatique et droit / Computers and law : Montreal, 1992.

Haider, G. ; Manusson Sjöberg, C. ; Quirchmay, G. ; Sebald, V. (1996). The Comparative Part of the Corpus legis Project : Using SGML for Intelligent Information Retrieval

of Legal Documents : Proceedings of EXPERSYS-96, Artificial Intelligence Applications, 1996. 181-186.

Information Processing : Text and Office Systems : Standard Generalized Markup Language (SGML). International Organization for Standardization, Geneve (1986). ISO 8879:1986 (1986).

Magnusson Sjöberg, C. (1997). DTD development for the legal domain : Proceedings of Swedis SGML 97, 1997. URL:<http://info.admin.kth.se/SGML/>

Rusty Harold, E. (1999). The XML Bible. IDG Books, 1999.

Sperberg-McQueen, C.M.; Burnard, Lou. (eds.) (1994). Guidelines for Electronic Text Encoding and Interchange. ALLC/ACH/ACL Text Encoding Initiative (1994).

Walsh, N.; Muellner, L. (1999). DocBook: The Definitive Guide. O'Reilly: 1st edition (October 1999).

Wilson, E. (1990). Links and structures in hypertext databases for law. // Rizk, Antoine; Streitz, Norbert A.; André, J. (eds.). European Conference on Hypertext, ECDHT'90. Paris: The Cambridge Series on Electronic Publishing, Cambridge University Press, 1990. 194-211.

Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation. Tim Bray, Jean Paoli, C.M. Sperberg-McQueen, Eve Maler, 6 October 2000. URL:<http://www.w3.org/TR/REC-xml>

XHTML[tm] 1.0: The Extensible HyperText Markup Language : A Reformulation of HTML 4 in XML 1.0 : W3C Recommendation 26-January-2000. URL:<http://www.w3.org/TR/2000/REC-xhtml1-20000126>

XSL Transformations (XSLT). W3C Recommendation. James Clark, 16 November 1999. URL:<http://www.w3.org/TR/1999/REC-xslt-19991116>

XML Path Language (XPath). W3C Recommendation. James Clark, Steve DeRose, 16 November 1999. URL:<http://www.w3.org/TR/1999/REC-xpath-19991116>

XML Pointer Language (XPointer). W3C Working Draft. Steve DeRose, Eve Maler, Ron Daniel, 8 January 2001. URL:<http://www.w3.org/TR/2000/WD-xptr-20010108>.

XML Linking Language(XLink). W3C Working Draft. Steve DeRose, Eve Maler, David Orchard, Ben Trafford, 21 February 2000. URL:<http://www.w3.org/TR/2000/WD-xlink-20000221>.