

Langages de l'internet

Intégrer XML et XSL dans des applications non web

Bruno Mascret

¹**Laboratoire d'InfoRmatique en Image et Systèmes d'information**

LIRIS UMR 5205 CNRS/INSA de Lyon/Université Claude Bernard Lyon 1/Université
Lumière Lyon 2/Ecole Centrale de Lyon

Equipe Silex

20, Avenue Albert Einstein - 69622 Villeurbanne cedex

<http://liris.cnrs.fr>

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Introduction

Objectif : utiliser les technologies web dans des applications non web (ou non 100% web)

Quels intérêts, quelles applications ?

- ≡ partage de formats d'échange
- ≡ réutilisation de formats existants
- ≡ portabilité des solutions
- ≡ langages dédiés plus performants (xsl)
- ≡ ...

Des API existent pour pratiquement chaque langage (php, java, c++, etc.)

Nous verrons dans ce cours des exemples en java

Introduction

Objectif : utiliser les technologies web dans des applications non web (ou non 100% web)

Quels intérêts, quelles applications ?

- ▢ partage de formats d'échange
- ▢ réutilisation de formats existants
- ▢ portabilité des solutions
- ▢ langages dédiés plus performants (xsl)
- ▢ ...

Des API existent pour pratiquement chaque langage (php, java, c++, etc.)

Nous verrons dans ce cours des exemples en java

Plan

- 1 Introduction
- 2 **Présentation des architectures**
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Présentation des architectures XML

Vous connaissez !

Présentation des architectures XSL

Vous connaissez aussi !

Attention, tous les parsers xsl n'implémentent pas xsl v2.

Le parser xsl de référence est SAXON

(<http://www.saxonica.com>)

Il existe des API de saxon pour java et dotnet.

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Les API XML/XSL sous java

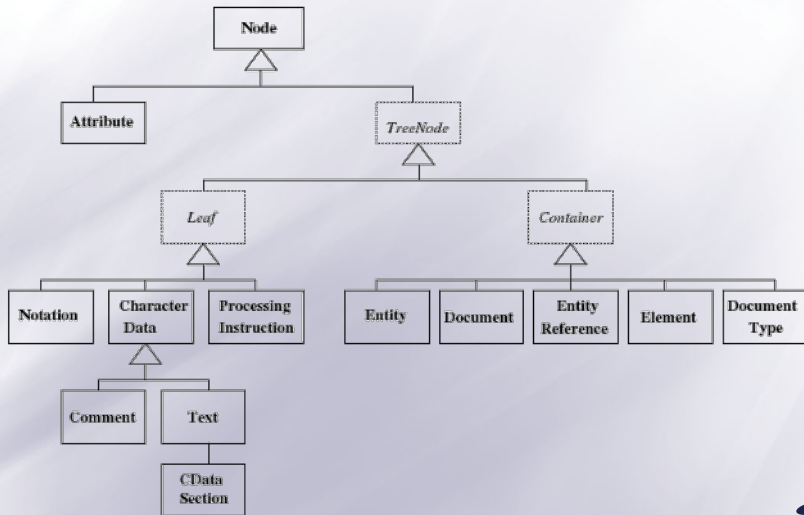
DOM

DOM (Document Object Model) est une API basée sur une **représentation hiérarchique** des documents XML.

- représentation sous forme d'arbre en mémoire
- pertinent si on souhaite modifier le document
- permet de traiter le document globalement (insertion en tant que fragments par exemple)
- c'est une recommandation du W3C
- orienté objet, mais pas forcément toujours dans le bon sens !

Les API XML/XSL sous java

DOM : les objets



Les API XML/XSL sous java

SAX

SaX (Simple API for XML) est une API basée sur des **déclencheurs** (événements et actions) du document XML traité.

- ≡ fonctionnement sous forme de flux (un peu comme pour xslt)
- ≡ pertinent si on souhaite extraire des informations du document ou travailler en mode synchrone
- ≡ permet de traiter le document globalement (insertion en tant que fragments par exemple)

Attention ! SAX et DOM ne sont pas compatibles !

Les API XML/XSL sous java

Xerces et autres

- ▢ chaque document doit d'abord être analysé (parser)
- ▢ le "parsage" via DOM produira un arbre
- ▢ le parsage via SAX produira un flux de données

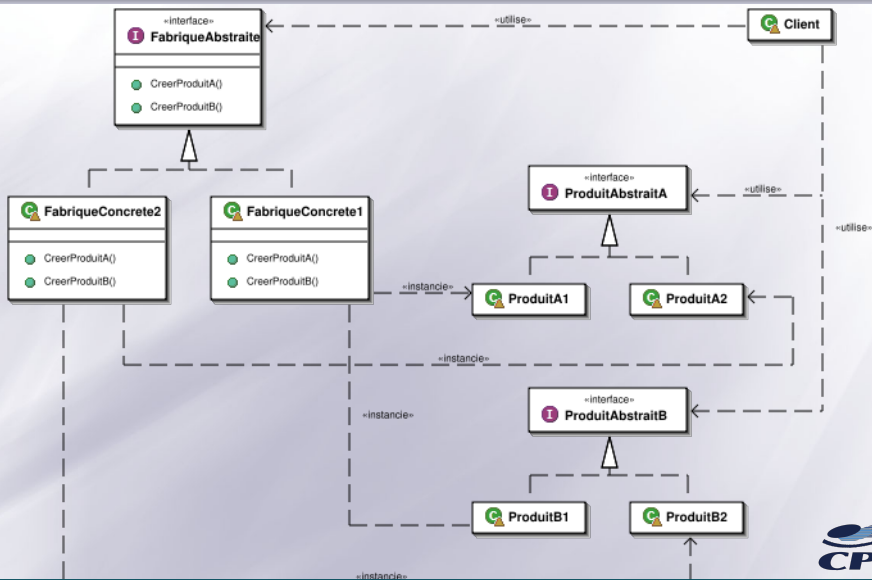
Il existe de nombreuses implémentations de parsers XML.
De même, il existe plusieurs processeurs XSLT : xalan (par défaut dans l'api sun), saxon, etc.

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Les fabriques

Rappel du design pattern Fabrique



Les fabriques

Les fabriques XML/XSL pour java

Pourquoi passer par des fabriques dans les API java ?

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


Les fabriques

Les fabriques XML/XSL pour java

Pourquoi passer par des fabriques dans les API java ?

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

Les fabriques

Exemple : création et lecture d'un doc 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(fils);  
    //ajout de la racine au document  
    document.appendChild(racine);
```

Les fabriques

Exemple : création et lecture d'un doc XML

```
/* 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);

}

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

Les fabriques

Exemple : transformation en XSL

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
//configuration de la fabrique
factory.setNamespaceAware(true);
factory.setValidating(true);
factory.setIgnoringElementContentWhitespace(true);
factory.setIgnoringComments(true);
factory.setIgnoringElementContentWhitespace(false);
try
{
    //sinon, génère parfois des null pointer exp au parsing (problème
    // avec les simples quote)
    factory.setFeature(
        "http://apache.org/xml/features/dom/defer-node-expansion", false);
    DocumentBuilder builder = factory.newDocumentBuilder();
    Document doc = builder.parse(new File(entree));
    doc.setStrictErrorChecking(true);
    TransformerFactory transformFactory = new TransformerFactoryImpl();
    StreamSource styleSource = new StreamSource(new File(ADRESSE_XSL));
    ...
}
```

Les fabriques

Exemple : transformation en XSL

```
...  
// lire le style  
Transformer transform = transformFactory.newTransformer(styleSource);  
// Si besoin : transform.setOutputProperty(name, value)  
DOMSource in = new DOMSource(doc);  
// Création du fichier de sortie  
File file = new File(cible);  
Result resultat = new StreamResult(fichier);  
// Si besoin : transform.setParameter("nomDuParam", valeur);  
transform.transform(in, out);  
}
```

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API**
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Principales fonctionnalités des API

Validation

- messages d'erreurs plus clairs
- le document xml doit être valide sous tout rapport !
- il existe des outils de "nettoyage" comme tagsoup par exemple (<http://home.ccil.org/~cowan/tagsoup/>) pour sax
- les processus de parsage et d'analyse sont finement configurables (validation, paramètres)
- on peut les utiliser dynamiquement (variable pour des paramètres xsl par exemple)
- on peut décider comment gérer les erreurs (génération et traitement des exeptions)

Principales fonctionnalités des API

Exceptions

```
try
{
    /*
     * instructions
     * sensibles
     */
}
catch (ParserConfigurationException pce) {pce.printStackTrace();}
catch (TransformerConfigurationException tce) {tce.printStackTrace();}
catch (TransformerException te) {te.printStackTrace();}
```


Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

XSL-FO

C'est quoi ?

- FO est un format xml de mise en page ;
- On parle de "XSL-FO" car très souvent, un document fo est généré à partir d'un fichier XML avec XSL ;
- Ce n'est pas obligatoire (on peut écrire directement du FO) !
- le fichier FO doit être interprété par un outil pour produire une sortie "lisible" (PDF par exemple)

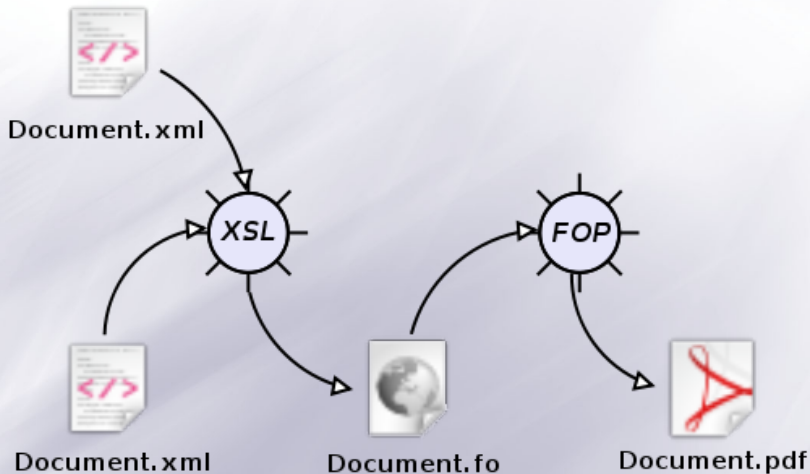
XSL-FO

Comment ça marche ?

- Le fichier FO est produit soit directement, soit via un document xml transformé ;
- 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).
- le document produit peut être un PDF, une page HTML, une image, etc.

XSL-FO

Shéma général



XSL-FO

Exemple de document FO

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

```
<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>
```

XSL-FO

Exécution avec FOP

fop document-fo.xml document-pdf.pdf

Il peut y avoir des erreurs/warnings suivant la configuration de votre système (propriétés de police, ou police non prises en charge notamment

Invalid property value encountered in font-style="bold" :

org.apache.fop.fo.expr.PropertyException :

file :/home/bruno/travail/cpe/coursFinal/document-fo.xml :13 :42 : No conversion defined bold ; property : 'font-style' (See position 17 :30)

Font "regular,normal,400" not found. Substituting with "any,normal,400".

XSL-FO

Exécution avec FOP

fop document-fo.xml document-pdf.pdf

Il peut y avoir des erreurs/warnings suivant la configuration de votre système (propriétés de police, ou police non prises en charge notamment

Invalid property value encountered in font-style="bold" :

org.apache.fop.fo.expr.PropertyException :

file :/home/bruno/travail/cpe/coursFinal/document-fo.xml :13 :42 : No conversion defined bold ; property :'font-style' (See position 17 :30)

Font "regular,normal,400" not found. Substituting with "any,normal,400".

XSL-FO

Résultat en PDF

Salut le monde

un autre

Plan

- 1 Introduction
- 2 Présentation des architectures
- 3 Les API XML/XSL sous java
 - DOM
 - SAX
 - Xerces, Xalan et autres
- 4 Les fabriques
 - Rappel du design pattern Fabrique
 - Les fabriques XML/XSL pour java
- 5 Principales fonctionnalités des API
- 6 XSL-FO
- 7 Quelques cas d'utilisation

Quelques cas d'utilisation