

Entorno de Publicación Web basados en XML

Hoy en día, la presencia en el Web es cada vez más relevante e importante para las empresas. Día a día se demandan más servicios en Internet. Por esto, son requeridos sistemas con gran capacidad de transformación y asimilación de las nuevas tendencias, de las nuevas tecnologías y del constante cambio del propio mercado. Una característica importante de este tema es lo que atañe a la presentación de la información en múltiples formas y dispositivos. Si tenemos en cuenta el manejo de la información como hasta actualmente se está haciendo y analizamos lo que significa para una empresa tener toda su presentación en páginas HTML, podemos notar que imponer un cambio de imagen en las propias páginas es una labor dispendiosa y debido a que se trata de "remendar" algo ya hecho, se torna en una tarea poco agradable.

Peor aún, si se trata de presentar la información de la empresa en un formato distinto al HTML, ya que además de crear la presentación se debe recolectar la información de la presentación en el formato que está manejando, es decir, el HTML.

Como podemos darnos cuenta, el tener la información en la misma capa en la que se presenta, genera inconvenientes grandes y desencadena una cantidad de factores negativos para las organizaciones tales como gastos en mantenimiento, mayor tiempo en los desarrollos, pérdida de tiempo y dinero en la capacitación de personas para que conozcan la estructura de presentación, etc.

Para las empresas en las cuáles los datos de presentación se generan de forma dinámica, el problema pasa del diseñador al programador. En estos casos el programador tiene que ir al código y cambiar la presentación, pero pensemos que en realidad éste no es el trabajo de un programador, es precisamente la presentación, trabajo del diseñador. Es claro que el diseñador no puede hacerlo (y tampoco debe, por que no es su oficio) ya que su línea de trabajo no está pensada para esto. Esto se da mucho en generación dinámica de aplicaciones que utilizan tecnologías del estilo de Servlets. Sin embargo nótese que el programador, al tener que modificar en su código fuente aspectos de presentación corre un riesgo alto de alterar el funcionamiento de la aplicación, lo cual cae en una solución peor e improductiva.

Una solución mejorada de los Servlets salió con la tecnología J2EE. En los *J2EE Beans* (que son la forma de presentar la información) se debería tener el mínimo código, es decir, el necesario para las clases que contienen toda la lógica del negocio. Por otro lado, con los *taglibs* (etiquetas XML para definir código) se posibilita crear páginas que no tienen una sola línea de código.

Bien, esto mejora las cosas; sin embargo se siguen teniendo básicamente tres problemas:

1. Si se quieren obtener distintas presentaciones, es necesario modificar el código Java que se encarga de dar formato a los datos
2. El mantenimiento puede no ser tan transparente ya que un diseñador puede alterar el código embebido en el HTML.
3. En ocasiones puede ocurrir que se incluya código Java más allá del necesario para que puedan funcionar correctamente los *beans*.

Sería óptimo poder tener productos de información que fueran independientes de las distintas formas de presentación y que si ese contenido se generara dinámicamente, ese dinamismo también fuera totalmente independiente. En pocas palabras se quiere *separar Contenido, Lógica y Presentación*.

Si se tuviera en un repositorio toda la información sin ninguna característica que la atara a algún tipo de presentación, se podrían implantar cada una de las vistas que se desearan sin que se debiera tener en cuenta alguna de las otras. Luego, el cambio o el mantenimiento de una de las vistas no le afectaría, sino a ella misma.

Es aquí, donde surgen los *Entornos de Publicación Web Basados en XML y XSL*.

Definición de XML

XML, es el estándar de Extensible Markup Language. XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. XML es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados.

En primer lugar para entenderlo bien hay que olvidarse un poco, sólo un poco de HTML. En teoría HTML es un subconjunto de XML especializado en presentación de documentos para la Web, mientras que XML es un subconjunto de SGML especializado en la gestión de información para la Web. En la práctica XML contiene a HTML aunque no en su totalidad. La definición de HTML contenido totalmente dentro de XML y por lo tanto que cumple a rajatabla la especificación SGML es XHTML (Extensible Hypertext Markup Language).

Desde su creación, XML ha despertado encontradas pasiones, y como para cualquier tema en Internet, hay gente que desde el principio se deja iluminar por sus expectativas, mientras otras muchas lo han ignorado.

XML fue creado al amparo del Word Wide Web Consortium (W3C) organismo que vela por el desarrollo de WWW partiendo de las amplias especificaciones de SGML.

Su desarrollo se comenzó en 1996 y la primera versión salió a la luz el 10 de febrero de 1998. La primera definición que apareció fue: *Sistema para definir validar y compartir formatos de documentos en la web.*

Durante el año 1998 XML tuvo un crecimiento exponencial, y con ello me refiero a sus apariciones en medios de comunicación, menciones en páginas web, soporte software, etc.

Respecto a sus objetivos son:

- XML debe ser directamente utilizable sobre Internet.
- XML debe soportar una amplia variedad de aplicaciones.
- XML debe ser compatible con SGML.
- Debe ser fácil la escritura de programas que procesen documentos XML.
- El número de características opcionales en XML debe ser absolutamente mínima, idealmente cero.
- Los documentos XML deben ser legibles por humanos y razonablemente claros.
- El diseño de XML debe ser preparado rápidamente.
- El diseño de XML debe ser formal y conciso.
- Los documentos XML deben ser fácilmente creables.
- La concisión en las marcas XML es de mínima importancia.

Esta especificación, junto con los estándares asociados (Unicode e ISO/IEC 10646 para caracteres, Internet RFC 1766 para identificación de lenguajes, ISO 639 para códigos de nombres de lenguajes, e ISO 3166 para códigos de nombres de países), proporciona toda la información necesaria para entender la Versión 1.0 de XML y construir programas de computador que los procesen.

Principales características de XML

- Es una arquitectura más abierta y extensible. No se necesitan versiones para que puedan funcionar en futuros navegadores. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en internet/intranet por medio de un validador de documentos (parser).
- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML (los RDF Resource Description FrameWork), en comparación a los atributos de la etiqueta <META> del HTML.

- Integración de los datos de las fuentes más dispares. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio PC como en una red local o extensa.
- Datos compuestos de múltiples aplicaciones. La extensibilidad y flexibilidad de este lenguaje nos permitirá agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Gestión y manipulación de los datos desde el propio cliente web.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- Se desarrollarán de manera extensible las búsquedas personalizables y subjetivas para robots y agentes inteligentes. También conllevará que los clientes web puedan ser más autónomos para desarrollar tareas que actualmente se ejecutan en el servidor.
- Se permitirá un comportamiento más estable y actualizable de las aplicaciones web, incluyendo enlaces bidireccionales y almacenados de forma externa (El famoso epígrafe "404 file not found" desaparecerá).
- El concepto de "hipertexto" se desarrollará ampliamente (permitirá denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del Lenguaje de enlaces extensible (XLL).
- Exportabilidad a otros formatos de publicación (papel, web, cd-rom, etc.). El documento maestro de la edición electrónica podría ser un documento XML que se integraría en el formato deseado de manera directa.

Estructura del XML

DTD (*Document Type Definition*): Definición del tipo de documento. Es, en general, un archivo/s que encierra una definición formal de un tipo de documento y, a la vez, especifica la estructura lógica de cada documento. Define tanto los elementos de una página como sus atributos. El DTD del XML es opcional. En tareas sencillas no es necesario construir una DTD, entonces se trataría de un documento "bien formado" (*well-formed*) y si lleva DTD será un documento "validado" (*valid*).

XSL (*eXtensible Stylesheet Language*): Es una familia de lenguajes que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio específico. Con estos

lenguajes podemos por ejemplo generar a partir de un documento XML un HTML con el contenido del XML.

Referencias

- Brogden, Bill, *Cocoon 2 programming: Web publishing with XML and Java*. 2003
- Tennison, Jeni, *Beginning XSLT*. 2002
- <http://cocoon.apache.org/>
- <http://es.tldp.org/Tutoriales/APACHE-COCOON-2/multiple-html/>