

W10

Mettre en oeuvre XML/XSL et la diffusion de l'information



Nom : Heinzer

Prénom : Michael

Classe : 3I1

Section : Informatique

Date : 12.01.2007

Table des matières

1. Introduction	5
Ce module nous donne les compétences de gérer des fichiers XML. Par exemple de gérer du HTML à partir d'un fichier XML et XSLT. Nous faisons aussi des PDFs. Une autre partie est de gérer la génération du XML avec Java.	
2. Résumer de toutes les notions acquises	6
DTD.....	
XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée. La norme XML définit ainsi une définition de document type appelée DTD (<i>Document Type Definition</i>), c'est-à-dire une grammaire permettant de vérifier la conformité du document XML. La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.	
Schema.....	6
XML Schema est un langage de description de format de document XML permettant de définir la structure d'un document XML. Un schéma XML est lui-même un fichier XML. La connaissance de la structure d'un document XML permet notamment de vérifier la validité de ce document. Un fichier de description de structure (XML Schema Description en anglais, ou fichier XSD) est donc lui-même un document XML. ...	
XPath	7
XPath est une syntaxe (non XML) pour désigner une portion d'un document XML. Initialement créé pour fournir une syntaxe et une sémantique aux fonctions communes à XPointer et XSL, XPath a rapidement été adopté par les développeurs comme un petit langage d'interrogation.	
XSLT	7
XSLT (eXtended Stylesheet Language Transformations), défini au sein de la recommandation XSL du W3C, est un langage de transformation XML de type fonctionnel.	
L'objectif principal est la transformation d'un document XML vers un autre, ou un dialecte XML (XHTML, XSL-FO, HTML, etc.). Cependant, le langage XSLT permet aussi les transformations vers tout autre type de document, au format texte ou dans un format binaire (bien que ceci ne soit pas nativement prévu par la recommandation XSLT).....	
</xsl:stylesheet>	
JDOM	10

Le principal rôle de DOM est de fournir une représentation mémoire d'un document XML sous la forme d'un arbre d'objets et d'en permettre la manipulation. JDOM est une version pour Java. 10

3. Conclusion 12

4. Bibliographie 13

<http://www.commentcamarche.net/xml/xmltd.php3> 13

<http://www.google.ch> (definitons)..... 13

1. Introduction

Ce module nous donne les compétences de gérer des fichiers XML. Par exemple de gérer du HTML à partir d'un fichier XML et XSLT. Nous faisons aussi des PDFs. Une autre partie est de gérer la génération du XML avec Java.

2. Résumer de toutes les notions acquises

DTD

XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée. La norme XML définit ainsi une définition de document type appelée DTD (*Document Type Definition*), c'est-à-dire une grammaire permettant de vérifier la conformité du document XML. La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.

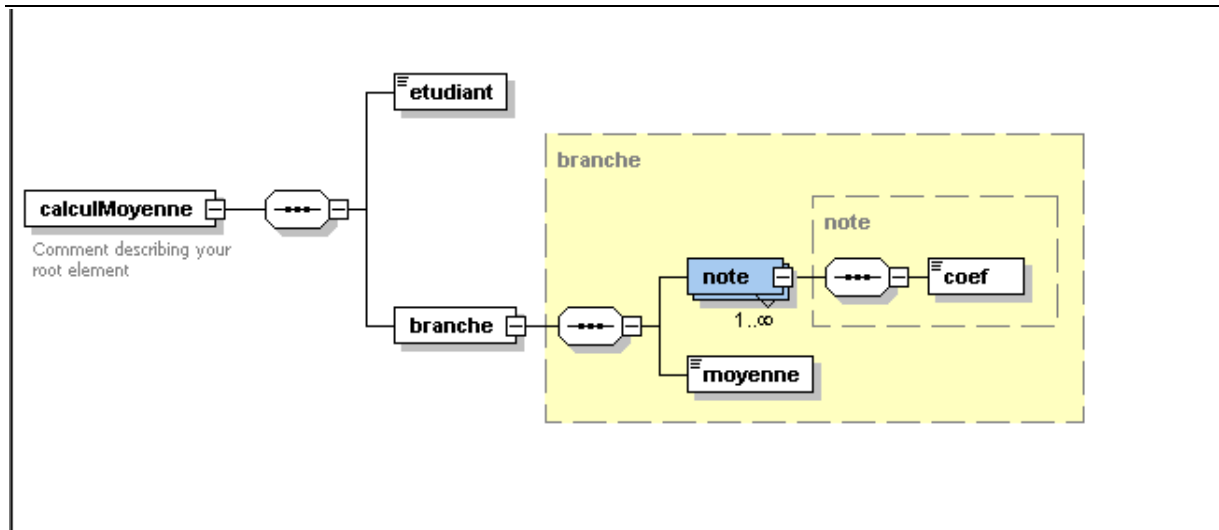
Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<!--DTD generated by XMLSpy v2005 rel. 3 U (http://www.altova.com)-->
<!--Comment describing your root element-->
<!ELEMENT calculMoyenne (etudiant, branche)>
<!ELEMENT etudiant (#PCDATA)>
<!ELEMENT branche (note+, moyenne)>
<!ATTLIST branche
    nom CDATA #REQUIRED
>
<!ELEMENT note (coef)>
<!ATTLIST note
    valeur NMTOKEN #REQUIRED
>
<!ELEMENT coef (#PCDATA)>
<!ELEMENT moyenne (#PCDATA)>
```

Schema

XML Schema est un langage de description de format de document XML permettant de définir la structure d'un document XML. Un schéma XML est lui-même un fichier XML. La connaissance de la structure d'un document XML permet notamment de vérifier la validité de ce document. Un fichier de description de structure (XML Schema Description en anglais, ou fichier XSD) est donc lui-même un document XML.

Exemple :



XPath

XPath est une syntaxe (non XML) pour désigner une portion d'un document XML. Initialement créé pour fournir une syntaxe et une sémantique aux fonctions communes à XPointer et XSL, XPath a rapidement été adopté par les développeurs comme un petit langage d'interrogation.

XSLT

XSLT (eXtended Stylesheet Language Transformations), défini au sein de la recommandation XSL du W3C, est un langage de transformation XML de type fonctionnel.

L'objectif principal est la transformation d'un document XML vers un autre, ou un dialecte XML (XHTML, XSL-FO, HTML, etc.). Cependant, le langage XSLT permet aussi les transformations vers tout autre type de document, au format texte ou dans un format binaire (bien que ceci ne soit pas nativement prévu par la recommandation XSLT).

Exemple pour générer du html :

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/xpath-functions">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
  <html>
    <head>
      <title>Bulletin</title>
    
```

```

</head>
<body bgcolor="#FFFCDD">
  <xsl:apply-templates select="calculMoyenne"/>
</body>
</html>
</xsl:template>
<xsl:template match="calculMoyenne">
  <h1>Bulletin de notes</h1>
  <h2>Etudiant <xsl:value-of select="etudiant" /></h2>
  <xsl:apply-templates select="branche"/>
</xsl:template>
<xsl:template match="branche">
  <h2>Branche <xsl:value-of select="@nom"/></h2>
  <table width="40%">
    <tbody>
      <tr align="left">
        <th>Notes</th>
        <th>Coefficient</th>
      </tr>
      <xsl:apply-templates select="note"/>
      <tr align="left">
        <th>Moyenne</th>
        <th><xsl:value-of select="moyenne"/></th>
      </tr>
    </tbody>
  </table>
</xsl:template>
<xsl:template match="note">
<tr align="left">
  <th><xsl:value-of select="@valeur"/></th>
  <th><xsl:value-of select="coef"/></th>
</tr>
</xsl:template>
</xsl:stylesheet>

```

Exemple pour générer du PDF :

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:fn="http://www.w3.org/2005/02/xpath-functions"
xmlns:xdt="http://www.w3.org/2005/02/xpath-datatypes"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml" version=".0" encoding="ISO-8859-1"
indent="yes"/>
  <xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master

```

```

        master-name="first"
        margin-left="2cm"
        margin-right="2cm"
        margin-bottom="0.5cm"
        margin-top="0.75cm"
        page-width="21cm"
        page-height="29.7cm">
        <fo:region-body></fo:region-body>
    </fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="first">
    <fo:flow flow-name="xsl-region-body" font-size="14pt" line-
height="14pt">
        <fo:block font-size="18pt">Bulletin de notes</fo:block>
        <xsl:apply-templates select="calculMoyenne"></xsl:apply-
templates>
    </fo:flow>
</fo:page-sequence>
</fo:root>
</xsl:template>

<xsl:template match="calculMoyenne">
    <fo:block font-size="12pt" text-align="left" line-height="40pt" space-
before="0.5cm" space-after="1.0cm">
        Etudiant <xsl:value-of select="etudiant"></xsl:value-of>
        Branche <xsl:apply-templates select="branche"/>
    </fo:block>
    <fo:table table-layout="auto" width="80%" border-style="solid">
    <fo:table-column column-number="1" column-width="20mm"></fo:table-
column>
    <fo:table-column column-number="2" column-width="20mm"></fo:table-
column>
        <fo:table-body text-align="left">
            <fo:table-row>
                <fo:table-cell border-style="solid" padding="4px">
                    <fo:block>Note:</fo:block>
                </fo:table-cell>
                <fo:table-cell border-style="solid" padding="4px">
                    <fo:block>Coef:</fo:block>
                </fo:table-cell>
            </fo:table-row>
            <xsl:apply-templates select="branche/note"/>
            <fo:table-row>
                <fo:table-cell border-style="solid" padding="4px">
                    <fo:block>Moyenne: </fo:block>
                </fo:table-cell>
                <fo:table-cell border-style="solid" padding="4px">
                    <fo:block><xsl:apply-templates
select="branche/moyenne"/></fo:block>
                </fo:table-cell>
            </fo:table-row>
        </fo:table-body>
    </fo:table>

```

```

</fo:table-row>
</fo:table-body>
</fo:table>
</xsl:template>

<xsl:template match="branche">
<xsl:value-of select="@nom"></xsl:value-of>
</xsl:template>

<xsl:template match="branche/note">
<fo:table-row>
  <fo:table-cell padding="4px">
    <fo:block><xsl:value-of select="@valeur"></xsl:value-
of></fo:block>
  </fo:table-cell>
  <fo:table-cell border-left="1px solid black" padding="4px">
    <fo:block><xsl:value-of select="coef"></xsl:value-of></fo:block>
  </fo:table-cell>
</fo:table-row>
</xsl:template>

<xsl:template match="moyenne">
<xsl:value-of select="."></xsl:value-of>
</xsl:template>
</xsl:stylesheet>

```

JDOM

Le principal rôle de DOM est de fournir une représentation mémoire d'un document XML sous la forme d'un arbre d'objets et d'en permettre la manipulation. JDOM est une version pour Java.

Créer un PDF :

```
public ByteArrayOutputStream createPDF() throws
    TransformerConfigurationException, TransformerException {

    Driver driver = new Driver();
    Logger logger = new ConsoleLogger(ConsoleLogger.LEVEL_INFO);
    MessageHandler.setScreenLogger(logger);
    driver.setLogger(logger);
    driver.setRenderer(Driver.RENDER_PDF);

    ByteArrayOutputStream out = new ByteArrayOutputStream();
    driver.setOutputStream(out);

    Source xslfo = new StreamSource(new File(
        "F:/EMF/3.Jahr/P/W10/exercices/testform.pdf.xsl"));
    TransformerFactory fab = TransformerFactory.newInstance();
    Transformer trans = fab.newTransformer(xslfo);
    Result res = new SAXResult(driver.getContentHandler());
    Source src = new StreamSource(new File(
        "F:/EMF/3.Jahr/P/W10/exercices/testform.données.xml"));

    trans.transform(src, res);

    return out;
}
```

3. Conclusion

C'était un module intéressant mais aussi difficile à comprendre. Il faut vraiment travailler et comprendre.

Bibliographie

<http://www.commentcamarche.net/xml/xmltd.php3>

<http://www.google.ch> (definitons)