

Quiz Taller 25%

1. Indique las diferencias entre DTD y XML SHEMA (Informe).

DTD VS SHEMA

DTD

- 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 procesamiento 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 explícitas las relaciones, es decir, que dos elementos que tienen definido el mismo modelo de contenido no presentan ninguna relación.

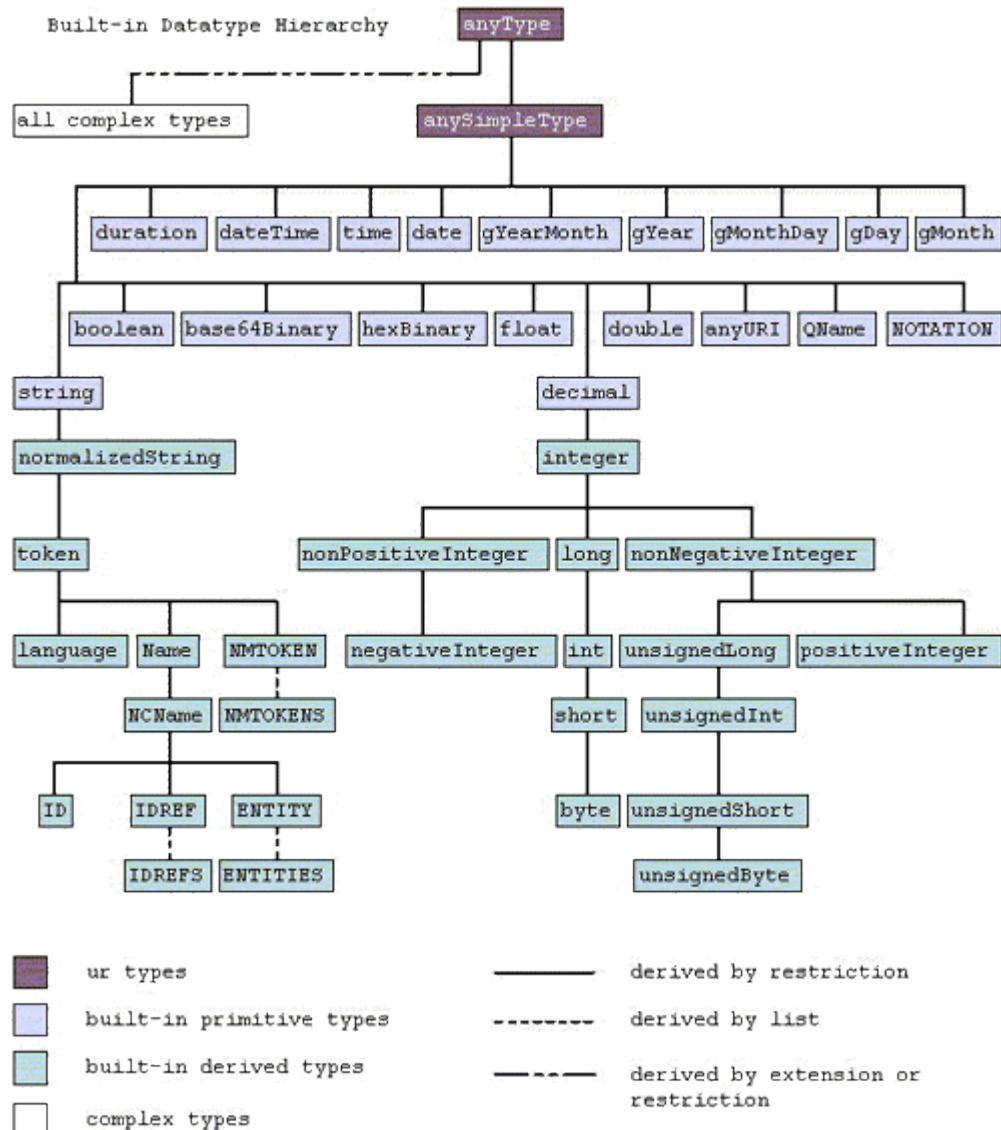
SHEMA

- 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, boolean, 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.

2. Especifique los tipos de datos soportados por DTD y XML SHEMA?

DTD	XML Schema	Comentarios
ELEMENT	<element>	Crea un vínculo entre un nombre y unos atributos, modelos de contenido y anotaciones
#PCDATA	Soportado como parte de un tipo simple	
ANY	<any>	Posee distintos comodines para un mayor conjunto de posibilidades. Existe también <anyAttribute> con comodines similares.
EMPTY	Soportado	Se elimina la existencia de elementos descendientes del actual, diferenciando de la presencia de un <i>string</i> vacío en un elemento.
Modelo de Contenido	<complexType>	
, (Conector de secuencia)	<sequence>	
(Conector de alternativas)	<disjunction>	
? (Opcional)	Soportado	Se han de emplear los atributos predefinidos de maxOccurs y minOccurs
+(Requerido y Repetible)	Soportado	Se han de emplear los atributos predefinidos de maxOccurs y minOccurs
*(Opcional y Repetible)	Soportado	Se han de emplear los atributos predefinidos de maxOccurs y minOccurs
ATTLIST	<attributeGroup>	Se pueden agrupar declaraciones de <attributes>
Tipo de atributo CDATA, ID, IDREF, NOTATION...	Tipos <simpleType>predefinidos	
ENTITY	NO Soportado	Las entidades son declaradas en declaraciones de marcas en el XML
ENTITY%Parameter	NO Soportada	<p>Las entidades paramétricas ofrecen un mecanismo de bajo nivel que permite distintas cosas, algunas de estas se han intentado cubrir en XML Schema:</p> <ul style="list-style-type: none"> *La separación entre <element> y <complexType> *Grupos de atributos *Grupos de modelos con nombre *Mecanismos de extensión y restricción de tipos *Los mecanismos de <import> e <include> para componer esquemas *El mecanismo de redefinición de elementos

Estructuración jerárquica que ofrece XML Schema



3. Es posible definir otro tipo de dato en DTD y XML SHEMA

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.

Con DTD No es soportado.

4. Especifique las restricciones y reglas para la definición de una estructura en DTD y XML SHEMA.

Reglas

XML SHEMA

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.

DTD

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

Restricciones

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.

5. Especifique librerías para utilizar DTD y XML SHEMA en (java, C#, Python, JavaScript PHP, C++)

Lenguaje	DTD	XML SHEMA
Java	xstream	Jetbrains
C#	XmlLite	
Python	The ElementTree XML	The ElementTree XML
JavaScript	Javascript XML Frameworks	Javascript XML Frameworks
PHP	simpleXML	simpleXML
C++	PugiXML	Xerces

Links

Xstream

<http://xstream.codehaus.org/>

XmlLite

<http://msdn.microsoft.com/es-es/magazine/cc163436.aspx>

PugiXML

<http://code.google.com/p/pugixml/>

MSXML

[MSXML at msdn.microsoft.com](http://msdn.microsoft.com)

Xerces

<http://xml.apache.org/xerces-c/index.html>

simpleXML

<http://www.php.net/downloads.php>

Javascript XML Frameworks

http://ajaxpatterns.org/Javascript_XML_Frameworks

The ElementTree XML

<https://docs.python.org/2/library/xml.etree.elementtree.html>

Jetbrains

<https://docs.python.org/2/library/xml.etree.elementtree.html>

Referencia bibliográfica y links de descarga de las librerías (W3CSchool) y recomendación.