

Tema 12

Transformació d' informació. XSLT

Objectius

- Justificar la necessitat de la transformació o conversió de documents XML i identificar-ne l'àmbit d' aplicació.
- Analitzar les tecnologies implicades en la transformació de la informació sobre documents XML i el seu mode de funcionament.
- Descriure l' estructura i sintaxi específica utilitzada en la conversió i adaptació de documents XML.
- Crear especificacions de transformació d' informació sobre documents XML.
- Identificar i caracteritzar eines específiques relacionades amb la conversió de documents XML.

Introducció

En els dos temes anteriors s' ha abordat la capacitat dels documents XML per emmagatzemar informació estructurada en format XML, així com les tècniques existents per poder accedir a la informació emmagatzemada. En aquest tema s' aborda l' estudi de les tècniques que permeten transformar o convertir un document XML en un altre document d' un tipus diferent com: HTML, PDF, text, RTF, etc. D' aquesta manera, el document resultant de la transformació pot presentar visualment la informació emmagatzemada en el document XML inicial d' una manera adequada. El contingut més important d'aquest tema es refereix a l'aplicació de la tecnologia XSLT (*eXtensible Stylesheet Language Transformations*). Els fulls de transformacions XSLT s' utilitzen juntament amb el llenguatge de consulta XPath que s' aplica per buscar i seleccionar la informació que es vol incorporar en el document resultant.

Índex

12.1 VISUALITZACIÓ DE DOCUMENTS XML	2
12.2 APLICACIÓ DE CSS EN DOCUMENTS XML	2
12.3 LA FAMÍLIA DE LLENGUATGES XSL	7
12.4 EL LLENGUATGE XSLT	9
12.5 EL PROCESSADOR XSLT. EINES ESPECÍFIQUES	23
Glossari de termes	24

12.1 VISUALITZACIÓ DE DOCUMENTS XML

Un document XML no disposa d'una visualització concreta sobre un navegador, perquè el seu contingut no defineix cap aspecte de visualització. Efectivament, els documents XML simplement emmagatzemen informació en format text. Com ja s'ha pogut comprovar en els temes anteriors, quan s'obre un document XML sobre un navegador es mostra el seu arbre de nodes, si és que el document XML està ben format.

Existeixen diferents maneres de presentar visualment les dades emmagatzemades en un document XML:

- **Utilitzant un full d'estil CSS.** Es construeix un full d'estil CSS específica que indica al navegador com ha de presentar la informació emmagatzemada en cada element del document XML. D'aquesta manera es produeix una conversió de cada element del document XML en un element visual d'HTML.
- **Utilitzant un full de transformacions XSL.** Es construeix un full de transformacions XSL per definir la forma com el processador XSL convertirà un document XML en un altre document d'un altre tipus com, per exemple: HTML, PDF, RTF, etc. Aquesta conversió es realitza atenent el formateig de la sortida que es defineix en el full de transformacions XSL.
- **Utilitzant un llenguatge de programació.** S'empra un llenguatge de programació procedural, com Java, C, Javascript, Basic, etc. per construir un programa que processi el document XML i obtingui el resultat de presentació visual desitjat.

En aquest tema s'estudien els dos primers mètodes. L'ús de fulls d'estil s'estudia breument a causa de les seves limitacions. S'incideix, especialment, en la creació i especificació dels fulls de transformacions XSL, mitjançant la utilització del llenguatge estàndard XSLT, com a mètode més adequat per realitzar conversions d'informació sobre els documents XML. El tercer mètode excedeix dels objectius d'aquest curs, tot i que pot tenir el seu interès en aplicacions concretes.

12.2 APLICACIÓ DE CSS EN DOCUMENTS XML

Un full d'estil CSS consisteix en un conjunt de regles d'estil amb prioritat que s'apliquen als diferents elements o etiquetes que formen un document HTML. Tanmateix, els fulls d'estil CSS també es poden aplicar a un document XML.

En el llenguatge de marques HTML les etiquetes estan definides i tenen una funció establerta. Per tant, els navegadors, o altres programes de processament, presenten els elements HTML d'una manera predeterminada mitjançant un estil inicial que pot adaptar-se a les característiques de visualització desitjades mitjançant fulls d'estil específiques. En els documents XML, les etiquetes que es poden definir són infinites i la seva funció depèn de cada cas concret. Per tant, els navegadors, o altres programes de processament, no poden tenir informació predeterminada per a la presentació dels documents XML. De manera que, la presentació visual dels documents XML depèn exclusivament de la informació d'estil que es pugui incorporar.

Un exemple senzill

A continuació, s'analitzarà un exemple d'ús senzill per comprendre el mecanisme d'aplicació d'un full d'estil CSS a un document XML. Segui el següent document XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="EjemploCSS_1.css" type="text/css"?>
<article>

  <!-- Capçalera del document -->
  <tituloP>
    Extensible Markup Language
  </tituloP>
  <autor>
    Wikipedia
  </autor>
  <intro>
    XML, sigles en anglès d'eXtensible Markup Language
  </intro>

  <!-- Cos del document -->
  <tituloA>
    Introducció
  </tituloA>
  <p>
    És un llenguatge de marques desenvolupat pel <n>World Wide
    Web Consortium (W3C)</n> utilitzat per emmagatzemar dades a
    forma llegible.
  </p>
  <p>
    Deriva del llenguatge <n>SGML</n> i permet definir la gramàtica
    de llenguatges específics
  </p>
  <tituloA>Utilitat</tituloA>
  <p> XML dona suport a bases de dades, sent útil quan diverses
    aplicacions s'han de comunicar entre si o integrar informació </p>
  <p> XML és una <n>tecnologia</n> senzilla que té al seu voltant
    d'altres que la <c>complementen</c> i la fan molt més gran i
    amb unes possibilitats molt més grans. </p>
  <tituloA>Evolució</tituloA>
  <p>Serà necessària la connexió al W3C per estar al dia en aquest
    constant procés d'evolució.</p>

  <!-- Comentaris del document -->
  <comentariF>
    Darrera data d'actualització: 8 d'Abril de 2014
  </comentariF>
  <comentario_oculto>
    Aquest document és un extracte
  </comentario_oculto>
</article>
```

En el codi anterior, es pot apreciar la instrucció de processament de tipus `<?xml-stylesheet>` que està situada en el pròleg del document. Aquesta instrucció permet relacionar el document XML amb l'arxiu extern de full d'estil CSS. El contingut de l'arxiu de full d'estil extern, anomenat `EjemploCSS_1.css`, és el següent:

```
tituloP { display:block; width:100%; font-size:20pt; font-family:Verdana; color: green;
border-bottom:double 4pt navy; text-align:left }
autor { display:block; font-size:12pt; font-family:Verdana; color:maroon;
text-align:right; margin:10pt }
intro { display:block; margin:25px; font-family:Verdana; font-size:10pt;
font-style:italic }
tituloA { display:block; font-family:Arial; margin-left:1cm; text-align:left;
font-size:14pt; font-weight:bold; margin-bottom:10pt }
comentarioF { display:block; margin:.2in; padding:5px; font-style:italic;
border:outset 3pt; text-align:center; font-size:10pt; background-color:aqua }
p { display:block; margin-bottom:10pt; font-family:Verdana,sans-serif; font-size:10pt;
text-align:justify; text-indent:1cm }
n { font-weight:bold; display:inline }
c { font-style:italic; display:inline }
comentario_oculto { display:none }
```

En el codi anterior, es pot observar com es defineixen les propietats d'estil associades a cadascun dels elements del document XML, per així donar el format de visualització desitjat a cada element. Cal tenir en compte que, de manera predeterminada, tots els elements XML són elements de línia. I, per tant, cal incloure la propietat `display`, de manera que es puguin definir els elements de bloc (block) o de línia (inline) de manera adequada.

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador.

Extensible Markup Language

Wikipedia

XML, siglas en inglés de eXtensible Markup Language

Introducción

Es un lenguaje de marcas desarrollado por el **World Wide Web Consortium (W3C)** utilizado para almacenar datos en forma legible.

Deriva del lenguaje **SGML** y permite definir la gramática de lenguajes específicos

Utilidad

XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información

XML es una **tecnología** sencilla que tiene a su alrededor otras que la *complementan* y la hacen mucho más grande y con unas posibilidades mucho mayores.

Evolución

Será necesaria la conexión al W3C para estar al día en este constante proceso de evolución.

Última fecha de actualización: 8 de Abril de 2014

Un altre exemple senzill

També és possible incloure etiquetes del llenguatge HTML dins dels documents XML per tractar d'ajudar a especificar la presentació visual del document. Per a això, s'inclou la definició de l'espai de noms corresponent d'HTML, de manera que es poden diferenciar les etiquetes específiques del llenguatge HTML d'aquelles altres etiquetes que poguessin tenir el mateix nom en el document XML. Segui el següent document XML:

```
<?xml version = " 1.0" encoding = " utf-8"? >
<?xml-stylesheet href="EjemploCSS_2.css" type="text/css"?>
<articulo xmlns:html="http://www.w3.org/1999/xhtml">
  <!-- Capçalera del document -->
  <html:a name="principi"></html:a>
  <tituloP>Extensible Markup Language</tituloP>
  <autor> <html:a href="mailto:fedrico@sanchez.org">Federico Sánchez</html:a> </autor>
  <intro>XML, sigles en anglès d'eXtensible Markup Language</intro>
  <!-- Cos del document -->
  <tituloA>Introducció</tituloA>
  <p> Desenvolupat pel <n>World Wide Web Consortium (W3C)</n> Deriva del llenguatge
    <n>SGML </n> i permet definir la gramàtica de llenguatges específics </p>
  <tituloA>Utilitat</tituloA>
  <p> XML és una <n> tecnologia </n> senzilla que té al seu voltant altres que la
    <c> complementen </c> i la fan molt més gran i amb unes possibilitats molt
    majors. </p>
  <html:br />
  <tituloA>Evolució</tituloA>
  <p> Serà necessària la connexió al W3C per estar al dia en aquest constant procés de
    evolució.</p>
  <html:br />
  <tituloA>Aclaració</tituloA>
  <p>Les normes necessàries per al desenvolupament de XML van publicant-se a poc a poc,
sent
    necessària la connexió al <html:a href="http://www.w3.org">W3C</html:a> per estar al
    dia en aquest constant procés d' evolució.</p>
  <!-- Comentaris del document -->
  <comentariF>Última data d'actualització: 8 d'Abril de 2014</comentariF>
  <html:p> Això és un paràgraf de html</html:p>
  <p>
  <html:center>
  <html:a href="#principio">[Principi de pàgina] </html:a>
  <html:font color="white">----</html:font>
  <html:a href="http://es.wikipedia.org/wiki/XML"> [Pàgina principal]</html:a>
  </html:center>
  </p>
  <comentariOcult>Aquest document és un extracte</comentariOcult>
</articulo>
```

En el codi anterior es pot apreciar que s'ha inclòs la definició de l'espai de noms d'HTML. I es pot observar que, per exemple, l'etiqueta `<html: p>` defineix un paràgraf d'HTML, mentre que l'etiqueta `<p>` fa referència a un element del document XML.

El contingut de l'arxiu de full d'estil CSS, anomenat `EjemploCSS_2.css`, és el següent:

```
@namespace "http://www.w3.org/1999/xhtml";
tituloP { display:block; width:100%; font-size:20pt; font-family:Verdana;
  color: green; border-bottom:double 4pt navy; text-align:left }
autor { display:block; font-size:12pt; font-family:Verdana; color:xarxa;
  text-align:right; margin:10pt }
intro { display:block; margin:25px; font-family:Verdana; font-size:10pt;
  font-style:italic }
tituloA { display:block; font-family:Arial; margin-left:1cm; text-align:left;
  font-size:14pt; font-weight:bold; margin-bottom:10pt }
comentariOF { display:block; margin:.2in; padding:5px; font-style:italic;
  border:outset 3pt; text-align:center; font-size:10pt;
```

```
background-color:aqua }
p { display:block; margin-bottom:10pt; font-family:Verdana,sans-serif;
font-size:10pt; text-align:justify; text-indent:1cm }
n { font-weight:bold; display:inline }
c { font-style:italic; display:inline }
comentarioOculto {display:none}
html|a { text-decoration:underline; color:xarxa; font-family:Verdana;
font-size:12pt }
html|p { color:orange; font-family:Verdana; font-size:24pt;
text-align:center }
```

En el codi anterior, es pot observar que també s'ha definit l'espai de noms d'HTML a l'arxiu de full d'estil extern. També es pot apreciar que, per donar estil a un element d'HTML s'utilitza el connector "|" per unir el prefix i el nom de l'element corresponent.

L'exemple anterior, produeix el següent resultat en la finestra de visualització del navegador:

Extensible Markup Language

[Federico Sánchez](#)

XML, siglas en inglés de eXtensible Markup Language

Introducción

Desarrollado por el **World Wide Web Consortium (W3C)** Deriva del lenguaje **SGML** y permite definir la gramática de lenguajes específicos

Utilidad

XML es una **tecnología** sencilla que tiene a su alrededor otras que la *complementan* y la hacen mucho más grande y con unas posibilidades mucho mayores.

Evolución

Será necesaria la conexión al W3C para estar al día en este constante proceso de evolución.

Aclaración

Las normas necesarias para el desarrollo de XML van publicándose poco a poco, siendo necesaria la conexión al [W3C](#) para estar al día en este constante proceso de evolución.

Última fecha de actualización: 8 de Abril de 2014

Esto es un párrafo de html

[\[Principio de página\]](#) [\[Página principal\]](#)

La utilització de fulls d'estil CSS com a mecanisme per definir la presentació visual de la informació continguda en un document XML té grans limitacions. Només és aconsellable si el contingut del document està format només per text i si les dades s'estructuren d'una forma senzilla i seqüencial. A més, no és possible seleccionar la informació i, consegüentment, es presentarà sempre tota la informació continguda en el document XML.

El **mètode recomanat per definir la presentació visual** de la informació emmagatzemada en documents XML **consisteix a aplicar transformacions XSL sobre els documents XML**. En els següents apartats s'estudia la família de llenguatges XSL i el llenguatge estàndard XSLT. El llenguatge XSLT és

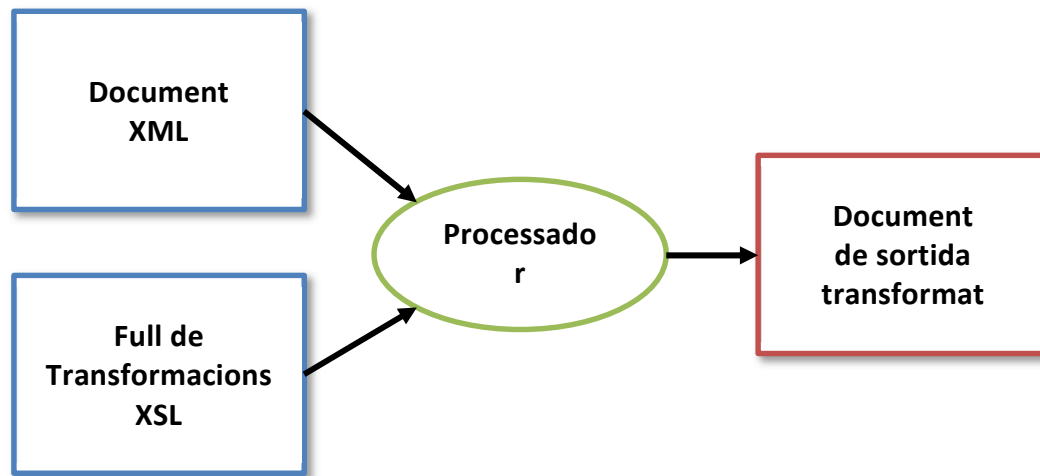
una tecnologia específica per realitzar la conversió i adaptació visual de documents XML que amplia les capacitats de presentació visual de la informació en format XML.

12.3 LA FAMÍLIA DE LENGUATGES XSL

La família de llenguatges XSL (*eXtensible Stylesheet Language*), o llenguatges extensibles de full d'estil, és una família de llenguatges basats en l'estàndard XML que permeten expressar com la informació continguda en un document XML ha de ser transformada, o formatejada, per aconseguir una presentació visual adequada en un determinat mitjà d'informació. XSL és, per tant, una família de llenguatges que permeten expressar fulls d'estil per a documents XML.

Alguns autors sostenen que XSL és per a XML, el que CSS és per a HTML. Per aquest motiu, els documents XSL solen denominar-se fulls d'estil de XML. Tanmateix, no s'ha de confondre CSS amb XSL, ni vol dir que un d'ells vagi a substituir l'altre, encara que tots dos tinguin la funció de facilitar la presentació d'informació en un format visual més llegible i atractiu. Des de l'inici de l'ús del llenguatge XML hi ha hagut un cert interès per proporcionar un mecanisme que faciliti la presentació visual dels documents XML al Web. Per aquest motiu es van adaptar algunes de les tecnologies existents en aquell moment, com per exemple CSS. No obstant això, en l'apartat anterior s'han pogut comprovar les importants limitacions de l'ús del llenguatge d'estil CSS en aquest àmbit. Aquestes limitacions, que estan marcades per la manca de construccions de sentències de control i de filtres d'informació adequats, fan que l'ús conjunt de CSS amb XML sigui insuficient, especialment si es compara amb els avantatges que proporciona l'aplicació de les tècniques de transformació de documents que proporciona XSL. Es pot concloure que la família de llenguatges XSL permet definir fulls d'estil adequats per a la presentació visual dels documents XML, mentre que CSS utilitza un mètode de presentació que està adaptat específicament al llenguatge HTML. A la pràctica, ambdós llenguatges, XSL i CSS, són complementaris, de manera que XSL se sol utilitzar amb documents XML i CSS amb document HTML.

La família de llenguatges XSL s'utilitzen per extreure informació de documents XML i generar altres documents amb formats de sortida diferents, com ara: XML, HTML, PDF, text plànel, RTF, PostScript, etc. El procés habitual consisteix a convertir les dades contingudes en un document XML en un altre document amb un format de sortida adequat, a partir d'un full de transformació, seguint un mecanisme de transformació com el que es mostra en la següent il·lustració.



La família de llenguatges extensibles de full d'estil, XSL, són un conjunt de recomanacions que defineixen la transformació i presentació de documents XML. Consta de tres parts o llenguatges:

- **XSLT (*eXtensible Stylesheet Language Transformations*) o Transformacions XSL.** És un llenguatge que s'utilitza per transformar documents XML en format HTML o XML.
- **XPath (*XML Path Language*).** És un llenguatge d'expressions utilitzat per XSLT per referenciar o accedir a parts de documents XML. Es va estudiar en un tema anterior.
- **XSL-FO (*eXtensible Stylesheet Language Transformations Formatting Objects*).** És un llenguatge de format per a documents XML. Permet generar documents en formats PDF, RTF, PostScript, etc. a partir de documents XML. Cal destacar que el document XSL-FO pot contenir tant les dades a presentar com la descripció del format a aplicar.

Utilitzant els tres components anteriors, es pot:

- Transformar un document XML en un altre document diferent d'un altre tipus de format.
- Filtrar i ordenar la informació continguda en un document XML.
- Definir parts en un document XML.
- Formatejar un document XML sobre la base dels valors de la informació emmagatzemada.
- Extreure dades contingudes en un document XML per aplicar una transformació XSL-FO per generar arxius PDF.

La família de llenguatges XSL va ser estandarditzada per l'organisme W3C en diverses etapes: la primera, el novembre de 1999, incloïa XSLT i XPath; la segona, l'octubre del 2000, incloïa la recomanació XSL-FO; posteriorment, s'han publicat noves versions de cadascun dels llenguatges. Per obtenir més informació, es pot visitar l'adreça web: <http://www.w3.org/Style/XSL>.

Aquest tema se centra, principalment, en l' estudi del llenguatge XSLT i en la creació de fulls de transformacions XSLT, també anomenats fulls d' estil XSLT, que permeten generar documents en formats de sortida XML, HTML o text a partir de documents XML. El llenguatge XSLT utilitza el llenguatge XPath per seleccionar la informació continguda en el document XML origen.

12.4 EL LLENGUATGE XSLT

El llenguatge XSLT és l' especificació concreta que, dins la família de llenguatges XSL, serveix per definir el procés de transformació d' un document XML en un altre document de destinació, principalment, en format XML o HTML. Un cop aclarides les diferències entre XSL i XSLT, cal tenir en compte que, habitualment, ambdós termes s' usen a la pràctica com a equivalents per referir-se al procés de transformació de documents XML.

La transformació XSLT es realitza convertint cada element d' un document XML en un altre element de sortida diferent. El llenguatge XSLT permet afegir o eliminar elements a la sortida, reordenar o recol·locar elements, avaluar condicions, seleccionar quins elements del document XML origen es vol mostrar, etc. durant el procés de transformació.

El llenguatge XSLT utilitza el llenguatge XPath per especificar o referenciar les parts del document XML que compleixen un o més patrons predefinits. Quan es troba una coincidència amb els patrons predefinits, el processador de XSLT transformarà la part coincident del document XML origen sobre el document destí. Les parts que no són coincidents no es transformen i, conseqüentment, quedaran en el document destí sense cap canvi. Tot i que el llenguatge XPath va ser dissenyat, en el seu origen, per ser utilitzat de forma independent, té la seva utilitat més completa com apareix embogit en un full de transformació XSLT, de manera que permet crear filtres d' informació que permeten obtenir els elements a traspasar al document de sortida.

Un primer exemple de transformació XSLT d'un document XML en un document HTML

A continuació, s' inclou un exemple de transformació XSLT d' un document XML, anomenat *EjemploXSLT_1.xml*, en un document HTML de destinació. Sigui el següent document XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="EjemploXSLT_1.xsl" type="text/xsl"?>
<strings>
  <s>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </s>
  <s>
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquid ex ea commodi consequat.
  </s>
  <s>
    Quis aute iure reprehenderit in voluptate velit esse cillum
    dolore eu fugiat nulla pariatur. Excepteur sint obcaecat
    cupiditat non proident, sunt in culpa qui officia deserunt
    mollit anim id est laborum.
```

```
</s>  
</strings>
```

En el codi anterior es pot observar com en el pròleg del document XML s'ha inclòs una instrucció de processament de tipus `<?xml-stylesheet href="..." type="text/xsl"?>`. Aquesta instrucció especifica l'accés a l'arxiu que defineix el full de transformacions XSLT associat al document XML. L'accés a aquest arxiu associat, que té extensió `.xsl`, s'especifica mitjançant l'atribut `href`, el valor del qual ha d'incloure el nom de l'arxiu, així com la ruta relativa des de la ubicació actual de l'arxiu corresponent al document XML a transformar.

El següent codi, que es correspon amb el full de transformacions XSLT denominada *EjemploXSLT_1.xsl*, transforma el document XML anterior en un document HTML de sortida. En aquest exemple, l'arxiu de transformacions, *EjemploXSLT_1.xsl*, i l'arxiu del document XML, *EjemploXSLT_1.xml*, hauran d'estar situats a la mateixa carpeta.

```
<?xml version="1.0" encoding="utf-8" ?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:template match="/">  
    <html>  
      <head>  
        <title>Ejemplo 1 de XSLT</title>  
      </head>  
      <body>  
        <h1>Lorem Ipsum</h1>  
        <xsl:for-each select = "/strings/s">  
          <p>  
            <xsl:value-of select="." />  
          </p>  
        </xsl:for-each>  
      </body>  
    </html>  
  </xsl:template>  
</xsl:stylesheet>
```

Atès el codi anterior, el processador o motor XSL realitzarà les següents accions:

- Cerca l'arrel del document XML a l'arbre de nodes del document XML origen. Aquesta recerca correspon amb el codi del llenguatge XSLT: `<xsl:template match="/">`.
- Compara l'arrel del document XML amb el contingut de la plantilla en el full d'estil.
- Genera la sortida dels elements HTML sobre el document de sortida.
- Processa els elements XSLT de la plantilla. El codi `<xsl:for-each select = "/strings/s">` fa que se seleccionin tots els elements `<s>` que siguin fills de l'element arrel, `<strings>`. I, a més, fa que per a cadascun dels elements seleccionats es generi un paràgraf, element `<p>`, que conté el valor de l'element, al qual s'accedeix mitjançant l'element `<xsl:value-of...>` de XSLT.

En obrir el document XML de l'exemple anterior amb un navegador Web modern que inclogui suport XSLT, s'obté el següent resultat a la finestra de visualització del navegador. El processador de XSLT

incorporat utilitzarà el full d'estils XSLT per realitzar la transformació del document XML d'origen en un document HTML destí, produint-se el següent resultat.

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed eiusmod tempor incididunt ut labore et dolore magna aliqua.

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquid ex ea commodo consequat.

Quis aute iure reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint obcaecat cupiditat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Estructura d'un full de transformacions XSLT

En el codi de l'arxiu *EjemploXSLT_1.xsl* anterior, es pot apreciar l'estructura bàsica d'un full de transformacions XSLT, també anomenat full d'estils XSLT, que és un arxiu de text compost per diverses parts:

- Una **declaració de document XML**: `<?xml version="1.0" encoding="utf-8" ?>`, ja que el llenguatge XSLT és un llenguatge basat en l'estàndard XML.
- Un **element arrel del full de transformacions XSLT** que es defineix amb l'etiqueta `<xsl:stylesheet>`, tot i que també es pot utilitzar l'etiqueta `<xsl:transform>`. Ambdues etiquetes són equivalents i vàlides. De fet, sol emprar-se indistintament el terme full d'estils XSLT, o bé, el terme full de transformacions XSLT. Aquesta etiqueta, a més de la declaració l'element arrel del full d'estils XSLT, identifica la versió de XSL utilitzada, així com espai de noms recomanat pel W3C per al llenguatge XSL.
- Diverses **porcions de codi HTML** que ajuden a formar el nou document HTML de sortida o de destinació a partir de la transformació del document XML original.
- Diversos **elements XSLT** que apareixen com a fills de l'element arrel `<xsl:stylesheet>`. Aquests elements formen part de l'espai de noms de prefix `xsl:`. Per exemple, poden ser els següents elements XSLT: `<xsl:template>`, `<xsl:for-each>`, `<xsl:value-of>`, `<xsl:output>`, `<xsl:sort>`, `<xsl:if>`, `<xsl:strip-space>`, `<xsl:preserve-space>`, `<xsl:import>`, etc.

Elements XSLT d'un full de transformacions XSLT

Defineix el valor d'un paràmetre que es passa a una plantilla.

Elemento XSLT	Vicente Aracil Miralles <varacil@iesmarenosttrum.com>
apply-imports	Desenvolupament d'Aplicacions Web

Elemento XSLT	Vicente Aracil Miralles <varacil@iesmarenostrum.com>
apply-templates	Busca y aplica la regla de plantilla adecuada al elemento actual o a los nodos secundarios del elemento actual.
attribute	Crea un nodo de atributo y lo añade al elemento resultante.
attribute-set	Define un conjunto de atributos con nombre.
call-template	Invoca a una plantilla con nombre
choose	Se utiliza junto con <xsl:when> y <xsl:otherwise> para expresar pruebas condicionales múltiples.
comment	Crea un nodo de comentario en el árbol de resultados.
copy	Crea una copia del nodo actual (sin atributos ni nodos secundarios).
copy-of	Crea una copia del nodo actual (con atributos y nodos secundarios).
decimal-format	Define los caracteres y símbolos que se utilizarán al convertir números en cadenas, con la función format-number().
element	Crea un nodo de elemento en el documento de salida.
fallback	Especifica un código alternativo para ejecutar, si el procesador no admite un elemento XSLT
for-each	Aplica una plantilla repetidamente, aplicándola para cada nodo del conjunto de nodos especificado.
if	Especifica una plantilla que se aplicará sólo si la condición especificada es verdadera.
import	Importa el contenido de una hoja de estilos XSLT externa que está definida en otro archivo. Una hoja de estilos importada tiene una precedencia inferior que la hoja de estilos en la que se importa.
include	Incluye el contenido de una hoja de estilos XSLT externa que está definida en otro archivo. Una hoja de estilo incluida tiene la misma precedencia que la hoja de estilos en la que se incluye.
key	Declara una clave que se puede utilizar con la función key().
message	Escribe un mensaje de texto. Se utiliza para informar de errores.
namespace-alias	Sustituye el prefijo de un espacio de nombres por otro prefijo.
number	Determina la posición entera del nodo actual y formatea un número.
otherwise	Especifica una acción predeterminada para el elemento <xsl:choose>.
output	Define el formato del documento de salida.
param	Declara un parámetro local o global.
preserve-space	Conserva los espacios en blanco en los elementos definidos.
processing-instruction	Escribe una instrucción de procesamiento a la salida.
sort	Especifica los criterios de ordenación de la salida para listas de nodos seleccionadas por <xsl:for-each> o <xsl:apply-templates>.
strip-space	Se eliminan los espacios en blanco en los elementos definidos.
stylesheet	Define el elemento raíz de una hoja de estilo XSLT.

Elemento XSLT	Vicente Aracil Miralles <varacil@iesmarenostrum.com>
template	Se utiliza para construir plantillas. Una plantilla contiene las reglas de transformación a aplicar sobre el nodo especificado.
text	Escribe texto literal a la salida.
transform	Hace la misma función que <xsl:stylesheet>. Ambos son equivalentes.
value-of	Escribe el valor de un nodo seleccionado en la salida.
variable	Declara una variable local o global.
when	Especifica una acción para el elemento <xsl:choose>.
with-param	Define el valor de un parámetro que se pasa a una plantilla.

A continuació, s' estudien els elements XSLT bàsics que solen utilitzar-se més habitualment per realitzar transformacions XSLT sobre documents XML. Es pot trobar més informació visitant l'adreça web: [https://msdn.microsoft.com/es-es/library/ms256058\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/ms256058(v=vs.120).aspx)

Element <xsl:template>

L' element <xsl: template> és un dels elements principals i més utilitzats del llenguatge de transformacions XSLT. Com a consideració prèvia, abans d' iniciar l' estudi aquest element, cal comprendre que un full d' estil XSLT està format per una sèrie de **plantilles o templers de transformació**.

Cada element <xsl:template> representa una plantilla que defineix una sèrie d'accions de transformació XSL. Aquestes accions es realitzaran si el patró definit a la plantilla coincideix amb algun element o elements de l' arbre de nodes del document XML. Els elements del document XML que no són coincidents amb el patró especificat no es transformaran i, conseqüentment, quedaran en el document destí sense cap canvi. Per especificar el patró al qual s' aplica la plantilla s' empra l' atribut *match*, el valor del qual és una expressió **XPath** que defineix els elements del document XML que es veuran afectats per la plantilla.

Per exemple, sigui el següent document XML que conté informació relativa a l' expedient acadèmic d' un determinat estudiant.

```

<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet href="EjemploXSLT_2.xsl" type="text/xsl"?>
<expedient alumne="José Rodríguez García">
  <signatura id="1">
    <nom>Programació</nom>
    <nota>Notable</nota>
  </signatura>
  <signatura id="2">
    <nom>Sistemes operatius</nom>
    <nota>Excel·lent</nota>
  </signatura>
  <signatura id="3">
    <nom>Disseny d'interfícies</nom>
    <nota>Suspens</nota>
  </signatura>
  <signatura id="4">

```

```
<nom>Bases de dades</nom>
<nota>Bien</nota>
</assignatura>
</expedient>
```

El següent full d'estil XSLT associat, l'arxiu del qual es denomina *EjemploXSLT_2.xsl*, transforma el document XML anterior en un document HTML de sortida.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Ejemplo</title>
        <style>
          .tabla { display: table; border: 2px solid black; }
          .fila { display: table-row; }
          .filaEncapçat { display: table-row; background-color:#9acd32;
                        font-weight: bold; }
          .celda { display: table-cell; border: 1px solid black; padding:3px; }
        </style>
      </head>
      <body>
        <h2>Expedient acadèmic</h2>
        <h3> Alumne: <xsl:value-of select="expedient/@alumno"></xsl:value-of> </h3>
        <div class="taula">
          <div class="filaEncapçat">
            <div class="celda">Asignatura</div>
            <div class="celda">Nota</div>
          </div>
          <xsl:for-each select="expedient/assignatura">
            <div class="fila">
              <div class="celda">
                <xsl:value-of select="nom"/>
              </div>
              <div class="celda">
                <xsl:value-of select="nota"/>
              </div>
            </div>
          </xsl:for-each>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

En el codi anterior, es pot apreciar que s'ha definit una única plantilla. En efecte, només es disposa d'un element `<xsl: template>` que s'aplica a l'element arrel del document XML, tal com especifica el valor de l'atribut `match = "/"`. També es pot apreciar com s'especifica l'estil de la composició tabular del document HTML resultant, mitjançant l'ús de propietats d'estil CSS que s'apliquen als elements HTML que incorpora la plantilla definida. La funcionalitat i l'ús de la resta dels elements XSLT que apareixen en el codi, més concretament dels elements `<xsl: value-of>` i `<xsl: for-each>`, s'analitzarà a continuació.

En obrir el document XML anterior, s'obté el següent resultat.

Expediente académico

Alumno: José Rodríguez García

Asignatura	Nota
Programación	Notable
Sistemas operativos	Excelente
Diseño de interfaces	Suspenso
Bases de datos	Bien

Element <xsl:value-of>

S' utilitza per afegir a la sortida, el valor de l' element o atribut que se selecciona mitjançant l' atribut select, i de tots els seus descendents. El valor de l' atribut select és una expressió XPath que especifica quin element o atribut s' extreu del contingut del document XML.

En el codi de l' exemple anterior, es pot apreciar l' ús que es fa de l' element <xsl: value-of> per extreure i afegir al document HTML de sortida els valors del nom de l' alumne, els noms de les assignatures i les notes corresponents.

Element <xsl:for-each>

Permet realitzar un bucle que es pot utilitzar per seleccionar cadascun dels elements del document XML que pertanyin a un conjunt de nodes determinat.

En el codi XSLT de l' exemple anterior, es pot apreciar que s' inclou un element <xsl: for-each> per recórrer totes les assignatures de l' expedient acadèmic de l' alumne.

Element <xsl:sort>

Envia a la sortida, la informació del document XML de partida ordenada segons un criteri d' ordenació. L' atribut select permet indicar l' element pel qual es realitzarà l' ordenació. L'atribut order permet definir l'ordre ascendent (*ascending*) o descendent (*descending*).

Continuant amb l' exemple anterior, es pot incloure un element <xsl: sort> dins l' element <xsl: for-each> de la plantilla del full d' estils XSLT, per obtenir l' expedient acadèmic de l' alumne ordenat, alfabèticament i de forma descendent, pel nom de les assignatures.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
```



```

<head>
  <title>Ejemplo</title>
  <style>
    .tabla { display: table; border: 2px solid black; }
    .fila { display: table-row; }
    .filaEncapçat { display: table-row; background-color:#9acd32;
                    font-weight: bold; }
    .celda { display: table-cell; border: 1px solid black; padding:3px; }
  </style>
</head>
<body>
  <h2>Expedient acadèmic</h2>
  <h3> Alumne: <xsl:value-of select="@alumno"></xsl:value-of> </h3>
  <div class="taula">
    <div class="filaEncapçat">
      <div class="celda">Asignatura</div>
      <div class="celda">Nota</div>
    </div>
    <xsl:for-each select="expedient/assignatura">
      <xsl:sort select="nom"/>
      <div class="fila">
        <div class="celda">
          <xsl:value-of select="nom"/>
        </div>
        <div class="celda">
          <xsl:value-of select="nota"/>
        </div>
      </div>
    </xsl:for-each>
  </div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
  
```

Aplicant el full d'estils XSLT anterior, s'obté el següent resultat.

Expediente académico

Alumno: José Rodríguez García

Asignatura	Nota
Bases de datos	Bien
Diseño de interfaces	Suspenso
Programación	Notable
Sistemas operativos	Excelente

Element **<xsl:if>**

S' utilitza per produir un comportament condicional. Permet aplicar una plantilla només en el cas que la condició especificada sigui veritable. L' element `<xsl: if>` només permet incorporar una única condició i actuar d' una determinada manera quan la condició es compleix.

Continuat amb l' exemple anterior, el següent full d' estils XSL inclou un element `<xsl: if>` dins l' element `<xsl: for-each>` per obtenir només les assignatures aprovades.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <title>Ejemplo</title>
        <style>
          .tabla { display: table; border: 2px solid black; }
          .fila { display: table-row; }
          .filaEncapçat { display: table-row; background-color:#9acd32;
                        font-weight: bold; }
          .celda { display: table-cell; border: 1px solid black; padding:3px; }
        </style>
      </head>
      <body>
        <h2>Expedient acadèmic</h2>
        <h3> Alumne: <xsl:value-of select="expedient/@alumno"></xsl:value-of> </h3>
        <div class="taula">
          <div class="filaEncapçat">
            <div class="celda">Assignatura</div>
            <div class="celda">Nota</div>
          </div>
          <xsl:for-each select="expedient/assignatura">
            <xsl:sort select="nom"/>
            <xsl:if test="nota != 'Suspens'">
              <div class="fila">
                <div class="celda">
                  <xsl:value-of select="nom"/>
                </div>
                <div class="celda">
                  <xsl:value-of select="nota"/>
                </div>
              </div>
            </xsl:if>
          </xsl:for-each>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Aplicant el full d' estils XSLT anterior, s' obté el següent resultat.

Expediente académico

Alumno: José Rodríguez García

Asignatura	Nota
Bases de datos	Bien
Programación	Notable
Sistemas operativos	Excelente

Element <xsl:choose>

L'element <xsl:choose>, juntament amb els elements <xsl:when> i <xsl:otherwise>, permet expressar un comportament condicional múltiple que inclogui diverses opcions i una opció predeterminada.

Per exemple, sigui el següent document XML que conté informació relativa a l'expedient acadèmic d'un estudiant. En aquest nou exemple, es pot apreciar que les notes són numèriques.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="EjemploXSLT_3.xsl"?>
<alumno nom="Antonio Sala Monfort">
  <numero_expediente>A1267</numero_expediente>
  <expedient>
    <asignatura>
      <nom>Matemàtiques</nom>
      <nota>4.00</nota>
    </asignatura>
    <asignatura>
      <nom>Lengua</nom>
      <nota>7.00</nota>
    </asignatura>
    <asignatura>
      <nom>Història</nom>
      <nota>9.50</nota>
    </asignatura>
    <asignatura>
      <nom>Física i Química</nom>
      <nota>5.00</nota>
    </asignatura>
  </expedient>
</alumno>
```

El següent full de transformació XSL associat, l'arxiu del qual es denomina *EjemploXSLT_3.xsl*, transforma el document XML anterior en un document HTML. Es pot apreciar que s'utilitzen les etiquetes <xsl:choose>, <xsl:when> i <xsl:otherwise> per definir una estructura condicional múltiple que permet incorporar informació textual per a cadascuna de les notes. A més, es pot observar que

s'empren les seqüències d'escapament > i < per representar els caràcters de major que, '>', i de menor que, '<', en les condicions que s'inclouen en l'atribut test.

```
<?xml version = " 1.0" encoding = " UTF-8"? >
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <head>
        <style>
          .tabla { display:table; border:2px solid black; font-family:Arial; width:30% }
          .fila { display:table-row; }
          .filaEncapçat { display:table-row; background-color:#9acd32;
                        font-weight:bold; }
          .celda { display:table-cell; border:1px solid black; padding:3px; }
        </style>
      </head>
      <body>
        <h2>Expedient acadèmic</h2>
        <h3>Alumne: <xsl:value-of select ="alumne/@nombre"/></h3>
        <div class ="taula">
          <div class ="filaEncapçat">
            <div class ="cel.la">Signatura</div>
            <div class ="cel.la">Nota</div>
          <div class ="celda">Descripció</div>
        </div>
        <xsl:for-each select ="alumne/expedient/assignatura">
          <div class ="fila">
            <div class ="celda">
              <xsl:value-of select ="nom"/>
            </div>
            <div class ="celda">
              <xsl:value-of select ="nota"/>
            </div>
            <div class ="celda">
              <xsl:choose>
                <xsl:when test = ". /nota < 5">
                  <span style ="color:red;">Suspens</span>
                </xsl:when>
                <xsl:when test = ". /nota >= 5 and . /nota < 7">
                  Aprovat
                </xsl:when>
                <xsl:when test = ". /nota >= 7 and . /nota < 9">
                  Notable
                </xsl:when>
                <xsl:otherwise>
                  Excel·lent
                </xsl:otherwise>
              </xsl:choose>
            </div>
          </div>
        </xsl:for-each>
      </div>
    </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

L' exemple anterior, produeix el següent resultat en el navegador.

Expediente académico

Alumno: Antonio Sala Monfort

Asignatura	Nota	Descripción
Matemáticas	4.00	Suspenso
Lengua	7.00	Notable
Historia	9.50	Excelente
Física y Química	5.00	Aprobado

Element <xsl:apply-templates>

L'element <xsl:apply-templates> és, juntament amb l'element <xsl:template>, un dels elements importants del llenguatge de transformacions XSLT. Indica al processador XSLT que trobi la plantilla a aplicar, segons el tipus i el context de cada node seleccionat. La plantilla **s'aplica a l'element actual o als elements fills de l'element actual**. L'atribut select permet especificar els nodes que se seleccionen per ser processats, mitjançant una expressió XPath. Si l'atribut select s'omet, llavors tots els nodes descendents del node actual seran processats.

Per exemple, sigui el següent document XML que conté informació relativa a l'expedient acadèmic d'una alumna.

```
<?xml version = " 1.0" encoding = " UTF-8"? >
<?xml-stylesheet type="text/xsl" href="EjemploXSLT_4.xsl"?>
<alumno nom="Marina Sánchez Borrell">
  <numero_expediente>A1267</numero_expediente>
  <curs>2016/2017</curs>
  <expedient>
    <asignatura>
      <nom>Matemàtiques</nom>
      <nota>3.00</nota>
    </asignatura>
    <asignatura>
      <nom>Lengua</nom>
      <nota>6.00</nota>
    </asignatura>
    <asignatura>
      <nom>Història</nom>
      <nota>8.50</nota>
    </asignatura>
    <asignatura>
      <nom>Física i Química</nom>
      <nota>5.50</nota>
    </asignatura>
  </expedient>
</alumno>
```

El següent full de transformació XSL associat, l'arxiu del qual es denomina *EjemploXSLT_4.xsl*, transforma el document XML anterior en un document HTML.

```
<?xml version="1.0" encoding="utf-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <html>
      <title>Ejemplo</title>
      <body>
        <h2>Expedient acadèmic</h2>
        <h3>Alumne: <xsl:value-of select="alumne/@nombre"/></h3>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <xsl:template match="curs">
    Curs acadèmic: <xsl:value-of select="."/><hr />
  </xsl:template>

  <xsl:template match="numero_expediente">
    Expedient: <xsl:value-of select="."/><br />
  </xsl:template>

  <xsl:template match="expedient">
    <xsl:apply-templates select="assignatura"/>
  </xsl:template>

  <xsl:template match="assignatura">
    <p>
      <xsl:apply-templates select="nom"/>
      <xsl:apply-templates select="nota"/>
    </p>
  </xsl:template>

  <xsl:template match="nom">
    Signatura: <xsl:value-of select="."/>
    <br />
  </xsl:template>

  <xsl:template match="nota">
    Nota: <span style="font-weight:bold;">
      <xsl:value-of select="."/>
    </span>
    <br />
  </xsl:template>

</xsl:stylesheet>
```

En l'exemple anterior, es pot apreciar que en l'etiqueta `<xsl:apply-templates/>` que apareix en la plantilla corresponent a l'element arrel, no s'especifica l'atribut `select`, la qual cosa provoca que es processin tots els elements descendents de l'element arrel del document XML. A més, es pot observar que s'utilitzen elements `<xsl:template>` per especificar la plantilla de transformació corresponent a cadascun dels elements que han de ser processats. També, es pot observar que la vinculació entre l'element `<xsl:apply-templates/>` i l'element `<xsl:template>` corresponent a cada

element, s'estableix a través del nom de l'element que s'especifica en el valor de l'atribut select i de l'atribut match, respectivament.

L' exemple anterior, produeix el següent resultat en el navegador.

Expediente académico

Alumno: Marina Sánchez Borrell

Expediente: A1267

Curso académico: 2016/2017

Asignatura: Matemáticas

Nota: **3.00**

Asignatura: Lengua

Nota: **6.00**

Asignatura: Historia

Nota: **8.50**

Asignatura: Física y Química

Nota: **5.50**

Com es pot apreciar en l' exemple anterior, les característiques específiques d' ús de l' element `<xsl: apply-templates>` fan que no sigui necessari incloure bucles de repetició del tipus `<xsl: for-each>` per recórrer el conjunt de nodes a processar en cada cas. De manera que, l' ús de l' element `<xsl: apply-templates>` dóna lloc a una organització del full d' estils XSLT molt més modular, simple i llegible que facilita la seva creació i manteniment. Per aquests motius se sol recomanar l'ús de l'element `<xsl:apply-templates>` per crear fulls d'estil XSLT.

Regles de resolució de conflictes en l' aplicació de plantilles

Cal tenir sempre present que l' ordre d' aplicació del processament XSLT queda marcat pel recorregut de l' arbre de nodes del document XML d' origen. El processador XSLT només aplica una plantilla o regla a cada node. Si s' han definit dues o més plantilles o regles per a un mateix node, el processador XSLT només aplica l' última plantilla a aparèixer. Llevat del cas en què s' hagin definit diverses regles per a un mateix node, l' ordre en què apareixen les plantilles en el full de transformacions XSLT no és representatiu.

S' han estudiat els elements XSLT bàsics i més utilitzats, atès que no és possible abordar un estudi més complet. Els continguts d' aquest tema poden servir per comprendre alguns dels aspectes més importants de la transformació de documents XML. I també, serviran per poder aprofundir en el seu estudi, en el cas que fos necessari. Es pot obtenir una referència més àmplia sobre els elements XSLT i les seves funcionalitats visitant les següents adreces Web:

- W3schools: https://www.w3schools.com/xml/xsl_intro.asp

- MSDN de Microsoft: [https://msdn.microsoft.com/es-es/library/ms256058\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/ms256058(v=vs.120).aspx)

12.5 EL PROCESSADOR XSLT. EINES ESPECÍFIQUES

Un processador XSLT és una aplicació informàtica que processa un document XML mitjançant un full de transformacions XSLT. Quan el processador XSLT llegeix un document XML genera una representació d'aquest document com un arbre de nodes. El processador XSLT va recorrent i processant node a node sobre el document XML d'origen. El node que s'està tractant en un moment donat es denomina node de context o node actual.

Els processadors XSLT es poden classificar en dos tipus: en línia i instal·lables. Alguns dels processadors XSLT en línia més utilitzats poden ser els següents:

- <http://online-toolz.com/tools/xslt-transformation.php>
- <http://www.shell-tools.net/index.php?op=xslt>

Com a processadors XSLT instal·lables es pot esmentar Xalan (<http://xml.apache.org/xalan-j/>). I alguns editors de codi que inclouen un processador XSLT que permeten realitzar transformacions XSLT com, per exemple: Altova StyleVision, <oXygen/>, XML Copy Editor, etc.

D'altra banda, els navegadors Web actuals incorporen un processador XSLT, de manera que quan s'obre un document XML que té associat un full d'estils XSLT, s'apliquen les transformacions XSLT definides i es genera el document de sortida HTML corresponent sobre la finestra de visualització del navegador. De manera que n'hi ha prou amb utilitzar un navegador modern, per la qual cosa no és necessari utilitzar altres eines específiques.

Procés de transformació XSLT

En aquest tema s'estudia, bàsicament, la transformació XSLT des d'un document XML a un document HTML de sortida. Aquest tipus de transformacions són les més comunes, perquè faciliten la presentació visual de la informació continguda en un document XML. Per realitzar aquest tipus de transformacions es precisen dos documents: el document XML de partida i el full d'estils XSLT. Un cop es disposa d'aquests dos documents, una transformació XSL es pot dur a terme en tres ubicacions diferents:

- En el servidor Web. Emprant per a això un processament basat en llenguatge de servidor, com: PHP, ASP.NET, JSP, etc. El processament en el servidor Web permet aplicar el full d'estils XSLT al document XML i enviar el document HTML al client com a resposta.
- En el client Web. El client Web és un navegador Web, com: Mozilla Firefox, Microsoft Edge, Google Chrome, etc. Els navegadors Web incorporen un processador XSLT que realitza les transformacions en obrir un document XML que té associada un full d'estil XSLT.

- En una aplicació específica. Es poden emprar diversos llenguatges de programació per crear una aplicació específica que realitzi una transformació XSLT requerida. Igualment, també es poden utilitzar programes genèrics per realitzar les transformacions XSLT.

El procés de transformació XSLT consta de diverses etapes que són les següents:

1. El full de transformacions XSLT és analitzat.
2. El document XML és processat i convertit en una estructura d'arbre.
3. El processador XSLT es posiciona en l'arrel del document XML que és el context inicial.
4. Els elements que no formen part de l'espai de noms de prefix `xsl:`, com poden ser les etiquetes HTML, es passen directament al flux de sortida, sense realitzar sobre ells cap tipus de conversió. Aquests elements es denominen elements de resultat literal.
5. S'aplica el processat XSLT segons el recorregut de l'arbre de nodes del document XML. El processador XSLT només aplica una plantilla o regla a cada node. Si s'han definit dues o més plantilles o regles per a un mateix node, el processador XSLT només aplica l'última plantilla a aparèixer. Excepte en aquest cas, l'ordre en què apareixen especificades les plantilles en el full de transformacions XSLT és irrelevant.

Glossari de termes

Arbre de nodes. Un arbre és una estructura de dades jeràrquica. Al seu torn, un arbre és un cas particular de graf. Els grafs permeten estudiar les interrelacions existents entre els elements que interactuen entre si unes amb les altres dins d'un conjunt. Aquests conceptes estan definits en la teoria de grafs. Els grafs són àmpliament utilitzats en la disciplina informàtica i la seva aplicació és particularment útil quan s'empren juntament amb els llenguatges de consulta, com XPath, per accedir a la informació estructurada de forma jeràrquica continguda en els documents XML.

Estructura jeràrquica dels documents XML. Els tipus de node que poden aparèixer en un document XML són: element, atribut, text, comentari, instrucció de processament i espai de noms. Les relacions entre els nodes d'un document XML són similars a les que poden trobar-se en un arbre genealògic: pares, fills, ascendents, descendents, germans, etc. Els comentaris i les instruccions de processament són part de l'arbre de nodes i han de ser tinguts en compte a l'hora de comptar nodes o iterar sobre ells.