XML, XSL

TPs du module TLI - 2014

1. Utilisation de formats XML

L'objectif de cette première partie est d'utiliser des formats XML courants pour se familiariser avec XML. Nous vous proposons d'étudier MathMl puis MusicXML.

La définition de MathMl est disponible sur http://www.w3.org/Math/.

La définition de MusicXml est disponible sur http://www.recordare.com (voir également le tutoriel http://www.recordare.com/xml/helloworld.html)

1.1. Intégrer un format XML

1.1.1. Intégration de MathMl

Regardez le fichier **maths.xhtml**. Quelles différences observez-vous avec un document XHTML classique?

Essayez d'afficher ce document dans un navigateur différent de firefox/iceweasel 3.0. Conclusion?

- intégrez quelques formules mathématiques sur une page web (fractions, racine, etc.).
- à quoi sert la balise mrow ?
- comment fonctionne la balise mfenced ?
- je ne veux pas répéter dans chaque balise math l'attribut « xmlns="http://www.w3.org/1998/Math/MathML" ». Quelle solution proposez-vous ?

1.1.2. Intégration de MusicXML

Lorsque vous utilisez MusicXML, vous devriez avoir un message indiquant qu'il n'existe aucune information de rendu permettant d'afficher le contenu. C'est ce qui se produit avec le fichier ***musique.xml***.

Pourquoi musique.xml n'est-il pas interprété par le navigateur ?

Il est possible d'utiliser un plugin spécialisé (celui proposé par la société myriad) : http://www.myriad-online.com/en/products/mmplugin.htm pour obtenir un rendu interprété (à faire si vous le souhaitez chez vous...).

1.2. Spécifier un format XML : les DTD, les schémas

L'objectif de cette section est de spécifier et réaliser maintenant vous-mêmes vos documents XML. Pour cela, il faut commencer par en donner le modèle dans une DTD (de moins en moins utilisé, cf. cours) ou beaucoup mieux, un schéma (XSD).

- ~~réalisez une définition d'un format XML **SIMPLE** (quelques types de balises et attributs, pas plus) de votre choix à l'aide d'une DTD puis d'un Schéma.~~
- créer un document XML de test
- valider votre DTD et votre schéma

Valider un document xml (syntaxe) : http://www.w3schools.com/xml/xml_validator.asp

Valider un schéma : http://www.utilities-online.info/xsdvalidation/

Valider une dtd :

2. Ressources sur XML

Tutoriel XML : http://www.gchagnon.fr/cours/xml/index.html

Schéma XML: http://www.gchagnon.fr/cours/xml/schema.html

Sur XML : les meilleures ressources sont souvent celle du w3c...

- http://www.w3schools.com/xml/default.asp (tutoriel);
- http://www.w3schools.com/dtd/ pour les dtd;
- http://www.w3schools.com/Schema/ pour les schémas xml;
- http://writer2latex.sourceforge.net/ wrriter2xhtml vous permet d'éxporter facilement vos documents odt en xhtml + mathml.

1) écrire vous-même une feuille de rendu (en XSL);

- ~~Essayez de modifier ***musique.xsl*** pour présenter les informations autrement.~~

3. XSL v1

L'objectif de cette partie est de découvrir XSL version 1.

Voir les questions-type en fin de document.

Les corrections sont données.

3.1. Prise en main de XSL par l'exemple

Ouvrez les documents ***musique.xsl*** et ***musiqueStyle.xml*** ;

Expliquez ce qui se passe quand vous ouvrez musiqueStyle.xml dans un navigateur

Récupérez le fichier ***document.xml*** : c'est un document utilisant la dtd ***document.dtd***

Utilisez des fonctions standards de XSL dans votre feuille XSL afin de proposer une analyse de ce document (voir questions-types plus bas).

3.2. Utilisation de fonctionnalités (un peu plus) avancées en XSL1

3.2.1. Template match et template name

En xsl 1, il y a deux technique pour regrouper des traitements sur des noeuds:

1) ***template match="expression xpath"***: ce template sera appelé lorsque l'expression xpath en paramètre sera vraie. Il est souvent nécessaire d'indiquer à xsl qu'on souhaite appliquer des template-match avec l'instruction ***xsl:apply-templates***

2) ***template name="nom du template"***: un peu comme une fonction ou une procédure, ce template doit être appellé avec:

<xsl:call-template name="nom">

3.2.2. Fonctions standards de XSL

Les fonctions peuvent être utilisée dans des expressions xpath, des valeurs pour les attributs "select". Elles concernent les noeuds, les chaines de caractères, les nombres, etc.

Quelques fonctions standards: **concat**, **translate**, **position()**, **number()**, etc.

3.2.3. Expressions Xpath et nodales

Utilisez des expressions XPATH afin de traiter certains noeuds en particulier.

3.2.4. Variables et paramètres

Il est possible en xsl de définir des variables (**<xsl:variable>**) et des paramètres (**<xsl:param>**).

Variables et paramètres ne sont pas modifiables une fois utilisées dans un template.

Les paramètres peuvent être utilisés soit à la racine de la feuille xsl, soit comme paramètres de **<template name= "xxx ">**. Lors de l'appel du template **xxx**, on peut spécifier les paramètres à l'aide de **<xsl:with-param>** dans le corps de **<xsl:call-template>**.

Il est possible également de passer des valeurs aux paramètres globaux lors de l'application de la feuille de style.

Intégrez ces fonctionnalités.

3.2.5. Modes

Proche de la surcharge en objet, vous pouvez préciser un mode pour l'exécution d'un **template match**. Essayez de mettre en application cette fonctionnalité (voir http://www.dpawson.co.uk/xsl/sect2/modes.html).

3.3. Mise en pratique

Le résultat attendu et une solution sont donnés (document.html et document.xsl)

Répondez aux questions suivantes :

1) Combien de phrases, de mots et d'expressions mathématiques y-a-t-il dans le document?

   *Utiliser **count(xpath)***

2) Afficher une table des matières du document

   *Se servir des expressions xpath, des templates **match** et éventuellement des fonctions **following** et **following-sibling***

3) Combien y-a-t-il d'expressions mathématiques "complexes" (qui utilisent au moins une fraction ou une racine carrée).

4) Y a-t-il un titre de niveau 1 non suivi de titre de niveau 2?
5) Y a-t-il deux titres avec le même nom? Si c'est le cas, renommer le deuxième titre en ajoutant (#2).
6) Mettre les titres en majuscule
7) Faites une fonction (template name) qui affiche un message si les lettres grecques Delta ou alpha sont présentes dans une expression. Elle prendra en paramètre un noeud de type "math"
8) Faites un rendu html de tout ça! Pour les maths, utiliser **xsl:copy-of**

4. XSL v2

4.1. Saxon : le parser xsl

Saxon est un analyseur/parseur xsl permettant d'exécuter une transformation xsl (http://www.saxonica.com). Il existe maintenant en version libre (home édition) et non-libre avec des extensions (Professional et enterprise editions).

Il peut être utilisé en ligne de commande, ou intégré directement dans des développements en java ou dotnet (les deux api existent).

4.2. Utilisation

Nous allons utiliser l'api java de saxon, en ligne de commande pour ce tp.

Récupérer les fichiers **saxon.jar** et **saxon-dom.jar**.

Exécuter votre feuille de style en ligne de commande :

```
java -jar saxon.jar -o sortie.html -xsl:document.xsl document.xml
```

4.3. Intérêts de saxon

1) saxon est beaucoup plus clair dans ses messages d'erreur
2) il est optimisé en temps de traitement
3) il gère xsl 2
4) il est intégrable dans des application

4.4. Questions xsl 2

1. transformez votre feuille xsl en xsl 2
2. réalisez une fonction ***estComplexe*** qui prend en paramètre un noeud math et qui renvoie un booléen (vrai si l'expression contient des racines ou des fractions).
3. utilisez votre fonction, par exemple:

```
<xsl:template match select="m:math">
<xsl:choose>
      <xsl:when test="mesFonctions:estComplexe(.)">
            <xsl:text>Trop compliqué</xsl:text>
      </xsl:when>
```

```
            <xsl:otherwise>
                  <xsl:copy-of select="."/>
            </xsl:otherwise>
      </xsl:choose>
</xsl:template>
```

4) en vous inspirant des fonctions écrites par Priscilla Walmsley (http://www.functx.com), réalisez des fonctions utilisant les expressions régulières, les manipulations de chaines et les manipulations d'arbres. Utilisez les syntaxes xslt et non xquery (vous avez les exemples de codes).

5) **intégrez et testez vos fonctions**

5. Documentation

http://www.zvon.org/xxl/XSLTutorial/Books/Book1/index.html : Le meilleur tutorial pour débuter:

http://www.dpawson.co.uk/ : le meilleur site d'information sur xsl

http://www.mulberrytech.com/xsl/xsl-list/ : LA liste sur xsl

6. Les API parser de JAVA

~~Utilisez les API SAX et DOM de Java pour :~~ 1) réaliser les transformations xml de vos documents. 2) créer un document XML au format que vous avez définit en début de TP

6.1. Utilisation de l'API DOM (exemple)

6.1.1. Création d'un fichier XML

```
try
      {
            // Création d'un nouveau DOM
        DocumentBuilderFactory fabrique = DocumentBuilderFactory.newInstance();
        DocumentBuilder constructeur = fabrique.newDocumentBuilder();
        Document document = constructeur.newDocument();

        // Propriétés du DOM
        document.setXmlVersion("1.0");
        document.setXmlStandalone(true);

        //Création de l'arborescence du DOM
        /racine
        Element racine = document.createElement("nomRacine");
        racine.setAttribute("version", "v1.4");

        Element fils = document.createElement("nomFils");
       // ajout du fils à la racine
        racine.appendChild(obs);
      //ajout de la racine au document
        document.appendChild(racine);
```

```
    /* Sauvegarde du fichier xml */
    Source source = new DOMSource(document);

    // Création du fichier de sortie
    File f = new File("fichier.xml");
    Result resultat = new StreamResult(f);

    // Configuration du transformer
    TransformerFactory tfabrique = TransformerFactory.newInstance();
    Transformer transformer = tfabrique.newTransformer();
    transformer.setOutputProperty(OutputKeys.INDENT, "yes");
    transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
    // Transformation
     transformer.transform(source, resultat);
}
catch (ParserConfigurationException pce) {pce.printStackTrace();}
catch (TransformerConfigurationException tce) {tce.printStackTrace();}
catch (TransformerException te) {te.printStackTrace();}
```

6.1.2. Lecture d'un fichier XML

```
// lecture du contenu d'un fichier XML avec DOM
File xml = new File("fichier.xml");
Document document =  constructeur.parse(xml) ;
```

6.2. Et avec SAX ?

Quels seraient les avantages et les inconvénients d'utiliser sax dans ce problème ?

Proposez une modification théorique de l'algorithme avec l'API sax.

6.3. Utilisation des parseurs XSL (SAX/DOM)

6.3.1. Utiliser saxon et pas xerxes

- Récupérer les .jar contenant saxon;
- Les intégrer au path d'exécution du projet;
- ajouter la ligne suivante:

```
System.setProperty("javax.xml.transform.TransformerFactory",
                   "net.sf.saxon.TransformerFactoryImpl");
```

6.3.2. Transformer un document par une feuille de style

cf.exemple : https://svn.liris.cnrs.fr/nat/trunk/nat/transcodeur/Transcodeur.java

7. XSL-FO

FO est un format xml de mise en page (un peu comme les css pour le HTML).

On parle de « XSL-FO » car très souvent, un document fo est généré à partir d'un fichier XML grace à XSL, mais **ce n'est pas une obligation !!!**

Le fichier *FO* doit être interprété par un parseur FO, le plus connu est le parseur FOP. C'est un programme java permettant d'interpréter le fichier fo (un peu comme mathML avec Firefox).

7.1. Exemple de fichier fo :

```
<fo:root>
      <fo:layout-master-set>
            <fo:simple-page-master master-name="A4">
                  <fo:region-body margin="2cm"/>
            </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence master-reference="A4">
            <fo:flow flow-name="xsl-region-body">
                  <fo:block          color="green"          font-size="12pt"
font-family="sans-serif">Salut le monde</fo:block>
                  <fo:block margin="1cm" font-size="16pt" font-style="bold"
font-family="regular">un autre</fo:block>
            </fo:flow>
      </fo:page-sequence>
</fo:root>
```

Ce fichier définit un document au format A4, avec une marge de 2cm.

Le contenu du document est composé de deux blocs, le premier en vert et en 12pt, le deuxième avec une marge de 1cm, une police 16pt gras, couleur par défaut.

7.2. Comment en faire un pdf ?

Plusieurs outils existent : le plus simple et le plus connu est FOP.

Pour l'installer sous ubuntu :

```
      apt-get install fop
```

Utilisation :

```
      fop document.fo test.pdf
```

7.3. Tutoriel correct sur XSL-FO

Le meilleur que j'ai trouvé est celui du w3c : http://www.w3schools.com/xslfo/default.asp