Comunicación

Ventajas de la tecnología XML dentro de la Ciencias de la Información

Lic. Raúl Entralgo Amaro
Lic. Jorge Dayán Aguiar Cedeño

RESUMEN

Se investigan las ventajas del lenguaje XML en el ámbito de la Ciencias de la Información. Se conceptualizan las diferentes tecnologías que lo antecedieron: lenguajes de marcado SGML y HTML, como también el papel de la W3C en su desarrollo. Puntualiza el estudio sobre la coyuntura histórica y tecnológica que fomentó el perfeccionamiento del lenguaje. Comparativamente se diferencian el lenguaje HTML del XML. A través del análisis se describen las tecnologías que se derivan del XML, los lenguajes de enlaces, de visualización y de recuperación de información. Realiza un especial énfasis en los lenguajes de recuperación de información XPointer, XPath y XQuery. Las investigaciones detallan los tres puntos neurálgicos del lenguaje que responden a las Ciencias de la Información y aporta en sus conclusiones las ventajas del lenguaje objeto de estudio.

Palabras clave: lenguaje HTML, lenguaje XML, documentos XML, recuperación de información.

ABSTRACT

This is a study of the advantages of the XML language in the field of Information Sciences and it conceptualizes the different preceding technologies: SGML and HTML marking languages, as well as the role W3C played in the development of this technology. It stresses the study of the historical and technological situation that fostered language improvement. Comparatively, HTML language was differentiated from XML. By way of the analysis, XML-derived technologies as well as information linking, displaying and retrieval languages are described. There is special emphasis on XPointer, XPath and XQuery information retrieval languages. The pieces of research result in detailing the three key points of language, which share common points with Information Sciences. The conclusions contribute the advantages of the language under study.

Keywords: XML language, XML documents, XQuery language, description tool.

Introducción

La Recuperación de Información (Information Retrieval), en la Bibliotecología y la Ciencias de la Información, es un proceso básico a la vez que posee una sustanciosa carga interdisciplinar: aportes y modelos de las Matemática, la Lingüística y la Computación.

Los diferentes modelos aplicados a la recuperación de información, aunque estén teóricamente bien sustentados, no han logrado en la práctica solucionar un problema aparentemente bien sencillo: *recuperar sólo la información pertinente de una solicitud de información.* Si se extrapola esta situación fuera de la biblioteca y se está en presencia de un ambiente automatizado, entonces el problema es más complejo.

Internet, por ejemplo, es un inmenso «almacén» donde hay información de todo tipo y formato, pero encontrar la que realmente un usuario desee sigue siendo una tarea necesitada de perfeccionamiento. Diferentes

teorías se han propuesto para alcanzar mejorías a la hora de recuperar información en ambiente Web. Una de ellas parte de la idea de que la información en la red no está estandarizada como lo debiera. Para lograr esta normalización la W3C presentó en el año 1998 un lenguaje basado en marcas con el fin de utilizarse en la WWW.

El lenguaje XML está creado para estandarizar todo tipo de documento en la Web, no importa su formato. Según se cree la inserción del lenguaje XML y las aplicaciones de sus tecnologías (XSL, XQuery, XPath, etc) en el ambiente Web, permitirá realizar motores de búsqueda mucho más eficaces; permitiendo así un acceso más rápido y eficiente a la información. Se espera, además, que se potencialice el intercambio de información y la cooperación entre las entidades, instituciones, empresas y demás organismos.

Marco conceptual del XML como lenguaje de marcado

Al conceptualizar este lenguaje es imprescindible citar a Howard Katz [1] quien sostiene que (traducido al español): «XML es un formato extremadamente versátil que ha sido usado para representar diferentes formas de datos, incluyendo páginas Web, libros, datos relacionados con los negocios y la contabilidad, representaciones XML de bases de datos relacionales, interfaces de programación, objetos, transacciones financieras, juegos de ajedrez, vectores de gráficos, presentaciones multimedia, aplicaciones de crédito y hasta manuscritos griegos….un documento XML es un lineamiento donde el orden y la jerarquía son las principales unidades estructurales.»

Otro autor plantea que «El lenguaje XML está emergiendo rápidamente como un estándar dominante para la representación de datos en Internet. Tal como el HTML, XML es una sección de SGML. Mientras las etiquetas HTML tienen como función principal la forma de presentación de las mismas, las etiquetas XML describen el dato en sí. La importancia de esta simple distinción no puede ser subestimada y, como el dato XML es autodescriptivo, posibilita entonces la interpretación de los datos por los programas. Esto significa que un programa que recibe un documento XML puede ser interpretado de maneras múltiples, puede filtrar el documento basándose en su contenido, reestructurarlo para que sirva a necesidades de aplicación, y demás» [2].

De los anteriores autores citados y otros consultados, se resume que XML (eXtensible Markup Language o Lenguaje de Marcado Extensible) es un metalenguaje, lo que significa que sirve para crear otros lenguajes. Se trata de una notación neutral para etiquetar diferentes partes del cuerpo de un documento y representar las relaciones entre ellas. Es un formato que permite la lectura de datos a través de diferentes aplicaciones. Este lenguaje se sustenta en un formato basado en texto, diseñado para el almacenamiento y la transmisión de datos. Un documento XML contiene texto anotado por cualquier cantidad de etiquetas. Estas últimas quedan a la libre elección del diseñador del documento en cuanto a tipología y cantidad.

Antecedentes

Definiciones pilares antes de adentrarse en XML, son los de Internet, WWW, lenguaje GML (General Markup Lenguage), SGML (Standard General Markup Lenguage), HTML, así como la caracterización de los procesos evolutivos que caracterizaron a cada uno de ellos. De estos evolucionó directamente el lenguaje moderno, objetivo de este trabajo.

Elemento clave en la evolución del formato aquí estudiado es la aparición, en los años ´70, del lenguaje de marcado GML confeccionado por IBM. Una vez normalizado y estandarizado en el año 1986 por la ISO, pasa a denominarse SGML (Standarized Markup Lenguage). El lenguaje de marcas extensible (XML), como lo conocemos hoy, no es otra cosa que una versión abreviada de este SGML; un subconjunto del mismo.

Es directamente del lenguaje de marcas SGML de quien deriva XML, y este último a su vez, posee un subconjunto de las funcionalidades del primero.

«SGML es un metalenguaje, es decir, un medio de describir formalmente un lenguaje, en este caso, un lenguaje de codificación etiquetado. Es un sistema «descriptivo» que se sirve de códigos que simplemente ofrecen nombres para categorizar e identificar partes de un documento. Esto significa que SGML es una norma elaborada para expresar estructuras de contenido en lugar de apariencia de documentos. Es decir, usa códigos de marcaje (etiquetas) que proporcionan nombres para categorizar las partes de un documento. Con metalenguajes de etiquetado descriptivo, como SGML, se diferencia claramente entre contenido y presentación…»[3].

SGML fue el primero de los metalenguajes y a partir de este se crean otros lenguajes, siendo HTML el más conocido y mundialmente utilizado de todos; con Berners Lee como su autor intelectual. HTML es

básicamente una especificación de SGML hecho para funcionar en la WWW.

En 1994 se constituye la W3C o World Wide Web Consortium, con el objetivo de desarrollar protocolos comunes para la evolución de Internet. Este consorcio trabajaría a partir de la segunda mitad de los 90 en base a perfeccionar los diferentes errores y dificultades que pronto saldrían a relucir de HTML.

El lenguaje HTML en su esencia carece de los mecanismos necesarios para describir y presentar los datos de un documento electrónico. De ahí nace la idea de etiquetar y estandarizar la información en el ambiente Web. Para ello la W3C presentó una posible opción que es el lenguaje XML, el cual aún hoy se encuentra en desarrollo. En teoría, el lenguaje XML debe aprovechar las innegables ventajas del HTML y, a su vez, permitir realizar muchas más aplicaciones.

Breve comparación con HTML

XML y HTML comparten muchas cosas en común. HTML es un tipo de documento SGML que se utiliza en la Web, en cambio, XML no es ningún tipo de documento SGML, sino más bien una versión abreviada de este último; optimizada especialmente para su utilización en Internet. Como escuetamente resume Michel Brundage en su libro *XQuery: The XML Query Language* [4]: «XML ofrece el 80% de las ventajas del SGML con un 20% de su complejidad».

En un principio, el lenguaje XML fue diseñado para que no rivalizar con HTML, estos se deberían complementar el uno al otro. De hecho, el primero se creó para que fuera idéntico a la hora de servir, recibir y procesar la información del otro, para aprovechar toda la tecnología implantada de este. No obstante, HTML es un formato que describe la apariencia, o sea, aporta datos estructurales sobre un documento Web. La diferencia entre uno y otro lenguaje radica precisamente en esto: HTML ofrece amplias facilidades de presentación de la información, pero no ofrece ninguna forma basada en estándares para administrar los datos; al menos no como XML. Este último está orientado hacía la información en sí misma y no hacia la representación, su función principal es describir datos, no mostrarlos. Visto así, HTML no indica lo que etiqueta, sino que se preocupa más por si el texto va en negritas, cursiva o subrayado, que por si es el título, autor o cualquier otra parte del documento. Su contraparte, el XML en cambio, se centra en el contenido de lo que etiqueta, no la forma y/o estética.

En la Tabla 1 se muestran las diferencias en cuanto a estructuración de datos, para esto se describe un mismo libro; según los dos lenguajes.

Tabla 1. Diferencias en cuanto a estructuración de datos, de un mismo libro; según los dos lenguajes.

HTML	XML
<TABLE>	<LIBROS>
<TR>	<LIBRO>
<TD>Título</TD>	<TITULO>Alexandros: El confín del mundo
<TD>Autor</TD>	<AUTOR>Valerio Massimo</AUTOR>
<TD>Precio</TD>	<PRECIO>12</PRECIO>
</TR>	</LIBRO>
</TR>	</LIBRO>
<TD> Alexandros: El confín del mundo	<TITULO>Cien años de soledad</TITULO>
</TD>	<AUTOR>Gabriel García Márquez</AUTOR>
<TD> Valerio Massimo </TD>	<PRECIO>32</PRECIO>
<TD>12</TD>	</LIBRO>
</TR>	</LIBROS>
<TR>	
<TD> Cien años de soledad </TD>	
<TD> Gabriel García Márquez </TD>	
<TD>32</TD>	
</TR>	
</TABLE>	

Estructura del lenguaje

Estructuralmente un documento en XML se compone de dos partes: una parte física y la otra lógica.

Para interés de este trabajo la parte física puede obviarse, por lo que se haría énfasis en el componente lógico que está integrado por declaraciones, elementos, atributos, comentarios, referencias a caracteres e instrucciones de procesamiento; todos los cuales están indicados por una marca explícita. Es importante que ambas estructuras encajen adecuadamente.

Cada documento XML contiene, como mínimo, un elemento. Los límites de estos elementos se delimitan mediante una etiqueta de comienzo, otra de final y una etiqueta de elemento vacío para el caso que lo requiera, tal como el lenguaje HTML. Cada elemento posee un tipo identificado por un nombre, denominado identificador genérico, y a la vez puede tener un conjunto de especificaciones de atributos. Los elementos XML pueden tener contenido, pero también ser elementos vacíos.

Los elementos y los atributos se declaran en el DTD, usando declaraciones de elementos y declaraciones de listas de atributos. Teniendo en cuenta que los componentes primarios descritos en un DTD son los elementos y los atributos, y que el elemento es la unidad lógica de información, entonces los atributos son las características de tal información.

Los elementos pueden tener atributos, que es una manera de incorporar características o propiedades a los elementos de un documento. La especificación de atributos sólo puede aparecer dentro de la etiqueta inicial y en los elementos vacíos. A cada una de las especificaciones de atributo se le asigna un nombre y un valor. Es importante puntualizar que estas especificaciones no restringen el contenido, el uso o los nombres de los tipos de elementos y atributos, que se reservan para estandarizar etiquetas o atributos en versiones posteriores.

Entre tanto, las entidades predefinidas son entidades utilizadas para representar caracteres especiales que no sean interpretados como marcado en un procesador XML.

Documentos bien formados y documentos válidos

Todo documento en XML debe cumplir con dos condiciones: ser válidos y estar bien formados.

Documentos bien formados «son todos los que cumplen las especificaciones del lenguaje respecto a las reglas sintácticas que después se van a explicar, sin estar sujetos a unos elementos fijados en un DTD (Document Type Definition). De hecho los documentos XML deben tener una estructura jerárquica muy estricta.....y los documentos bien formados deben cumplirla» [5].

En otras palabras un documento considerado well-formed (bien formado) debe cumplir, ante todo, con la regla denominada Document. Esta regla implica que el documento contenga uno o más elementos, y que haya exactamente uno, denominado «raíz» o también «elemento documento», del cual ninguna parte aparece en el contenido de ningún otro elemento. Para el resto de los elementos, si la etiqueta de comienzo está en el contenido de algún otro elemento, la etiqueta de fin está en el contenido del mismo elemento.

En cuanto a los documentos válidos es importante precisar que el concepto de validez, implica que no sólo el documento es bien formado, sino que también su estructura se corresponde con la definida en un documento externo, o sea, además de estar bien formados, siguen una estructura y una semántica determinada por un DTD, donde sus elementos y sobre todo la estructura jerárquica que define el DTD, además de los atributos, deben ajustarse a lo que él mismo especifique.

Los documentos XML con un DTD se reconocen como «XML válido». En este caso, un intérprete de XML podría comparar los datos entrantes con las normas definidas en el DTD, para comprobar que los datos se han estructurado como corresponde.

Especificaciones del XML

DTD

Las siglas de la DTD responden a lo que se traduce al español como Definición de Tipo de Documento. Una DTD «es una definición de los elementos que pueden haber en el documento XML y su relación entre ellos, sus atributos, posibles valores, etc.»[5]. Una DTD es un archivo que encierra una definición formal de un tipo de documento, a la misma vez que especifica la estructura lógica de cada uno. Se encarga además, de definir los elementos y atributos de una página Web. Como propone Goldfarb C, una DTD «es una definición exacta de la gramática de un documento» [6].

Cuando se hace referencia a la frase «crear una definición DTD», significa que el usuario crea su propio lenguaje

de marcado para una aplicación específica. La DTD define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos. Es importante aclarar que los documentos XML que se ajustan a su DTD se denominan válidos. Que un documento sea bien formado no quiere decir que sea válido, un documento bien formado simplemente es aquel que respeta la estructura y sintaxis definidas por la especificación de XML. Un documento bien formado puede ser válido si cumple las reglas de una DTD determinada.

Ead como ejemplo de DTD´s

«La EAD es una DTD XML que refleja la estructura lógica y jerárquica de un instrumento de descripción de archivo, que es compatible con la norma internacional para la descripción de material de archivo (ISAD-G) y que posibilita la difusión, acceso y navegabilidad, a través de la tecnología de redes, de información descriptiva de archivo» [3].

La EAD es, esencialmente, una estructura de datos normalizada que reproduce en formato digital los instrumentos de descripción archivística. La EAD permite incluir información suplementaria opcional que no describe directamente los registros, pero facilita su uso por parte de los investigadores, por ejemplo, una bibliografía. Los documentos EAD se benefician además de las posibilidades aplicables a los archivos XML, por ejemplo, los archivos EAD pueden ser enlazables entre sí o integrar imágenes de los documentos descritos empleando la tecnología XLink.

Una EAD contiene varios tipos de elementos:

- aquellos que codifican puntos específicos en la descripción de partes componentes del instrumento de descripción o el material que describe (elementos descriptivos - título de la unidad, fecha de la unidad, productor, etcétera).

- los utilizados para el acceso (nombre de entidad corpname-, nombre de persona, etcétera); de enlace y aquellos que podrían codificar cualquier característica del documento (elementos genéricos).

«A un nivel muy básico, un documento «instrumento de descripción» codificado utilizando EAD, consta de tres segmentos: uno que proporciona información sobre el instrumento de descripción en sí mismo (su título, compilador, fecha de compilación), <eadheader>; un segundo componente que incluye las cuestiones preliminares necesarias para la publicación formal del instrumento de descripción, <frontmatter>; y un tercero que proporciona la descripción del material archivístico en sí misma, además de la información contextual y administrativa asociada, <findaid>» [3].

XML Schemas

Un esquema XML es una versión de la DTD. Una XML Shema o XML Schema Definition (XSD), como se denomina en inglés, se utiliza con el fin de definir qué elementos puede contener un documento XML, cómo están organizados y qué atributos y de qué tipo pueden tener sus elementos.

Un XML Schema define elementos y atributos que pueden aparecer en un documento, como se planteó anteriormente, además, define cuáles elementos son elementos «hijos», qué elementos están «vacíos» o si incluyen texto, entre otras definiciones.

La aparición de las XML Schemas pretende sustituir, con carácter definitivo, a las DTD´s en un futuro no muy lejano. Esto se da por diferentes razones, entre ellas por el hecho que las primeras:

1. Utilizan sintaxis de XML, al contrario de los DTDs. Esta es una diferencia que las hace sumamente poderosas, entre otras cosas porque así se puede usar un editor en XML para diseñar sus propios schemas, se puede utilizar los mismos XML «parser», se puede también transformar los Schema con el uso de la tecnología XSLT, entre otros posibles usos.

2. Posibilitan especificar los tipos de datos.

3. Son extensibles, precisamente porque están escritos en lenguaje XML; lo que posibilita hacer referencias a diferentes schemas en un mismo documento, entre otras posibilidades.

A diferencia de una DTDs, un schema nos permite definir el tipo de contenido de un elemento o de un atributo, y especificar si debe ser un número entero, una cadena de texto o una fecha, etcétera. En otras palabras más información, más descripción de datos.

Un XML esquema es sumamente útil. Con un esquema, un autor define exactamente qué nombres de elementos se permiten en un documento y, dentro de cada elemento, qué subelementos, atributos y relaciones se admiten. Los XML esquemas son fruto de la idea de que debía surgir una opción mejor para suplir las debilidades de las DTD´s. Los documentos esquema se concibieron como una alternativa a las DTD más complejas, con el objetivo de superar sus puntos

débiles y buscar nuevas capacidades a la hora de definir estructuras para documentos XML.

El aporte principal del XML Schema es el gran número de tipos de datos que incorpora, como ya se planteó anteriormente. Por esta característica, los XML Schema aumenta las posibilidades y funcionalidades de aplicaciones de procesado de datos, incluyendo tipos de datos complejos como fechas, números y strings.

Tecnologías XML

Las tecnologías XML pueden ser agrupadas, a nuestro juicio, en tres grandes grupos: uno que se dedica a la visualización del contenido de un documento XML, otro dedicado a la recuperación de la información contenida en él y un último grupo que permite los enlaces entre documentos XML.

Dentro del primer grupo tenemos a las tecnologías que se encargan de la presentación de un documento XML: cómo se va a mostrar, cómo se va a imprimir, cómo va a visualizarse, cómo será su conversión a otros formatos, etcétera; en otras palabras, el aspecto estético de los documentos XML. En este grupo están los CSS (Cascading Style Sheets o en español hojas de estilo en cascada) y XLS (Extensible Stylsheet Language), XLST, etcétera.

Este primer grupo, al brindar las facilidades para la visualización de la información contenida en un documento XML, tiene una estrecha relación con la arquitectura de información y, dentro de esta, la parte que se dedica al diseño de las interfaces gráficas; en dependencia de la comunidad a la cual vaya dirigida dicha información.

El segundo grupo reúne en él a las tecnologías de recuperación de información Xpath, Xpointer y Xquery, de interés clave para este trabajo. Al centrarse en la recuperación de información, constituye un núcleo importante, pues a partir de los mismos es que se pueden implementar las diferentes vías de acceso a un documento. Más allá de la recuperación de un documento, a partir de palabras predefinidas por especialistas o postdefinidas por los usuarios, este grupo ha ido evolucionando para permitir la recuperación de aquellas partes de diferentes documentos que traten el tema de interés para el usuario.

El último grupo donde se encuentra el XLink (Extensible Linking Lenguage), mantiene vigente la lectura hipertextual, tan difundida en nuestros días, que permite conectar diferentes materiales relacionados con un tema, lo cual le permite al usuario una lectura guiada entre diferentes materiales, la posibilidad de conocer a través de un material la existencia de otros o la inclusión de comentarios, conceptos, imágenes, entre otros que enriquezcan el texto; siempre que se haga de una forma inteligente y estudiada, pues estas mismas potencialidades cuando no están diseñadas correctamente, lejos de ayudar al usuario, hacen que este se pierda en su lectura.

Lenguajes de recuperación

Xpath (XML Path Lenguange)

Xpath es un lenguaje relativamente complejo y bien sofisticado. Es un lenguaje que, como el mismo XML, todavía está en fase de desarrollo; lo que provoca que sea difícil encontrar herramientas que incorporen todas sus funcionalidades. Este lenguaje se utiliza para seleccionar y hacer referencia a textos, elementos, atributos, así como cualquier otro aspecto informativo en un documento XML. Es el lenguaje de Rutas XML el que ofrece la posibilidad de acceder a partes de un documento XML. Para Víctor Manuel Rivas Santos [7], XPath «sirve para decir cómo debe procesar una hoja de estilo, el contenido de una página XML, pero también para poder poner enlaces o cargar en un navegador zonas determinadas de una página XML, en vez de toda la página».

XPath, en un principio, fue parte de XSL 1.0 y luego se desarrolló como una especificación separada. Este lenguaje sentó las bases para diferentes herramientas que luego se analizarán y que pertenecen al sistema complejo de XML: Xquery y Xpointer, las que también se encuentran en pleno proceso y perfeccionamiento de desarrollo en la actualidad.

Xpointer

Esta tecnología está, como se introdujo con anterioridad, diseñada directamente del Xpath; es esencialmente una extensión de la misma. De hecho, un conocimiento previo de Xpath es obligatorio para trabajar con este lenguaje.

Xpointer es un lenguaje de direccionamiento XML, capaz de permitir acceso a estructuras internas de un documento XML. Estas estructuras internas son los denominados elementos, atributos y contenidos.

Xpointer tiene la facultad de cargar en un visualizador de documentos XML que sean de interés al usuario, tal como Xpath sigue siendo una tecnología en

desarrollo, incluso de todas las conocidas tecnologías del lenguaje XML es la menos extendida. Al ser XPointer una extensión de XPath, tiene todas las ventajas de este último y además, permite establecer un rango en un documento XML, es decir, con XPointer es posible establecer un punto final y un punto de inicio, lo que incluye todos los elementos XML dentro de esos dos puntos. Finalmente XQL, lenguaje de consulta, se basa en operadores de búsqueda de un modelo de datos para documentos XML que puede realizar consultas en infinidad de tipos de documentos: los estructurados, las colecciones, bases de datos, estructuras DOM, catálogos, etcétera.

«Una expresión Xpointer se añade a un URI (Uniform Resourse Identifier), como puede ser un URL (Uniform Resourse Locator) o un URN (Uniform Resourse Name)» [7].

XQuery (Lenguaje de Consulta XML)

Este es un lenguaje de consulta basado también en XML, que se caracteriza por su capacidad de extraer datos de múltiples fuentes, por ejemplo: de los mismos documentos XML, de las bases de datos relacionales, de repositorios de objetos, servicios Web, aplicaciones y sistemas heredados, entre otros. Es un lenguaje publicado por el W3C (World Wide Web Consortium) que utiliza la notación XML para definir consultas y manejar los resultados. Permite también la posibilidad de obtener datos de un archivo XML y una tabla de la base de datos relacional con tan solo una consulta.

«XQuery es un lenguaje funcional, lo que significa que en vez de ejecutar una lista de comandos como un lenguaje procedimental clásico, cada consulta es una expresión que es evaluada y devuelve un resultado, al igual que en SQL. Diversas expresiones pueden combinarse de una manera muy flexible con otras, para crear nuevas expresiones más complejas y de mayor potencia semántica» [8].

Como se había planteado, este lenguaje permite la posibilidad a los usuarios de realizar consultas flexibles con la intención de extraer datos de documentos XML en el ambiente Web. Es importante decir que no es el único para XML, pues hay otros previos a él y que sirvieron de base para idearlo, como son: XQL 99, XML QL, XSLT, Lorel, XPath 1.0, XQL, XML-QL, SQ, OQL y Quilt. El desarrollo de este lenguaje de consulta está fundamentado en la fusión de las ventajas de cada uno de los lenguajes anteriores. Xquery es el resultado de todos los elementos positivos de estos. «XQuery se nutre de las potencialidades de otros lenguajes de consulta, eliminando las deficiencias de estos y proponiendo nuevas formas que optimizan la pertinencia de la información a recuperar» [9].

Xquery es un subconjunto del Xpath, de un lenguaje capaz de localizar información dentro de un documento XML.

No obstante, todos los parentescos que se le puedan asignar a cada uno, vale señalar que XQuery tiene características que lo hacen una tecnología de mayor envergadura. Este añade a XML la funcionalidad de las bases de datos, además, proporciona potencialidad de búsqueda y selección. Con XQuery es posible realizar conexiones a través de bases de datos y proveedores, entre diferentes tipos de datos, incluyendo documentos XML, almacenamientos nativos de XML, tablas de bases de datos relacionales, entre otros muchos.

Ventajas del Lenguaje XML

A grosso modo las principales características de XML son:

- **Extensible:** es capaz de admitir la definición de un conjunto ilimitado de etiquetas. XML es un lenguaje donde los creadores pueden diseñar sus propios documentos, utilizando las estructuras que ellos prefieran, sin tener que regirse por un esquema cerrado como el de HTML. Estas cualidades han motivado a algunos autores a nombrarlo abierto y creativo.

- **Estandarización**: XML proporciona una representación estructural de los datos de probada implementabilidad y fácil distribución.

- **Datos separados de la presentación y el proceso**: XML únicamente usa etiquetas para describir los datos. Con el fin de mostrar estos en un navegador XML, utiliza hojas de estilo como son las Hojas de Estilo en Cascada (CSS) y el Lenguaje de Estilo Extensible (XSL).

- **Datos autodescriptivos**: las etiquetas descriptivas están entremezcladas con los datos.

Importancia del lenguaje en el espectro de las Ciencias de la Información:

1) Datos autodescriptivos

Las etiquetas son creadas al antojo del programador, editor o creador del documento, como se mostró en la Tabla 1. Esto implica que las etiquetas están desde

la base estrechamente ligadas al contenido. Si en HTML hay que insertar metadatos para crear un mecanismo con el fin de recuperar un documento satisfactoriamente, en XML los metadatos ya vienen definidos desde su esencia. Lógicamente, la recuperación de información será más exacta sobre documentos donde las «etiquetas» están, única y exclusivamente, dirigidas a definir el contenido.

El texto anterior no hace más que apoyarse en la idea de Berners Lee y su «Web Semántica», la misma tiene como uno de sus elementos o componentes el formato XML como base. Si bien esta Web no ha sido aplicada a niveles macro, se han realizado algunos experimentos probando la eficiencia de un buscador, trabajando en un ambiente diseñado con etiquetas XML y con la ayuda de una Ontología. Un ejemplo de ello es la ONTOWEB, objeto del artículo de Hyun Hee Kim, en la JASIST. En este artículo se representa y compara el rendimiento de un Sistema de Recuperación en ambiente Web, basado en ontologías, y un motor de búsqueda en Internet. La comparación se realiza bajo los parámetros relevancia y tiempo de búsqueda [10].

El punto de todo está en construir un documento mediante el uso de términos lo más representativo posible. Para ello es aconsejable el uso de un lenguaje controlado, sea un Tesauro, Ontología u otro sistema de control de metadatos: Dublin Core, EAD o Marc 21.

2) Profundidad en la búsqueda

El poder identificar textos le ofrece la ventaja de tener un lenguaje de marcas, que hace fácil identificar y extraer pedazos específicos de información con significados especiales, o sea, este tipo de lenguaje ayuda a recuperar información dentro de un documento. Dígase capítulos de un libro digital, párrafos dentro de un mismo documento, frases, segmentos, etcétera. Este tipo de recuperación es la tendencia que siguen muchos sistemas de recuperación automatizados en la actualidad [11].

Las tecnologías de este lenguaje, aplicadas a la recuperación de información, como XPointer, XPath o XQuery tienen la capacidad de «acceder» a partes específicas de un documento, mediante el desglose del mismo en forma de árbol. Estas herramientas han ido evolucionando, permitiendo no sólo recuperar documentos que traten sobre un tema, sino la parte del mismo que específicamente lo trata; demostrando así tener un enfoque más personalizado. Válido es agregar también que permite la consulta de otros documentos relacionados con esa temática a través de enlaces, lo cual también es muy positivo pues amplía la información que pueda tener un solo material.

3) Intercambio de Información

Las DTD son las encargadas de normalizar las etiquetas que debe tener un documento XML, lo cual es de vital importancia cuando una comunidad determinada está interesada en intercambiar información en formato electrónico, pues les permite crear su propio estándar para el intercambio de información.

Las DTD pueden ser creadas por un comunidad para un fin específico o pueden ser externas, es decir, existen DTD que por su relevancia son de uso internacional por grandes comunidades, tal es el caso de las DTD existentes de MARC21 para el campo de la bibliotecología y la EAD, para el campo de la archivística. La DTD Dublín Core también es de relevante importancia no solo para la comunidad profesional en temas de información, sino también para todos aquellos proyectos cuyo fin es organizar información electrónica.

Aún cuando cada comunidad pueda crear su propia DTD, es notorio destacar que desde la visión del mundo de la información, lograr el uso masivo de un número reducido de estándares, es decir DTD, ayuda a homologar y comprender mejor las estructuras de los documentos de cualquier comunidad.

Los Schemas, por su parte, tienen un desempeño crucial en la organización de la información en XML. Al estar dotadas de más facilidades que las DTD, permiten no solo normalizar las etiquetas que debe tener un documento XML, sino que también permiten controlar el contenido de estas etiquetas, aspecto relevante para poder implementar el *control de autoridad*, punto neurálgico en la organización de información.

Otro aspecto que lo hace potente para el mundo informacional es el hecho de permitir hacer referencias a diferentes schemas en un mismo documento, lo que permite la homologación de documentos que tengan diferentes Schemas o DTD; facilitando así el intercambio de información entre comunidades que usen diferentes tipos de estructuras de datos.

Los costos asociados a la compatibilidad entre diferentes estructuras para la organización de información son elevados y no todas las empresas están en condiciones de asumirlos, es por esta razón que actualmente se están definiendo esquemas por grupos sectoriales con intereses iguales. Esto dará como resultado la existencia de esquemas estándares avalados por asociaciones de empresas y organismos, que garanticen que cualquier usuario que las adopte trabaje con las mismas etiquetas.

Áreas del conocimiento también parecen ponerse de acuerdo para trabajar de la misma forma y delimitar sus etiquetas, es el caso de CML (Chemical Markup Language) quien atiende el sector químico, MathML (Mathematical Markup Language) el cual define datos matemáticos, así como el SMIL (Synchronized Multimedia Integration Language) quien define presentaciones de recursos multimedia). Dentro del sector informacional tenemos las ya nombradas EAD más las DTD Dublín Core y MARC21.

Como plantea Daniel Martínez Ávila en su *Tratamiento de documentos digitales con tecnologías XML:* «Uno de los aportes principales del XML es la posibilidad de aunar, por medio de la estandarización, el acceso e intercambio a muchos tipos de registros,…otra de las ventajas de XML es la posibilidad de presentar la misma información para diferentes perfiles de usuarios» [12].

De la misma forma Peng Liu y Amit Chetal sostienen que «en XML, las etiquetas pueden ser creadas para representar cada función para una línea particular de trabajo. Además, estas etiquetas XML pueden ser transmitidas vía http o cualquier otro protocolo con la intención de que otras entidades la utilicen como propia, crear nuevas o sobreponer las ya existentes.» [13]

Conclusiones

XML es una tecnología que permite la construcción de documentos autodescriptivos, partiendo del hecho de que describe el contenido de los mismos a partir de etiquetas descriptivas.

XML no sólo es un formato de intercambio, es un estándar para la organización de información electrónica.

La filosofía de las tecnologías XML defienden la organización normalizada de la información en la Web, con el uso de las DTD y los Schemas.

Desde un punto de vista tecnológico podemos decir que alrededor de XML existe un ordenado grupo de tecnologías con funciones muy bien definidas, que hacen del mismo una herramienta de gran potencia.

Separar en diferentes tecnologías aspectos como contenido, representación visual, recuperación y conexión a través del hiperenlace, hacen que se facilite la comprensión de cada una de ellas y permite que las mismas se potencialicen independiente.

Varios especialistas plantean que el hecho de permitir a cada usuario crear sus etiquetas puede ser una debilidad y no una fortaleza, cuestión que desde un punto de vista organizativo es muy acertada. Cuando se habla de intercambio y cooperación de información entre instituciones, es necesario que cada una de ellas estructure su información del mismo modo. Cuando se desea integrar sistemas con diferentes organizaciones, el lenguaje puede mostrar ciertas limitaciones. XML se concentra en la descripción de los datos, sin embargo únicamente se puede considerar útil una vez que la misma pueda ser utilizada en macro contextos y no en empresas e instituciones ocasionales.

Referencias bibliográficas

1) Katz, Howard et al. Xquery from the experts: A Guide to the W3C XML Query Language. Addison Wesley, Estados Unidos, p. 512, 2003.

2) Shanmugasundaram, Jayavel et al. Relational Databases for Querying XML Documents: Limitations and Opportunities. Department of Computer Sciences University of Wisconsin-Madison.

3) Peis, Eduardo y Ruiz-Rodríguez, Antonio A. «EAD (Encoded Archival Description): Desarrollo, estructura, uso y aplicaciones». [en línea]. Gijón: Universidad de Oviedo, 2004. Disponible en: <http://www.hipertext.net> [Consultado: 17 de marzo de 2007].

4) Brundage, Michael. XQuery: The XML Query Language. Addison Wesley, Estados Unidos, p. 544, 2004.

5) Barbero Paniagua, Angel. XML tutorial. [en línea]. Disponible en: <wwwdi.topaen.es/xmltutorial.com> [Consultado: 1 de febrero de 2007].

6) Goldfarb C. y Precod, P. Manual de XML. España: Prentice Hall, 1999. [en línea]. Disponible en: <wwwdi.goldfard.es/manualde xml.com> [Consultado: 08 de febrero de 2007].

7) Rivas Santos, Victor Manuel. Tutorial de Xpath. [en línea]. Disponible en: <wwwdi.ujaen.es/~vrivas.com> [Consultado: 21 de enero de 2007].

8) Gutiérrez, J.J., et al. XQuery. Sevilla: Universidad de Sevilla, 2005. [en línea]

Disponible en: <www.lsi.us.es/docs/informes/LSI-2005-02.pdf> [Consultado: 6 de junio de 2006].

9) Rodríguez Mederos, Mabel. «Xquery: Una aproximación al tema.» Proyecto de tesis inédito para optar por el grado de Doctor en Ciencias de la Información. Universidad de La Habana, Facultad de Comunicación, Ciudad de La Habana, 2006.

10) Hyun Hee Kim. ONTOWEB: implementing an Ontology- based web retrieval system. Journal of the American Society for Information Science and Technology (JASIST) 56(11), 2005.

11) Katz, Howard et al. Xquery from the experts: A Guide to the W3C XML Query Language. (ed.). Estado Unidos: Addison Wesley, p. 512 . 2003. ISBN 0-321-18060-7. En: Rodríguez Mederos, Mabel. «Xquery: Una aproximación al tema.» Proyecto de tesis inédito para optar por el grado de Doctor en Ciencias de la Información. Universidad de La Habana, Facultad de Comunicación, Ciudad de La Habana, 2006.

12) Daniel Martínez Ávila. Tratamiento de documentos digitales con tecnologías XML: Tendencias y usos de XML en Biblioteconomía y Documentación. 2007-2008

13) Liu, Peng y Amit Chetal. Trust-Based secure information sharing between Federal Goverment Agencies. Journal of the American Society for Information Science and Technology. (JASIST) 53 (3), 2005

Recibido: 15 de julio de 2009.
Aprobado en su forma definitiva: 18 de septiembre de 2009.

Lic. Raúl Entralgo Amaro
Biblioteca Nacional José Martí
Correo electrónico:
<raulentralgoamaro@gmail.com>

Lic. Jorge Dayán Aguiar Cedeño
Jefe del Dpto. de Canje Internacional,
Biblioteca Nacional José Martí
Correo electrónico:
<jdayan@infomed.sld.cu>

.