XML DTD

Jara Rodriguez Carlos Eduardo.
cj102509@ingenieria.sanmartin.edu.co
Fundación Universitaria San Martin.

Resumen— La necesidad de jerarquizar y estructurar correctamente la información, no sólo para almacenarla, sino también para acceder a ella, se ha convertido en una labor que ha cobrado especial relevancia en los últimos años, en los que se han producido importantes avances en este campo..

I. INTRODUCCIÓN

nicialmente se usaron las Definiciones del Tipo de Documento (DTDs) en el lenguaje SGML para describir el vocabulario necesario para identificar todos los elementos de que iba a constar el documento y para expresar la estructura. La aparición y el desarrollo del lenguaje XML, hizo que este lenguaje incorporara también las DTDs, no en vano, hay que recordar que el lenguaje XML no es sino un subconjunto del lenguaje SGML.

Pero las DTDs no satisficieron todas las necesidades inherentes a XML y pronto se vio necesario utilizar otros métodos más rigurosos y sofisticados para tratar la estructura y la semántica dentro de un documento XML. Así surgieron los Esquemas XML (XML Schema), como una forma de ampliación y mejora de las primitivas DTDs. Las DTDs y los Schemas son usados por los analizadores sintácticos o parsers para comprobar si un documento XML es válido.

Así pues, vemos que para proceder a la estructuración o especificación formal dentro de un documento XML existen distintas soluciones, entre las que cabe destacar principalmente dos: las DTDs y los XML Schemas. Pero veamos con más profundidad las diferencias entre la utilización de DTDs y Esquemas.

La Declaración de Tipo de Documento (DTD-Document Type Data):

Al definir el lenguaje XML ya nos referimos a la Definición del Tipo de Documento (Document Type Definition DTD) que, en resumen, cumple las siguientes funciones:

Una DTD especifica la clase de documento
Describe un formato de datos
Usa un formato común de datos entre aplicaciones
Verifica los datos al intercambiarlos
Verifica un mismo conjunto de datos
Una DTD describe
Elementos: cuáles son las etiquetas permitidas y cuál es el contenido de cada etiqueta
Estructura: en qué orden van las etiquetas en el documento
Anidamiento: qué etiquetas van dentro de cuáles
Los elementos de una DTD son los siguientes:

Elementos con “contenido ELEMENT”:
Un elemento tiene contenido ELEMENT, si solo puede contener a otros elementos, opcionalmente separados por espacios en blanco.
Elementos con “contenido TEXT”
Un elemento tiene contenido TEXT, si solo puede contener texto
(PCDATA = printable character data)
Elementos con “contenido MIXED”
Un elemento tiene contenido MIXED, si puede contener texto u otros elementos
Elementos con “contenido EMPTY”
Un elemento tiene contenido EMPTY, si no puede contener otros elementos
Y los atributos son los siguientes:

CDATA: texto
NMTOKEN: “abc...z0123..9-_:.” (tipo de lista)
NMTOKENS: NMTOKEN + espacios
ID: empezar con letra
IDREF: ser un ID
Así pues, la DTD especifica la clase de documento XML. Una DTD indica sólo qué elementos, atributos, etc; tiene un documento y cómo se anidan, pero no dice nada acerca de tipos de dato. El único tipo de dato que conoce es CDATA (texto plano), por tanto, las DTDs se quedan algo cortas y cuando se necesita algo más potente y rígido, se usa Schema

Recordemos que una DTD se puede guardar en un archivo de texto, como por ejemplo, en el archivo "list.dtd". Una DTD muy simple es la siguiente:

```
<!ELEMENT List (Item)+>
<!ELEMENT Item (#PCDATA)>
<!ATTLIST Item
  id CDATA IMPLIED
  color CDATA IMPLIED>
<!ELEMENT Separator EMPTY>
```

Veamos ahora el documento .xml que hace referencia a esa DTD:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE list SYSTEM "list.dtd">
<List>
 <Item>París</Item>
 <Item>Madrid</Item>
 <Separator />
 <Item color="rojo">Londres</Item>
</List>
```

XML Schema:

Al igual que las DTDs, los Schemas describen el contenido y la estructura de la información, pero de una forma más precisa. Los esquemas indican tipos de dato, número mínimo y máximo de ocurrencias y otras características más específicas.

Según la Especificación del W3C XML Schema (http://www.w3.org/XML/Schema), los esquemas expresan vocabularios compartidos que permiten a las máquinas extraer las reglas hechas por las personas. Los esquemas proveen un significado para definir la estructura, contenido y semántica de los documentos XML.

Un esquema XML (XML schema) es algo similar a un DTD, es decir, define qué elementos puede contener un documento XML, cómo están organizados, y qué atributos y de qué tipo pueden tener sus elementos, pero la utilización de schemas ofrece nuevas posibilidades en el tratamiento de los documentos.

La ventaja de utilizar los schemas con respecto a los DTDs son:

Usan sintaxis de XML, al contrario que los DTDs.
Permiten especificar los tipos de datos.
Son extensibles (esto es, permite crear nuevos elementos).

Por ejemplo, un schema nos permite definir el tipo del contenido de un elemento o de un atributo, y especificar si debe ser un número entero, una cadena de texto, una fecha, etc. Las DTDs no nos permiten hacer estas cosas.

Veamos un ejemplo de un documento XML, y su schema correspondiente:

```
<documento xmlns="x-schema:personaSchema.xml">
<persona id="fulanito">
<nombre>Fulano Menganez</nombre>
</persona>
</documento>
```

Como podemos ver en el documento XML anterior, se hace referencia a un espacio de nombres (namespace) llamado "x-schema:personaSchema.xml". Es decir, le estamos diciendo al analizador sintáctico XML (parser) que valide el documento contra el schema "personaSchema.xml".

El schema sería algo parecido a esto:

```
<Schema xmlns="urn:schemas-microsoft-com:xml-data"
xmlns:dt="urn:schemas-microsoft-com:datatypes">
<AttributeType name='id' dt:type='string' required='yes'/>
<ElementType name='nombre' content='textOnly'/>
<ElementType name='persona' content='mixed'>
<attrubyte type='id'/>
<element type='nombre'/>
</ElementType>
<ElementType name='documento' content='eltOnly'>
<element type='persona'/>
</ElementType>
</Schema>
```

El primer elemento del schema define dos espacios de nombre. El primero "xml-data" le dice al analizador (parser) que esto es un schema y no otro documento XML cualquiera. El segundo "datatypes" nos permite definir el tipo de elementos y atributos utilizando el prefijo "dt".

ElementType: Define el tipo y contenido de un elemento, incluyendo los sub-elementos que pueda contener.

AttributeType: Asigna un tipo y condiciones a un atributo.

attribute: Declara que un atributo previamente definido por AttributeType puede aparecer como atributo de un elemento determinado.

element: Declara que un elemento previamente definido por ElementType puede aparecer como contenido de otro elemento.

Así pues, es necesario empezar el schema definiendo los elementos más profundamente anidados dentro de la estructura jerárquica de elementos del documento XML. Es decir, tenemos que trabajar "desde dentro hacia fuera", o lo que es lo mismo, las declaraciones de tipo ElementType y AttributeType deben preceder a las declaraciones de contenido element y attribute correspondientes.

Un esquema también puede verse como una colección (vocabulario) de definiciones de tipos y declaraciones de elementos cuyos nombres pertenecen a un determinado espacio de nombres llamado espacio de nombres de destino. Los espacios de nombres de destino hacen posible la distinción entre definiciones y declaraciones de diferentes vocabularios. Por ejemplo, los espacios de nombres de destino

facilitarían la declaración del elemento element en el vocabulario del Esquema XML, y la declaración de element en un hipotético vocabulario de lenguaje químico. El primero es parte de espacio de nombres de destino http://www.w3.org/2001/XMLSchema, y el segundo es parte de otro espacio de nombres de destino.

Además, a medida que los esquemas se hacen más grandes, es posible y deseable dividir su contenido entre varios documentos esquema con el fin de facilitar su mantenimiento, control de acceso, y legibilidad.

Las Recomendaciones establecidas por el W3C en relación a los esquemas XML son las siguientes:

XML Schema Part 0: Primer. Es un documento no normativo que pretende ofrecer una fácil descripción de las funcionalidades de XML Schema y que está orientado a comprender de forma rápida cómo crear esquemas utilizando el lenguaje XML Schema. http://www.w3.org/TR/xmlschema-0/

XML Schema Part 1: Structures. Especifica la definición del lenguaje XML Schema y ofrece las herramientas para describir la estructura y constreñir el contenido de los documentos de XML 1.0, incluyendo las que tratan los espacios de nombre (XML Namespace). El lenguaje de esquemas que se representa en XML usa espacios de nombre, reconstruye sustancialmente y extiende las capacidades de las DTDs de los documentos del lenguaje XML 1.0. http://www.w3.org/TR/xmlschema-1/

XML Schema Part 2: Datatypes. Establece las herramientas para definir los tipos de datos que se usan en los esquemas XML y en otras especificaciones XML. El lenguaje de tipo de datos que se representa en XML 1.0, ofrece un conjunto de capacidades que se encontraban en las DTDs en XML 1.0 para especificar tipos de datos sobre elementos y atributos. http://www.w3.org/TR/xmlschema-2/

DTD versus Schema

Las limitaciones de la DTD son las siguientes:

Posee un lenguaje propio de escritura lo que ocasiona problemas a la hora del aprendizaje, pues no sólo hay que aprender XML, sino también conocer el lenguaje de las DTDs.

Para el procesado del documento, las herramientas y analizadores (parsers) empleados para tratar los documentos XML deben ser capaces de procesar también las DTDs .

No permite el uso de namespaces y estos son muy útiles ya que permiten definir elementos con igual nombre dentro del mismo contexto, siempre y cuando se anteponga un prefijo al nombre del elemento.

Tiene una tipología para los datos del documento extremadamente limitada, pues no permite definir el que un elemento pueda ser de un tipo número, fecha, etc. sino que sólo presenta variaciones limitadas sobre cadenas.

El mecanismo de extensión es complejo y frágil ya que está basado en sustituciones sobre cadenas y no hace explicitas las relaciones, es decir, que dos elemento que tienen definido el mismo modelo de contenido no presentan ninguna relación.

Estos problemas son superados gracias a la especificación de XML Schema, ya que los esquemas permiten un lenguaje mucho más expresivo y un intercambio de información mucho más robusto. Pero, aparte de solventar los problemas antes expuestos, XML Schema, permite una serie de ventajas adicionales:
XML Schema presenta una estructura de tipos mucho más rica. En la segunda parte de la especificación de XML Schema (XML Schema Part 2: Datatypes) se definen los tipos base que se pueden emplear dentro de esquema de XML, como ejemplo podemos destacar: byte, integer, bolean, string, date, sequence, etc. Este sistema de tipos es muy adecuado para importar y exportar sistemas de bases de datos y, sobre todo, distingue los requerimientos relacionados con la representación léxica de los datos y el conjunto de información dominante y subyacente.

Permite tipos definidos por el usuario, llamados Arquetipos. Dando un nombre a estos arquetipos, se pueden usar en distintas partes dentro del Schema.

Es posible agrupar atributos, haciendo más comprensible el uso de un grupo de aspectos de varios elementos distintos, pero con un denominador común, que deben ir juntos en cada uno de estos elementos.

Es posible trabajar con espacios de nombre, según la Especificación XML Schema Part 0: Primer, permitiendo validar documentos con varios namespaces.

Con XML Schema es posible extender Arquetipos de un modo específico, es decir permite lo que en términos de orientación a objetos se llama herencia. Considérese un esquema que extiende otro previamente hecho, se permite refinar la especificación de algún tipo de elemento para, por ejemplo, indicar que puede contener algún nuevo elemento del tipo que sea; pero dejando el resto del esquema antiguo completamente intacto.

En la siguiente imagen podemos comprobar la gran riqueza de los tipos de datos y la estructuración jerárquica que ofrece XML Schema:

REFERENCIAS

[1] http://www.hipertexto.info/documentos/dtds.htm

[2] http://www.w3schools.com/xml/default.ASP

Autores