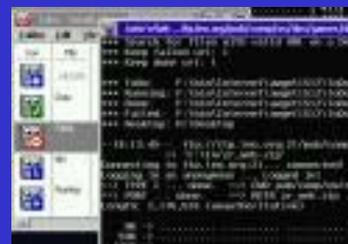
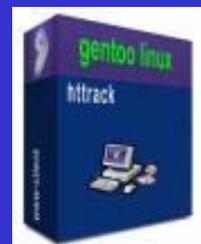
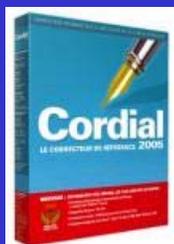


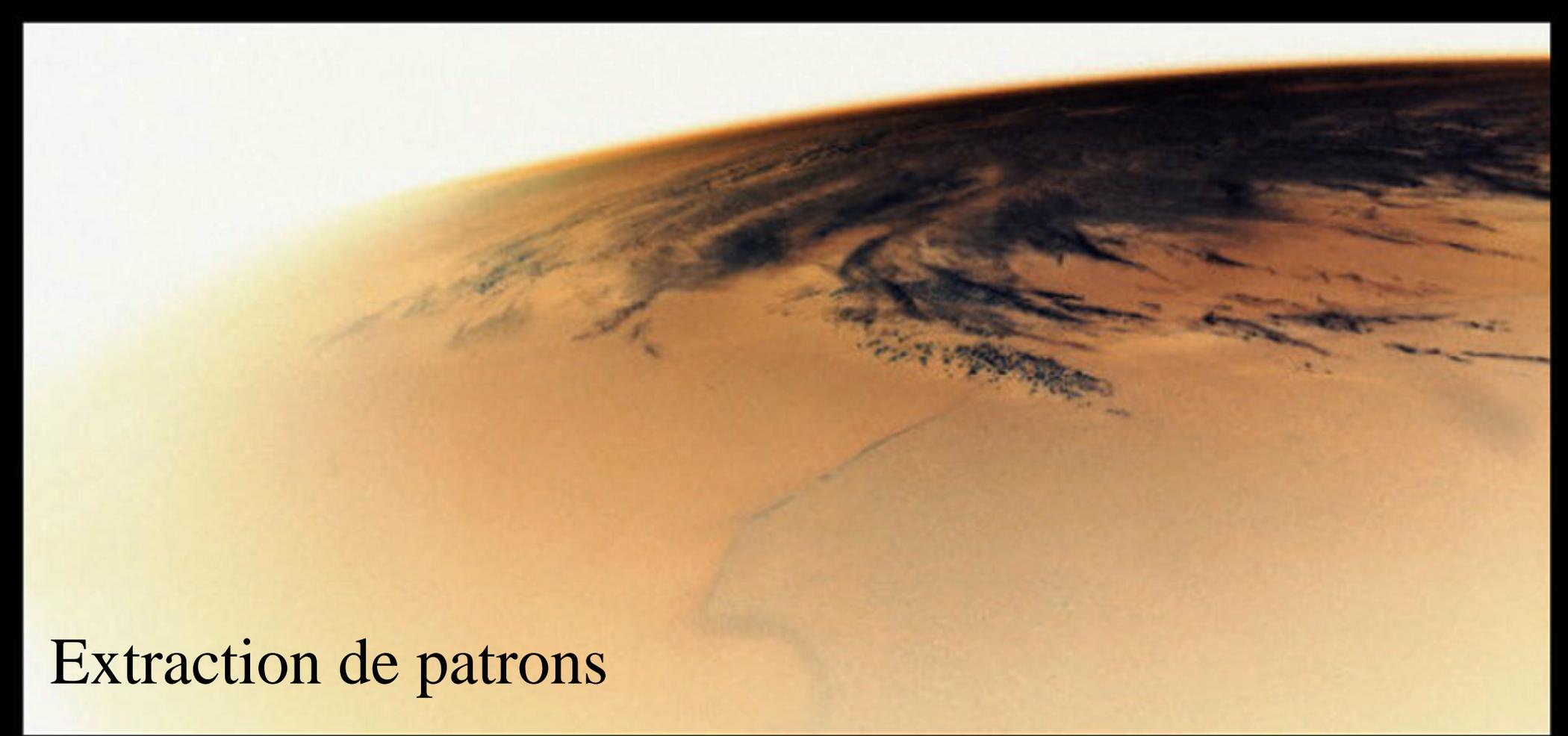


# *Programmation et projet encadré*

## *Boîte à outils*

### *Série 3 : extraction de patrons*





Extraction de patrons

*Boîte à outils : série 3*

# Objectif

- Extraire des patrons dans les sorties produites à l'issue de la *BàO série 2*
- 2 méthodes
  - Un programme qui prend en entrée les sorties «brutes» de l'étiquetage
    - En particulier la sortie obtenue avec Cordial
  - Requête XPATH sur le fichier XML construit
    - En particulier avec *treetagger* et le programme `treetagger2xml`

# Travail personnel [*série 3*]:

extraction de patrons dans les contenus des fils

- Objectif :
  - Vous devez construire une liste de patrons à partir des contenus textuels des fils traités
    - Liste des patrons à extraire :
      - NOM ADJ
      - NOM NOM
      - NOM PREP NOM
  - En sortie, une liste de patrons au format TXT
  - Cette liste sera le point d'entrée de la *BàO série 4*

# Phase 1

- Extraction des patrons *via* XPath
- En entrée, la sortie XML produite dans la *BàO série 2*
  - Celle construite avec *treetagger* et reformatée par le programme fourni `treetagger2xml`
- Écriture d'une requête pour extraire les éléments correspondant au patron...

```

<?xml version="1.0" encoding="iso-8859-1" ?>
<?xml-stylesheet type="text/xsl" href="SORTIE-etiquetage-xsl.xsl"?>
<PARCOURS>
<NOM>Votre nom</NOM>
<ETIQUETAGE><file>
<name>arbo-fils/2005/Nov/4/01-00-00/0,2-3208,1-0,0.xml</name>
<element><data type="type">VER:pres</data><data type="lemma">retrouver</data><data type="string">Retrouvez</data></element>
<element><data type="type">NUM</data><data type="lemma">L</data><data type="string">1</data></element>
<element><data type="type">PUN</data><data type="lemma">'</data><data type="string">'</data></element>
<element><data type="type">ADV</data><data type="lemma">ensemble</data><data type="string">ensemble</data></element>
<element><data type="type">PRP:det</data><data type="lemma">du</data><data type="string">des</data></element>
<element><data type="type">NOM</data><data type="lemma">dépêche</data><data type="string">dépêches</data></element>
<element><data type="type">PRP</data><data type="lemma">sur</data><data type="string">sur</data></element>
<element><data type="type">NOM</data><data type="lemma">http</data><data type="string">http</data></element>
<element><data type="type">PUN</data><data type="lemma">:</data><data type="string">:</data></element>
<element><data type="type">PUN</data><data type="lemma">/</data><data type="string">/</data></element>
<element><data type="type">PUN</data><data type="lemma">.</data><data type="string">.</data></element>
<element><data type="type">NOM</data><data type="lemma">www</data><data type="string">www</data></element>
<element><data type="type">SENT</data><data type="lemma">.</data><data type="string">.</data></element>
<element><data type="type">NOM</data><data type="lemma">lemonde</data><data type="string">lemonde</data></element>
<element><data type="type">SENT</data><data type="lemma">.</data><data type="string">.</data></element>
<element><data type="type">NOM</data><data type="lemma">fr</data><data type="string">fr</data></element>
</file>
<file>
<name>arbo-fils/2005/Nov/4/01-00-00/0,2-3210,1-0,0.xml</name>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">Le</data></element>
<element><data type="type">NOM</data><data type="lemma">désastre</data><data type="string">désastre</data></element>
<element><data type="type">ADJ</data><data type="lemma">écologique</data><data type="string">écologique</data></element>
<element><data type="type">PRP</data><data type="lemma">de</data><data type="string">de</data></element>
<element><data type="type">DET:ART</data><data type="lemma">la</data><data type="string">la</data></element>
<element><data type="type">NOM</data><data type="lemma">rivière</data><data type="string">rivière</data></element>
<element><data type="type">ADJ</data><data type="lemma">Songhua</data><data type="string">Songhua</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">PRP</data><data type="lemma">dans</data><data type="string">dans</data></element>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
<element><data type="type">NOM</data><data type="lemma">Heilongjiang</data><data type="string">Heilongjiang</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">KON</data><data type="lemma">et</data><data type="string">et</data></element>
<element><data type="type">DET:ART</data><data type="lemma">les</data><data type="string">les</data></element>
<element><data type="type">NOM</data><data type="lemma">tentative</data><data type="string">tentatives</data></element>
<element><data type="type">VER:pres</data><data type="lemma">faire</data><data type="string">faites</data></element>
<element><data type="type">PRP:det</data><data type="lemma">au</data><data type="string">au</data></element>
<element><data type="type">NOM</data><data type="lemma">niveau</data><data type="string">niveau</data></element>
<element><data type="type">ADJ</data><data type="lemma">local</data><data type="string">local</data></element>
<element><data type="type">PRP</data><data type="lemma">pour</data><data type="string">pour</data></element>
<element><data type="type">VER:infi</data><data type="lemma">étouffer</data><data type="string">étouffer</data></element>
<element><data type="type">NUM</data><data type="lemma">L</data><data type="string">1</data></element>
<element><data type="type">PUN</data><data type="lemma">'</data><data type="string">'</data></element>
<element><data type="type">NOM</data><data type="lemma">affaire</data><data type="string">affaire</data></element>

```

Exemple de sortie issue de *BàO* série 2

# La requête

```
( //element[ ./data[contains(text(
, 'NOM' )]]/following-
sibling::element[1][ ./data[conta
ins(text( , 'ADJ' )]]/data[3] ) |
( //element[ ./data[contains(text(
), 'ADJ' )]]/preceding-
sibling::element[1][ ./data[conta
ins(text( , 'NOM' )]]/data[3] ) )
```

**En rouge** : on recherche parmi les éléments contenant la chaîne NOM, ceux qui sont suivis juste après par un élément contenant la chaîne ADJ, et pour ceux là on affiche le 3<sup>ème</sup> nœud fils data (le mot)

**En bleu** : on recherche parmi les éléments contenant la chaîne ADJ, ceux qui sont précédés juste avant par un élément contenant la chaîne NOM, et pour ceux là on affiche le 3<sup>ème</sup> nœud fils data (le mot)

L'opérateur | permet de rechercher les 2 types de nœuds visés...



## Affichage partiel du résultat avec Cooktop Avec uniquement les contenus des éléments « mot »

```

nodes: (//element[./data[contains(text(), 'NOM')]]/following-sibling::element[1][./data[contains(text(), 'ADJ')]]/data[3]) |
<!--nodes xpath (//element[./data[contains(text(), 'NOM')]]/following-sibling::element[1][./data[contains(text(), 'ADJ')]]/da
<data type="string">désastre</data>
<data type="string">écologique</data>
<data type="string">rivière</data>
<data type="string">songhua</data>
<data type="string">niveau</data>
<data type="string">local</data>
<data type="string">Xie</data>
<data type="string">Zhenhua</data>
<data type="string">Agence</data>
<data type="string">d</data>
<data type="string">ministre</data>
<data type="string">britannique</data>
<data type="string">immigration</data>
<data type="string">n</data>
<data type="string">Comité</data>
<data type="string">interministériel</data>
<data type="string">ressources</data>
<data type="string">minières</data>
<data type="string">place</data>
<data type="string">financière</data>
<data type="string">places</data>
<data type="string">boursières</data>
<data type="string">chasse</data>
<data type="string">s</data>
<data type="string">footballeur</data>
<data type="string">britannique</data>
<data type="string">ressources</data>
<data type="string">minières</data>
<data type="string">champion</data>
<data type="string">d</data>
<data type="string">distance</data>
<data type="string">respectable</data>

```

# Une feuille de style XSL pour afficher le résultat

1. Après avoir testé la requête XPATH...
  - Via XMLCooktop, XML Copy Editor...
2. On pourra construire une feuille de styles XSLT ou utiliser la feuille de style suivante pour donner à voir le résultat dans un navigateur :
  - les patrons NOM ADJ par exemple
3. On construira une feuille de styles pour construire les listes de patrons recherchées au format TXT

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="2.0">
<xsl:output method="html"/>

<xsl:template match="/">
<html>
<body bgcolor="#81808E">
<table align="center" width="90%" bgcolor="white" bordercolor="#3300FF"
  border="1">
<tr bgcolor="black">
<td colspan="3" width="90%" valign="top"><font color="white"><h1>Extraction
  de patron <font color="red"><b>NOM</b></font> <font
  color="blue"><b>ADJ</b></font></h1></font></td>
</tr>
<tr><td> <xsl:apply-templates select="./PARCOURS/ETIQUETAGE/file/element"/>
  </td></tr>
</table>
</body>
</html>
</xsl:template>

<xsl:template match="element">
<xsl:choose>
<xsl:when test="(./data[contains(text(),'NOM')]) and (following-
  sibling::element[1][./data[contains(text(),'ADJ')]])">
<font color="red"><xsl:value-of select="./data[3]"/></font><xsl:text>
  </xsl:text>
</xsl:when>
<xsl:when test="(./data[contains(text(),'ADJ')]) and (preceding-
  sibling::element[1][./data[contains(text(),'NOM')]])">
<font color="blue"><xsl:value-of select="./data[3]"/></font><br/>
</xsl:when>
</xsl:choose>
</xsl:template>
</xsl:stylesheet>

```

Règle n°1 : préparation du  
tableau HTML final  
Et lancement de la règle  
concernant les éléments visés

Règle n°2 : on filtre  
les éléments visés et  
on applique la  
requête déjà vue...

## Extraction de patron **NOM** **ADJ**

désastre écologique  
 rivière Songhua  
 niveau local  
 Xie Zhenhua  
 Agence d  
 ministre britannique  
 immigration n  
 Comité interministériel  
 ressources minières  
 place financière  
 places boursières  
 chasse s  
 footballeur britannique  
 ressources minières  
 champion d  
 distance respectable  
 président vénézuélien  
 élections législatives  
 ministre britannique  
 finances européens  
 angoisses identitaires  
 crise sociale  
 modèle politique  
 ressources minières  
 chasse s  
 temps glacial  
 footballeur britannique  
 Agence mondiale  
 symposium international  
 dopage génétique  
 Tracy Chapman  
 observatrice lucide  
 champion d  
 distance respectable

Affichage du fichier issu  
 de la *BàO série 2* (dans un  
 navigateur) sur lequel  
 on applique la feuille de  
 styles repérant les patrons visés





```
<?xml:version="1.0" encoding="iso-8859-1" ?>
<?xml:stylesheet type="text/xsl" href="SORTIE-extraction-patron-NOM-ADJ.xsl" ?>
<PARCOURS>
<NOM>Votre nom</NOM>
<ETIQUETAGE><file>
<name>arbo-fils/2005/Nov4/01-00-00/0,2-3208,1-0,0.xml</name>
<element><data type="type">VER:pres</data><data type="lemma">retrouver</data><data type="string">Retrouvez</data></element>
<element><data type="type">NUM</data><data type="lemma">L</data><data type="string">L</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">ADV</data><data type="lemma">ensemble</data><data type="string">ensemble</data></element>
<element><data type="type">PRP:det</data><data type="lemma">du</data><data type="string">des</data></element>
<element><data type="type">NOM</data><data type="lemma">dépêche</data><data type="string">dépêches</data></element>
<element><data type="type">PRP</data><data type="lemma">sur</data><data type="string">sur</data></element>
<element><data type="type">NOM</data><data type="lemma">http</data><data type="string">http</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">NOM</data><data type="lemma">www</data><data type="string">www</data></element>
<element><data type="type">SENT</data><data type="lemma">.</data><data type="string">.</data></element>
<element><data type="type">NOM</data><data type="lemma">lemonde</data><data type="string">lemonde</data></element>
<element><data type="type">SENT</data><data type="lemma">.</data><data type="string">.</data></element>
<element><data type="type">NOM</data><data type="lemma">fr</data><data type="string">fr</data></element>
</file>
<file>
<name>arbo-fils/2005/Nov4/01-00-00/0,2-3210,1-0,0.xml</name>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
<element><data type="type">NOM</data><data type="lemma">désastre</data><data type="string">désastres</data></element>
<element><data type="type">ADJ</data><data type="lemma">écologique</data><data type="string">écologiques</data></element>
<element><data type="type">PRP</data><data type="lemma">de</data><data type="string">de</data></element>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
<element><data type="type">NOM</data><data type="lemma">rivière</data><data type="string">rivière</data></element>
<element><data type="type">ADJ</data><data type="lemma">Songhua</data><data type="string">Songhua</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">PRP</data><data type="lemma">dans</data><data type="string">dans</data></element>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
<element><data type="type">NOM</data><data type="lemma">Heilongjiang</data><data type="string">Heilongjiang</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">KON</data><data type="lemma">et</data><data type="string">et</data></element>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">les</data></element>
<element><data type="type">NOM</data><data type="lemma">tentative</data><data type="string">tentatives</data></element>
<element><data type="type">VER:pres</data><data type="lemma">faire</data><data type="string">faites</data></element>
<element><data type="type">PRP:det</data><data type="lemma">au</data><data type="string">au</data></element>
<element><data type="type">NOM</data><data type="lemma">niveau</data><data type="string">niveau</data></element>
<element><data type="type">ADJ</data><data type="lemma">local</data><data type="string">local</data></element>
<element><data type="type">PRP</data><data type="lemma">pour</data><data type="string">pour</data></element>
<element><data type="type">VER:infi</data><data type="lemma">étouffer</data><data type="string">étouffer</data></element>
<element><data type="type">NUM</data><data type="lemma">L</data><data type="string">L</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">NOM</data><data type="lemma">affaire</data><data type="string">affaires</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">VER:pres</data><data type="lemma">avoir</data><data type="string">ont</data></element>
<element><data type="type">PRP:det</data><data type="lemma">au</data><data type="string">au</data></element>
<element><data type="type">NOM</data><data type="lemma">limogeage</data><data type="string">limogeage</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">NOM</data><data type="lemma">vendredi</data><data type="string">vendredi</data></element>
<element><data type="type">NUM</data><data type="lemma">@card</data><data type="string">2</data></element>
<element><data type="type">NOM</data><data type="lemma">décembre</data><data type="string">décembre</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">PRP</data><data type="lemma">de</data><data type="string">de</data></element>
<element><data type="type">NOM</data><data type="lemma">Xie</data><data type="string">Xie</data></element>
<element><data type="type">ADJ</data><data type="lemma">Zhenhua</data><data type="string">Zhenhua</data></element>
<element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
<element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
```

# Avec XML Copy Editor

Menu XML

Evaluate XPath

**Evaluate XPath** ✖

Enter XPath:

//element

OK Cancel

# Transformer un document XML (1) via une feuille de styles XSLT

## • Saxon

- *Hypertoile* :  
<http://saxon.sourceforge.net/>
- (à récupérer ici :  
[programme SAXON pour windows](#))

```

Invite de commandes
SAXON 6.4.3 from Michael Kay
Usage: java con.icl.saxon.StyleSheet [options] source-doc style-doc {param-value}...
Options:
-a          Use xsl-stylesheet FI, not style-doc argument
-da        Use standard tree data structure
-dt        Use tinstree data structure (default)
-o filename Send output to named file or directory
-w classname Use specified Emitter class for xsl:message output
-r classname Use specified URIResolver class
-t         Display version and timing information
-T         Set standard TraceListener
-TL classname Set a specific TraceListener
-u         Names are URIs not filenames
-u0        Recover silently from recoverable errors
-u1        Report recoverable errors and continue (default)
-u2        Treat recoverable errors as fatal
-x classname Use specified SAX parser for source file
-y classname Use specified SAX parser for stylesheet
-?         Display this message

C:\Documents and Settings\Serge Fleury\Bureau\ba0-29822886\GraphML2FaJek\Saxon>
saxon.bat test ../Hawaii.xml ../GraphML2FaJek.xsl

C:\Documents and Settings\Serge Fleury\Bureau\ba0-29822886\GraphML2FaJek\Saxon>
java -classpath lib\saxon.jar con.icl.saxon.StyleSheet ../Hawaii.xml ../GraphML2Fa
Jek.xsl
*Vertices 13
1 "Maui"
2 "Molokai"
3 "Pearl City"
4 "Lai"
5 "Kaneohe"
6 "Honolulu"
7 "Kahului"
8 "Lahaina"
9 "Keohe"
10 "Hana"
11 "Maunaloa"
12 "Kona"
13 "Hilo"
*Edges
1 2 15.0
1 3 20.0
2 3 12.0
2 4 20.0
3 6 13.0
4 5 24.0
5 6 11.0
7 8 22.0
7 9 16.0
7 10 60.0
11 12 31.0
11 13 45.0
12 13 114.0

C:\Documents and Settings\Serge Fleury\Bureau\ba0-29822886\GraphML2FaJek\Saxon>
java -classpath lib\saxon.jar con.icl.saxon.StyleSheet ../Hawaii.xml ../GraphML2Fa
Jek.xsl > test.txt

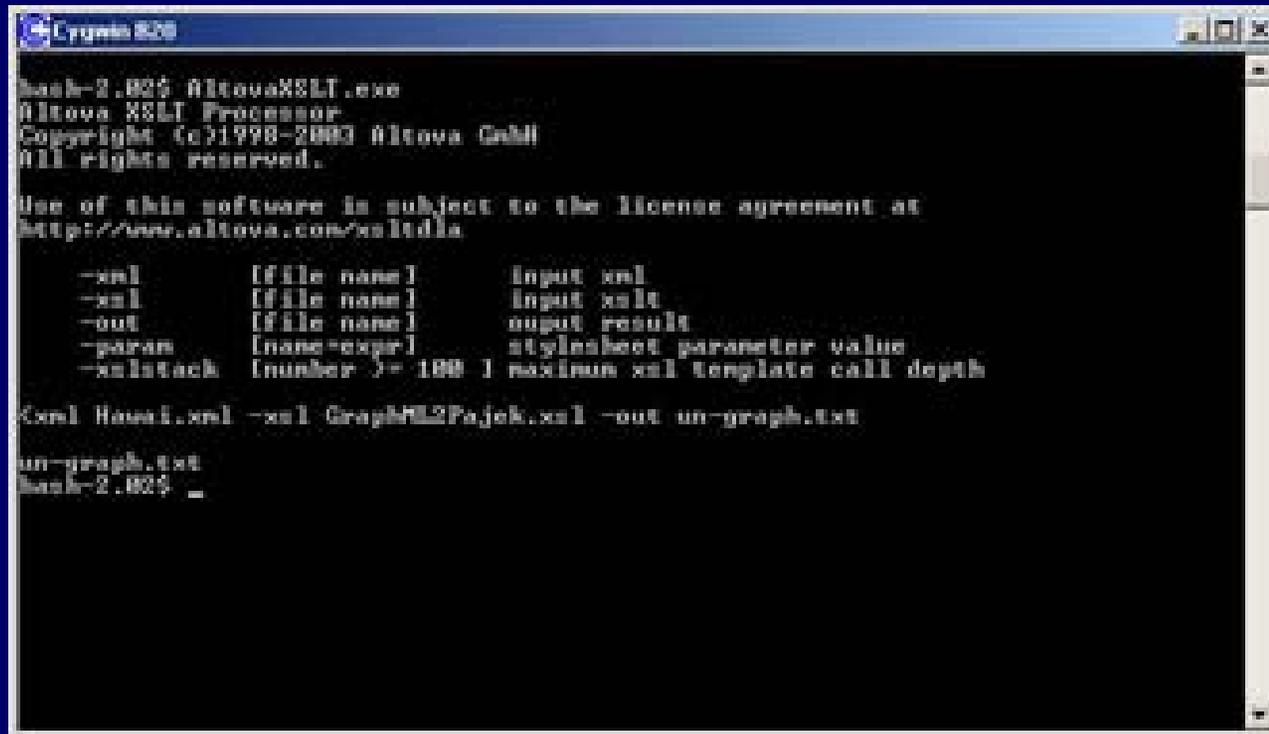
C:\Documents and Settings\Serge Fleury\Bureau\ba0-29822886\GraphML2FaJek\Saxon>

```

# Transformer un document XML (2) *via* une feuille de styles XSLT

## • AltovaXSLT

- Processeur développé par Altova
- DISPONIBLE dans votre archive du cours (à récupérer aussi ici [AltoXSLT.zip](#) ou [là](#))



```
Cygwin 820
bash-2.03$ AltovaXSLT.exe
Altova XSLT Processor
Copyright (c)1998-2003 Altova GmbH
All rights reserved.

Use of this software is subject to the license agreement at
http://www.altova.com/xsltdia

  -xsl      [file name]      input xsl
  -xslt    [file name]      input xslt
  -out     [file name]      output result
  -param  [name=expr]      stylesheet parameter value
  -xslstack [number >= 100] maximum xsl template call depth

xsl Hawaii.xml -xsl GraphML2Pajek.xsl -out un-graph.txt
un-graph.txt
bash-2.03$ _
```

# Transformer un document XML (3) *via* une feuille de styles XSLT

## • AltovaXML

– *Hypertoile* : <http://www.altova.com/altovaxml.html>

– Disponible dans votre ARCHIVE du cours

– *Mode d'emploi* :

• Dans la ligne de commandes, tapez :

```
AltovaXML -xslt1 xsltfile -in xmlfile [-out outputfile] [options]
```

```
AltovaXML -xslt2 xsltfile -in xmlfile [-out outputfile] [options]
```

# Transformer un document XML (4) via une feuille de styles XSLT

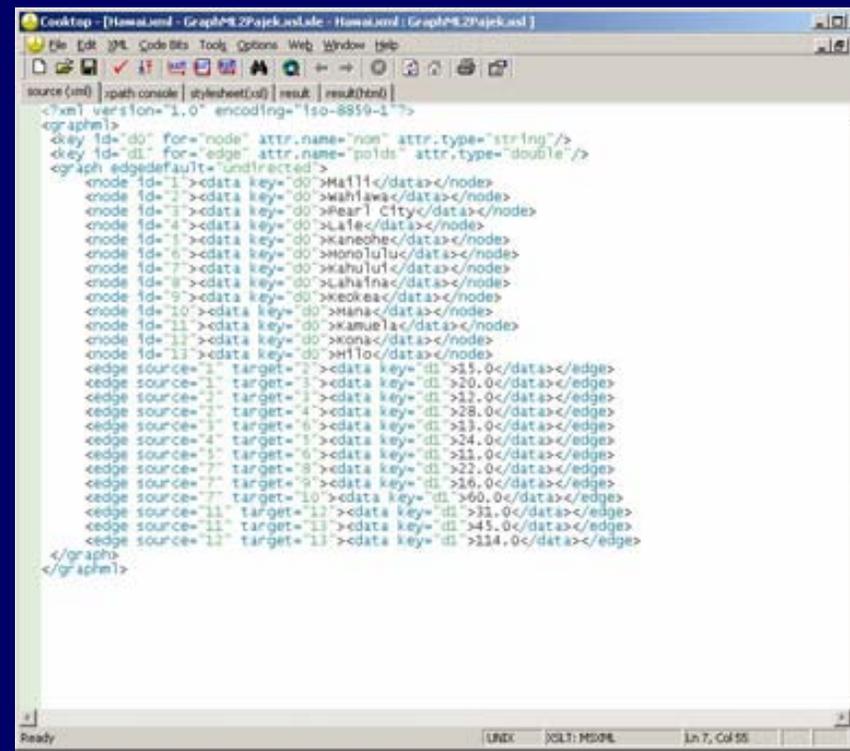
- Avec XMLCooktop

- *Hypertoile* : <http://www.xmlcooktop.com/>

- *Mode d'emploi* :

- Tout se passe dans l'interface proposée par l'outil :

## 1. Onglet XML



```
source.xml | xpath console | stylesheet(xsl) | result | result.html
<?xml version="1.0" encoding="iso-8859-1"?>
<graphml>
  <key id="do" for="node" attr.name="nom" attr.type="string"/>
  <key id="dl" for="edge" attr.name="poids" attr.type="double"/>
  <graph edgedefault="undirected">
    <node id="1"><data key="do">Mallé</data></node>
    <node id="2"><data key="do">Wahlawa</data></node>
    <node id="3"><data key="do">Pearl City</data></node>
    <node id="4"><data key="do">Lale</data></node>
    <node id="5"><data key="do">Kaneohe</data></node>
    <node id="6"><data key="do">Honolulu</data></node>
    <node id="7"><data key="do">Kahului</data></node>
    <node id="8"><data key="do">Lahaina</data></node>
    <node id="9"><data key="do">Keokea</data></node>
    <node id="10"><data key="do">Hana</data></node>
    <node id="11"><data key="do">Kamuela</data></node>
    <node id="12"><data key="do">Kona</data></node>
    <node id="13"><data key="do">Hilo</data></node>
    <edge source="1" target="2"><data key="dl">25.0</data></edge>
    <edge source="1" target="3"><data key="dl">20.0</data></edge>
    <edge source="3" target="1"><data key="dl">12.0</data></edge>
    <edge source="1" target="4"><data key="dl">28.0</data></edge>
    <edge source="1" target="6"><data key="dl">13.0</data></edge>
    <edge source="4" target="1"><data key="dl">24.0</data></edge>
    <edge source="1" target="6"><data key="dl">11.0</data></edge>
    <edge source="7" target="9"><data key="dl">22.0</data></edge>
    <edge source="7" target="9"><data key="dl">16.0</data></edge>
    <edge source="7" target="10"><data key="dl">60.0</data></edge>
    <edge source="11" target="11"><data key="dl">31.0</data></edge>
    <edge source="11" target="13"><data key="dl">45.0</data></edge>
    <edge source="12" target="13"><data key="dl">14.0</data></edge>
  </graph>
</graphml>
```

```

Cooktop - [Hawaii.xml - GraphML2Pajek.xsl: Hawaii.xml - GraphML2Pajek.xsl]
File Edit View Code Bits Tools Options Web Window Help
source.xml zpath console stylesheet.xml result result.html
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="text" encoding="iso-8859-1"/>

<!-- <xsl:text></xsl:text> -->

<xsl:template match="/">
<xsl:text>*Vertices </xsl:text><xsl:value-of select="count(graphml/graph/node)"/><xsl:text>
</xsl:text>
<xsl:apply-templates select="graphml/graph/node"/>
<xsl:text>*Edges
</xsl:text>
<xsl:apply-templates select="graphml/graph/edge"/>
</xsl:template>

<xsl:template match="edge">
<xsl:value-of select="./@source"/><xsl:text> </xsl:text><xsl:value-of select="./@target"/><
</xsl:text>
</xsl:template>

<xsl:template match="node">
<xsl:value-of select="./@id"/><xsl:text> "</xsl:text><xsl:value-of select="./data"/><xsl:ite
</xsl:text>
</xsl:template>
</xsl:stylesheet>
    
```

## 2. Onglet XSL

## 3. Onglet RESULT

```

Cooktop - [Hawaii.xml - GraphML2Pajek.xsl: Hawaii.xml - GraphML2Pajek.xsl]
File Edit View Code Bits Tools Options Web Window Help
source.xml zpath console stylesheet.xml result result.html
*vertices 13
1 "Maui"
2 "Hawaii"
3 "Pearl City"
4 "Lala"
5 "Kaneohe"
6 "Honolulu"
7 "Kahului"
8 "Lahaina"
9 "Keokea"
10 "Hana"
11 "Kamuela"
12 "Kona"
13 "Hilo"
*Edges
1 2 15.0
1 3 20.0
2 3 12.0
2 4 28.0
3 6 13.0
4 5 24.0
5 6 11.0
7 8 22.0
7 9 16.0
7 10 60.0
11 12 31.0
11 13 45.0
12 13 114.0
    
```

# Transformer un document XML (5)

## via une feuille de styles XSLT

### • Avec XML Copy Editor

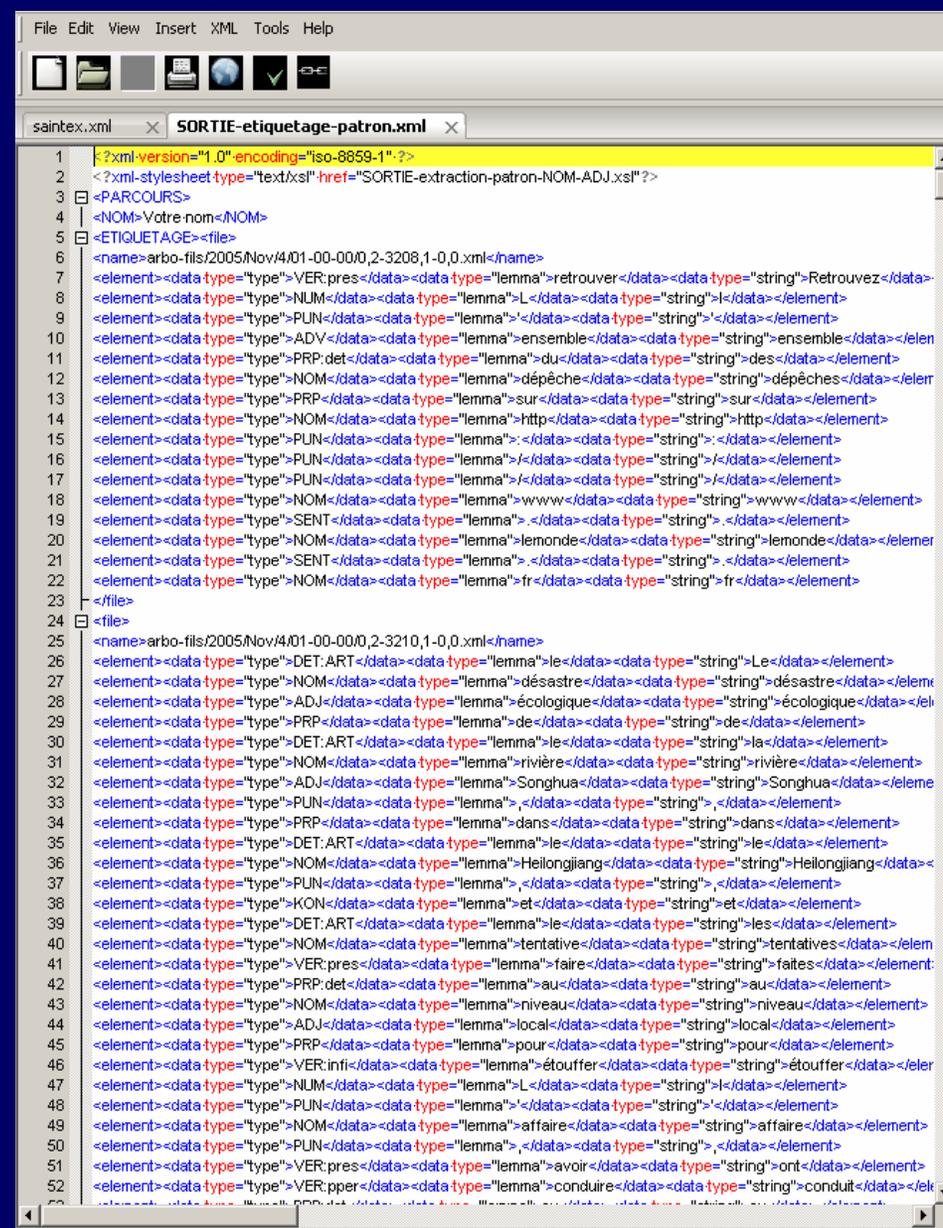
- *Hypertoile* : <http://xml-copy-editor.sourceforge.net/>
- *Mode d'emploi* :

Menu XML

XSL

XSL transform

(ou F8)



```

1  <?xml version="1.0" encoding="iso-8859-1" ?>
2  <?xml-stylesheet type="text/xsl" href="SORTIE-extraction-patron-NOM-ADJ.xsl"?>
3  <PARCOURS>
4  <NOM>Votre nom</NOM>
5  <ETIQUETAGE><file>
6  <name>arbo-fils/2005/Nov/4/01-00-00/0,2-3208,1-0,0.xml</name>
7  <element><data type="type">VER:pres</data><data type="lemma">retrouver</data><data type="string">Retrouvez</data>
8  <element><data type="type">NUM</data><data type="lemma">L</data><data type="string">|</data></element>
9  <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
10 <element><data type="type">ADV</data><data type="lemma">ensemble</data><data type="string">ensemble</data></element>
11 <element><data type="type">PRP:det</data><data type="lemma">du</data><data type="string">des</data></element>
12 <element><data type="type">NOM</data><data type="lemma">dépêche</data><data type="string">dépêches</data></element>
13 <element><data type="type">PRP</data><data type="lemma">sur</data><data type="string">sur</data></element>
14 <element><data type="type">NOM</data><data type="lemma">http</data><data type="string">http</data></element>
15 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
16 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
17 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
18 <element><data type="type">NOM</data><data type="lemma">www</data><data type="string">www</data></element>
19 <element><data type="type">SENT</data><data type="lemma"></data><data type="string"></data></element>
20 <element><data type="type">NOM</data><data type="lemma">lemonde</data><data type="string">lemonde</data></element>
21 <element><data type="type">SENT</data><data type="lemma"></data><data type="string"></data></element>
22 <element><data type="type">NOM</data><data type="lemma">fr</data><data type="string">fr</data></element>
23 </file>
24 <file>
25 <name>arbo-fils/2005/Nov/4/01-00-00/0,2-3210,1-0,0.xml</name>
26 <element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">Le</data></element>
27 <element><data type="type">NOM</data><data type="lemma">désastre</data><data type="string">désastre</data></element>
28 <element><data type="type">ADJ</data><data type="lemma">écologique</data><data type="string">écologique</data></element>
29 <element><data type="type">PRP</data><data type="lemma">de</data><data type="string">de</data></element>
30 <element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">la</data></element>
31 <element><data type="type">NOM</data><data type="lemma">rivière</data><data type="string">rivière</data></element>
32 <element><data type="type">ADJ</data><data type="lemma">Songhua</data><data type="string">Songhua</data></element>
33 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
34 <element><data type="type">PRP</data><data type="lemma">dans</data><data type="string">dans</data></element>
35 <element><data type="type">DET:ART</data><data type="lemma">le</data><data type="string">le</data></element>
36 <element><data type="type">NOM</data><data type="lemma">Heilongjiang</data><data type="string">Heilongjiang</data></element>
37 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
38 <element><data type="type">KON</data><data type="lemma">et</data><data type="string">et</data></element>
39 <element><data type="type">DET:ART</data><data type="lemma">les</data><data type="string">les</data></element>
40 <element><data type="type">NOM</data><data type="lemma">tentative</data><data type="string">tentatives</data></element>
41 <element><data type="type">VER:pres</data><data type="lemma">faire</data><data type="string">faites</data></element>
42 <element><data type="type">PRP:det</data><data type="lemma">au</data><data type="string">au</data></element>
43 <element><data type="type">NOM</data><data type="lemma">niveau</data><data type="string">niveau</data></element>
44 <element><data type="type">ADJ</data><data type="lemma">local</data><data type="string">local</data></element>
45 <element><data type="type">PRP</data><data type="lemma">pour</data><data type="string">pour</data></element>
46 <element><data type="type">VER:infi</data><data type="lemma">étouffer</data><data type="string">étouffer</data></element>
47 <element><data type="type">NUM</data><data type="lemma">L</data><data type="string">|</data></element>
48 <element><data type="type">PUN</data><data type="lemma"></data><data type="string"></data></element>
49 <element><data type="type">NOM</data><data type="lemma">affaire</data><data type="string">affaire</data></element>
50 <element><data type="type">PUN</data><data type="lemma">,</data><data type="string">,</data></element>
51 <element><data type="type">VER:pres</data><data type="lemma">avoir</data><data type="string">ont</data></element>
52 <element><data type="type">VER:pper</data><data type="lemma">conclure</data><data type="string">concluit</data></element>

```

## Phase 2

- Extraction des patrons via un script PERL à tester
  - Présentation générale
  - Présentation du script sur le blog pluriTAL
    - <http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/cours/bao3-extractionterminologique.html>
    - <http://tal-p3.wordpress.com/2006/02/15/boite-a-outils-serie-3/>
    - Ces ressources sont aussi dans votre archive de travail (dossier bao-extraction-terminologique)

# A vous de jouer...

- Les listes de patrons construites ici seront réutilisées dans la *BàO série 4* !