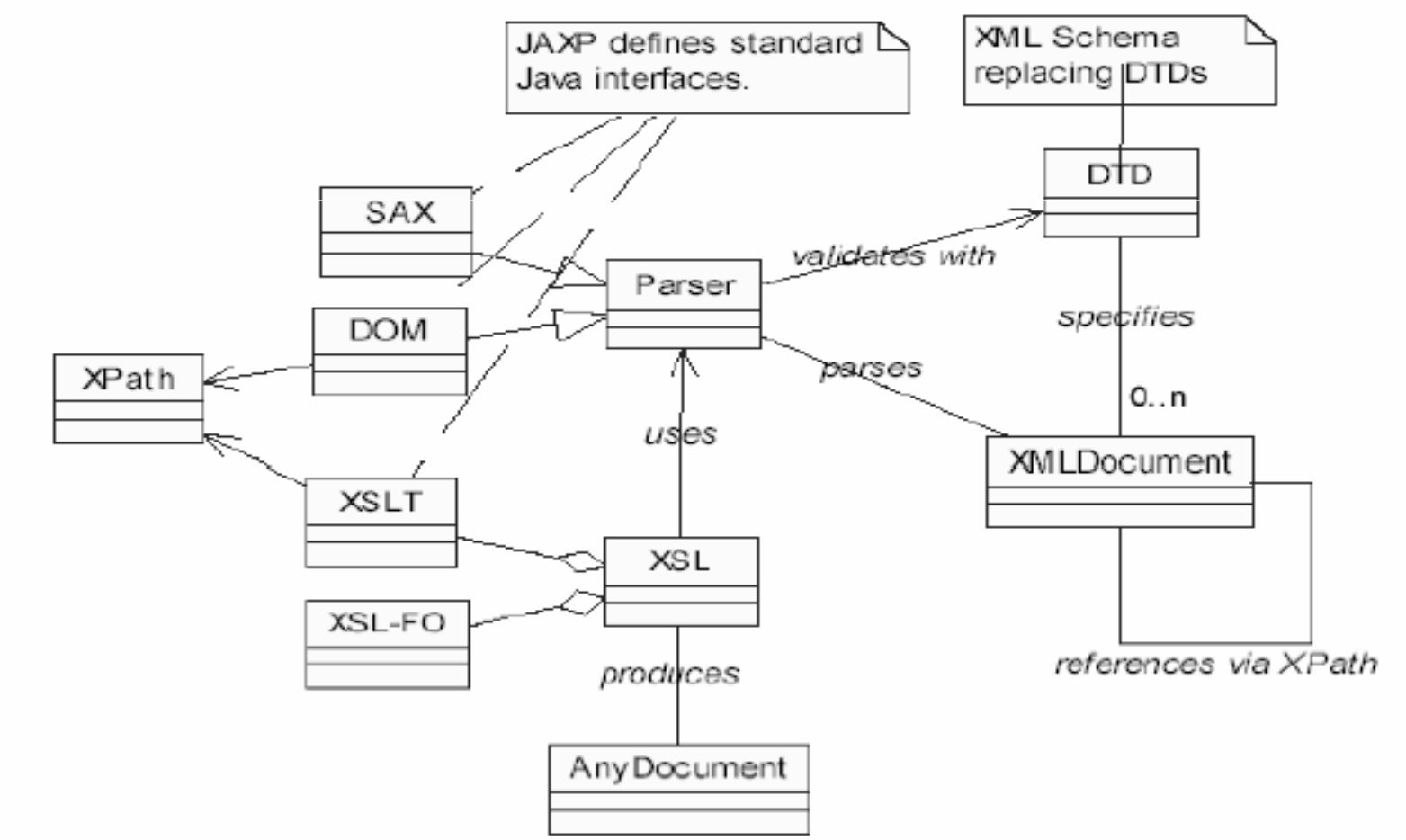
XSL: eXtensible Style Language

Anabel Fraga

Tabla de Contenidos

La Familia XML



Presentación en XML

- La presentación en HTML esta básicamente en los navegadores.
- Sería interesante *programar* la presentación (re-uso de código)
- Surgen las hojas de estilo:

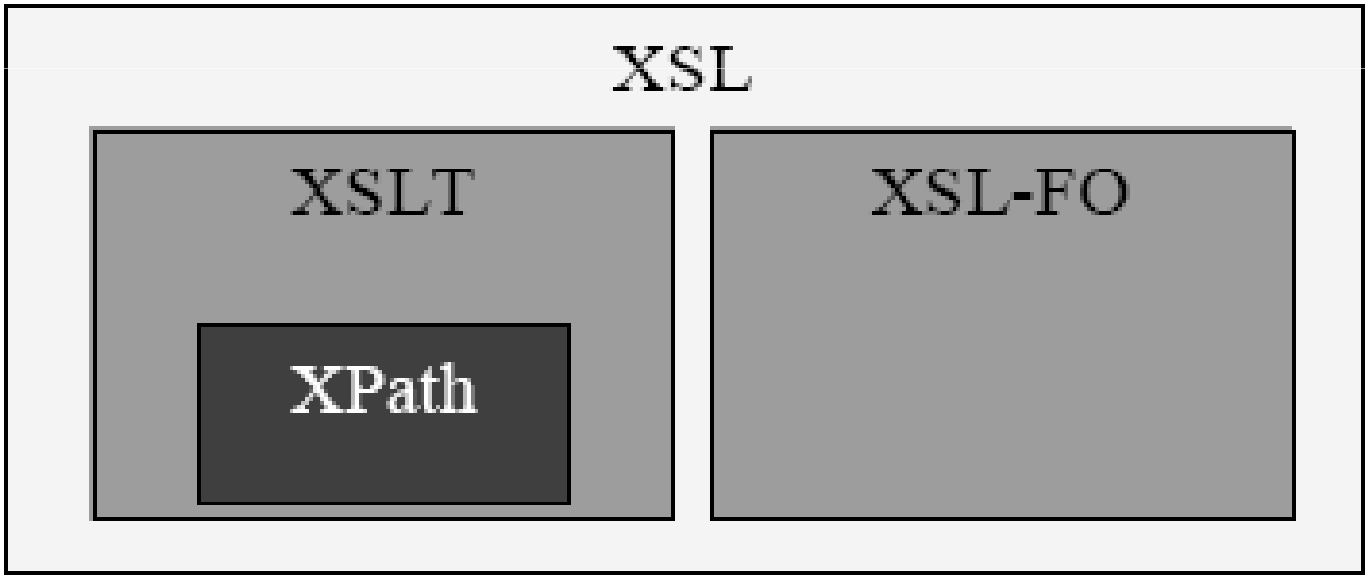
– **CSS**: Cascading Style Sheets (HTML)

– **XSL**: eXtensible Style Language (XML) (XML + DTD o XML Schema + Fichero de Estilo XSL)

XSL (I)

- XSL es una familia de recomendaciones para definir documentos XML, consiste de tres partes:
- XSL Transformations (XSLT)
  - Lenguaje para transformaciones XML
- XML Path Language (XPath)
  - Lenguaje de expresión utilizado por XSLT para referirse a partes de un documento XML. (XPath es también usado por XML Linking)
- XSL Formatting Objects (XSL-FO)
  - Vocabulario XML para especificar semántica de formato

XSL (II)



XSL (III)

	CSS	XSL
¿Puede usarse con HTML?	Si	No
¿Puede usarse con XML?	Si	Si
¿Tiene/Usa transformación?	No	Si
Sintaxis utilizada	CSS	XML

XSLT

- XSLT es la parte más importante de XSL. Usada para transformar un documento XML en otro documento XML, HTML, WML, etc.
- XSLT puede añadir nuevos elementos o incluso eliminarlos. Incluso hacer pruebas o tomar decisiones.
- Comúnmente nos referimos a un XML **source tree** transformado en un XML **result tree.**

Elementos

- Reglas de plantillas
  - **xsl:template**
  - **xsl:apply-templates**
  - **xsl:call-template**
- Salida
  - **xsl:value-of**
  - **xsl:copy-of**
  - **xsl:output**
- Control de flujo
  - **xsl:for-each**
  - **xsl:if**
  - **xsl:choose, xsl:when, xsl:otherwise**

Ejemplo

```
<?xml version="1.0“
      encoding="iso-8859-
      1"?>
<?xml-stylesheet
      type="text/xsl"
      href=“refrencia.xsl"?>
<fuente>
  <titulo> XSL</titulo>
  <autor>John Smith
  </autor>
</fuente>
```

XML

```
<h1>XSL</h1>

<h2>John Smith</h2>
```

HTML

XSL

John Smith

Final

```
<xsl:stylesheet version = '1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:template match="/">
    <h1>
      <xsl:value-of select="//titulo"/>
    </h1>
    <h2>
      <xsl:value-of select="//autor"/>
    </h2>
  </xsl:template>
</xsl:stylesheet>
```

XSLT

Reglas de Plantilla

- xsl:template
- xsl:apply-templates
- xsl:call-template

xsl:template (I)

Definición de regla de plantilla - *template rule*

<xsl:template match="*patrón*">

cuerpo

</xsl:template>

xsl:template (II)

- Es una regla de plantilla que:
  - define los nodos o subárboles de un árbol jerárquico XML dado (*source xml*) a los que es aplicable por medio de un ***patrón*** *(location path) en* ***XPath***
  - define una salida por medio de un ***cuerpo*** *que contiene:*
    - Texto de salida (ejemplo: HTML)
    - Aplicaciones de otras reglas de plantilla

xsl:apply-template

Aplicación de las reglas de plantilla (*template rule*)

<xsl:apply-templates/>

- Procesa todos los nodos hijos de tipo elemento, texto, comentario e instrucción de procesamiento
- **<xsl:apply-templates select="*expresión*"/>**

  sólo procesa los nodos seleccionados por medio de la ***expresión*** *en* ***XPath***

xsl:call-template (I)

- **<xsl:template name="*nombre*"/>** da nombre a una regla
- **<xsl:call-template name="*nombre*"/>** la llama por su nombre
- Puede haber parámetros o no según la definición que se de a la regla

xsl:call-template (II)

```
<xsl:template name=“CoordX">
<xsl:param name="x"/>
<xsl:value-of select="concat(‘(’,$x,‘)’)"/>
</xsl:template>

<xsl:variable name=“Y">
<xsl:call-template name=“CoordX"/>
<xsl:with-param name="x" select="$y"/>
</xsl:call-template>
</xsl:variable>
```

Aplicación de Reglas (I)

- Si hay una regla aplicable
  – Se aplica
- Si no hay regla aplicable
  – Se aplica la regla predefinida (*default*)
- Si hay varias reglas aplicables
  1) tienen prioridad las reglas de una hoja de estilo sobre las hojas que importa la misma
  2) se aplican las prioridades (definidas por el usuario o asignadas por el sistema: se aplica la más específica)
  3) algunos procesadores generan un error y otros aplican la última regla definida

Aplicación de Reglas (II)

Nodo	Regla Predefinida
Raíz (/)	Procesar Hijo
Elemento (*)	Procesar Hijo
Atributo (@*)	Copia atributo como texto
Texto (**text()**)	Copiar el texto
Comentario (**comment()**)	No hacer nada
Instr. Proc. (**processing-instruction()**)	No hacer nada
Espacio de Nombres	No hacer nada

Reglas de Salida

- xsl:value-of
- xsl:copy-of
- xsl:output

xsl:value-of

<xsl:value-of select="*expresión*">

se evalúa el valor (como cadena de caracteres) correspondiente a la ***expresión***

xsl:copy-of y xsl:copy

<xsl:copy-of select="*expresión*"/>

copia un subárbol como salida.

<xsl:copy select="*expresión*"/>

sólo copia el nodo en cuestión (sin los hijos).

xsl:output

- Controla el formato de salida.
- El procesamiento se realiza en dos fases:
  1. se genera el árbol resultado (**result tree**)
  2. se serializa el árbol (**xsl:output** tiene el control)

Ejemplo:

```
<xsl:output method="xml" indent="yes">
<xsl:output method="text" encoding="iso-8859-1">
```

Control de Flujo

- xsl:for-each
- xsl:if
- xsl:choose, xsl:when, xsl:otherwise

xsl:for-each

- Repite el procesamiento para cada uno de los nodos de un conjunto de nodos

<xsl:for-each select="*expresión*">

cuerpo

</xsl:for-each>

xsl:if

- Sólo se procesa si se cumple la condición

<xsl:if test="*expresión*">

cuerpo

</xsl:if>

Ejemplo:

<xsl:if test="position()=last()">

<hr/>

</xsl:if>

xsl:choose

- Permite la selección entre varias posibilidades de procesamiento

Ejemplo:

```
<xsl:choose>
<xsl:when test="x=28">Madrid</xsl:when>
<xsl:when test="x=08">Barcelona</xsl:when>
<xsl:otherwise>?</xsl:otherwise>
</xsl:choose>
```

Otros elementos

- Estructura
  - **xsl:stylesheet**
  - **xsl:include**
  - **xsl:import**
- Orden
  - **xsl:sort**
  - **xsl:number**
- Otros
  - **xsl:element**
  - **xsl:attribute**
  - **xsl:comment**
  - **xsl:processing-instruction**
  - **xsl:text**
- **…**

Ejercicio: Hola Mundo! (I)

- Hacer un documento XML y un XSLT que presente “Hola Mundo!” en HTML

Ejercicio: Hola Mundo! (II)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="hola.xsl"?>
<saludo>Hola, Mundo!</saludo>
```

Ejemplo: Hola Mundo! (III)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
        <head><title>Un Saludo</title></head>
        <body>
            <p><font color="red“ face=“arial”><strong>
                <xsl:value-of select="saludo"/>
            </strong></font></p>
        </body>
        </html>
    </xsl:template>

</xsl:stylesheet>
```

Procesadores XSLT

Para aplicar una hoja de estilo XSL a un documento XML podemos utilizar:

- ***MSXML3 SP4 (Internet Explorer)***
  http://www.microsoft.com/xml
- ***Saxon***
  http://users.iclway.co.uk/mhkay/saxon/
- ***Xalan***
  (Usado en Clases – Apache.org)
  http://xml.apache.org/xalan/overview.html

Documento XML

ie5.xsl
ie4.xsl
nav3.xsl

nokia.xsl
sony.xsl

edi_x.xsl
sap_y.xsl
flat_z.xsl

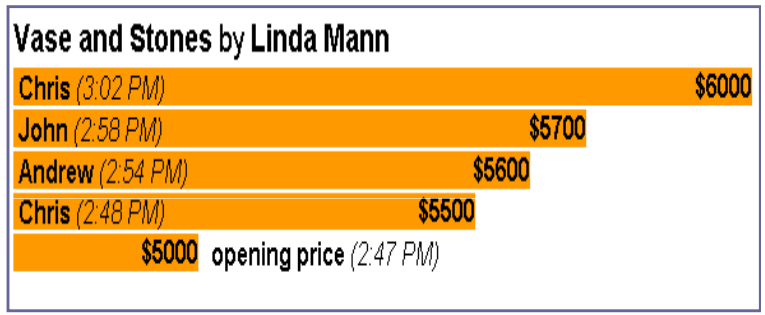
Documento XML

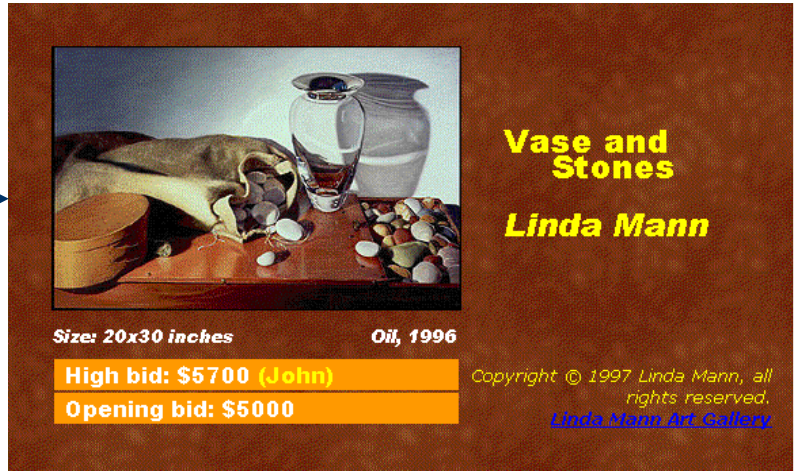
tabla.xsl

barra.xsl

arte.xsl

Price	Time	Bidder
$6000	3:02:22 PM	Chris
$5700	2:58:42 PM	John
$5600	2:54:32 PM	Andrew
$5500	2:48:08 PM	Chris
$5000	2:47:58 PM	opening price

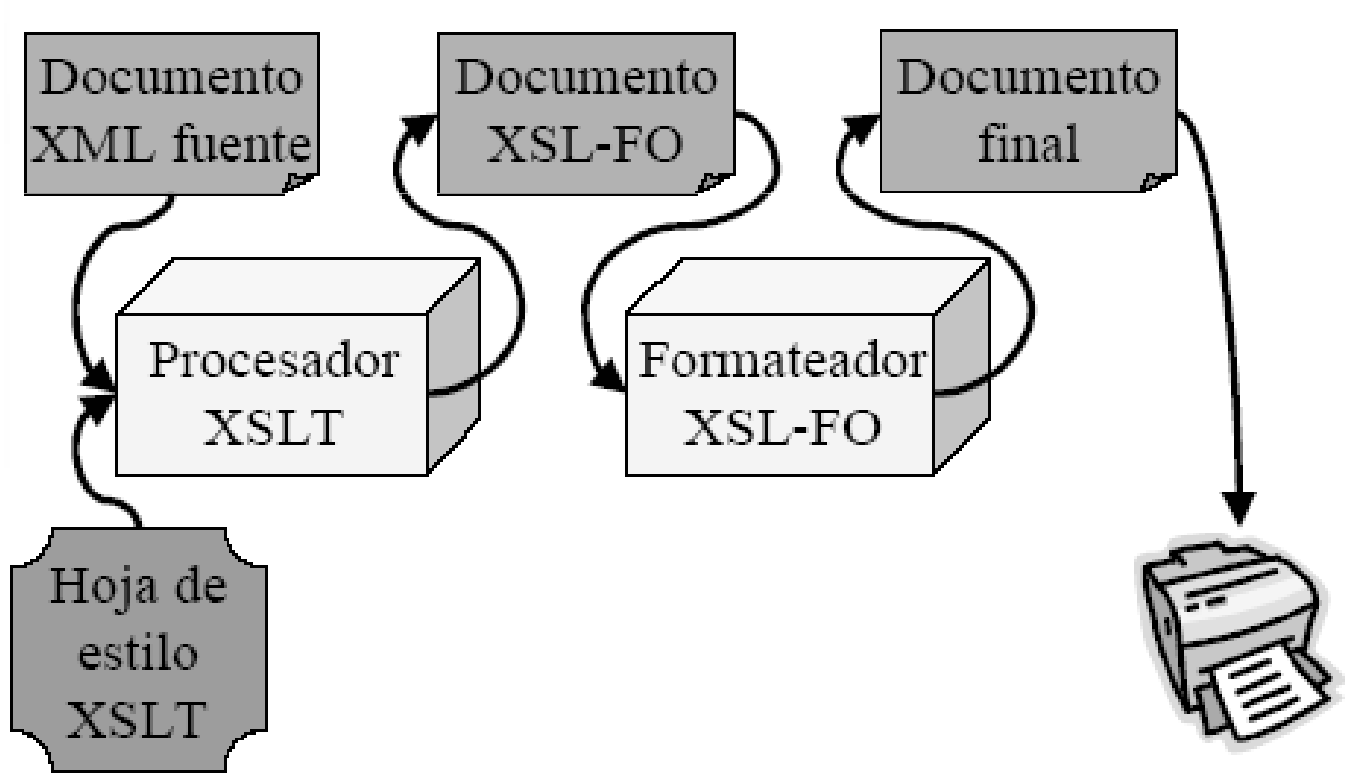




XSL-FO (Formatting Objects)

- Una hoja de estilo XSL-FO provee la especificación de un documento XML para su posterior transformación y utiliza para ello vocabulario de formateado.
- Permite generar impresiones de alta calidad como PDF o PS
- Útil para producir documentos visualmente elaborados y compuestos

XSL – XSLT – XSL-FO



Procesadores XSL-FO

- Antenna House XSL Formatter: Una herramienta interactiva para XSL-FO
  - http://www.AntennaHouse.com
- Adobe Acrobat: Una herramienta de visualización de documentos PDF. Creada por RenderX
  - http://www.RenderX.com

Referencias

- http://www.w3.org/TR/1999/REC-xslt-19991116
- http://www.w3.org/TR/xsl/
- http://www.zvon.org/index.php?nav_id=tutorials
- http://www.bayes.co.uk/xml/index.xml
- http://www.xml-web.de
- http://www.xsl-rp.dexml.coverpages.org/xsl.html
- http://www.ibiblio.org/xml
- http://xml.apache.org/fop (procesador)
- http://foa.sourceforge.net (editor)
- http://www.alphaworks.ibm.com/tech/xfc (editor y procesador)
- Tecnet Consultores. Estudio de XSLT por Juan Carlos Alonso.
- http://www.xml.com/pub/a/2002/03/20/xsl-fo.html?page=1

Ejercicio: XML, XSLT (PARA EL CUADERNILLO)

- Un Restaurante desea automatizar sus recetarios para ello desea crear un libro de recetas (al menos cinco recetas) en XML, y debe ser presentado por lo menos de dos formas diferentes:
  - En HTML para presentar a los clientes al llegar al local como Carta de Menú.
  - En HTML para presentar a los eventos culinarios a los que es nominado el local dada su relevancia en este ámbito.