Développer des projets en XML

Les namespaces

Le problème

- XML est extensible
  - donc chacun peut concevoir ses schémas
  - chacun peut concevoir des applications qui supportent ses propres schémas
- mais que ce passe-t-il lorsque l'on combine deux applications différentes ?
  - souvent, des conflits de noms, d'où conflit sémantique

Exemple de conflit

- titre = "Le Seigneur des Anneaux"
-
```
<book>
   <title>The
   Lord of
   the
   Rings</title>
   <type>
      fantasy
   </type>
</book>
```

- titre = Ministre, Comte, Monseigneur
-
```
<person>
   <title>Prince
   </title>
   <name>
      Philippe
  </name>
</person>
```


Conflits

- le pire c'est quand les deux doivent être réutilisés ensemble
  - on ajoute l'élément personne dans l'élément book
- que fait une application dans ce cas-là ?

Solution

- enregistrer les noms d'éléments
  - difficile pour ne pas dire impossible
- inscrire les noms d'éléments dans un contexte plus global
  - les noms d'éléments sont inscrits dans le contexte d'une URI
  - on bénéficie du mécanisme d'enregistrement des URIs

En pratique

- un préfixe devant chaque élément
- le préfixe pointe vers une URI
  - le test d'équivalence se fait sur l'URI
  - majuscules et minuscules comptent
-
```
<ps:person
  xmlns:ps="http://www.psol.com/person/1.0">
  <ps:title>Prince</ps:title>
  <ps:name>Philippe</ps:name>
</ps:person>
```
- défaut déclaré via xmlns tout seul

Uniform Resource Identifier

- Uniform Resource Locator
  - bénéficie de l'enregistrement de noms de domaine
- Uniform Resource Name
  - bénéficie de l'enregistrement propre au schéma choisit
  - urn:ISBN:0-7897-2242-9

Créer une URL

- l'URL agit comme identifiant
  - elle ne doit pas pointer vers une ressource
  - le test se fait sur l'URI, pas la ressource
- adopter des règles de constructions
  - http://www.psol.com/xmlns/order/1.0/en
    - http://www.psol.com/xmlns : préfixe du propriétaire
    - order : spécifie le namespace
    - 1.0 : numéro de version
    - en : langue

Et le DTD?

- l'attribut xmlns:xxx doit être déclaré
- il peut être utile de lui donner une valeur #FIXED
  - mais attention à l'aspect standalone
-
```
<!ELEMENT ps:person (ps:title,
ps:name)>
<!ATTLIST ps:person xmlns:ps CDATA
                    #REQUIRED>
```


Exemple : feuille de styles

- le problème
  - une feuille de style est écrite en XML
  - elle contient
    - des éléments du document d'origine (statuts spécial)
    - des éléments du document de destination
    - des éléments propre au langage de feuille de style
  - comment s'y retrouver ?
- solution : XML namespace

Exemple : XSL

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
 xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
 xmlns="http://www.w3.org/TR/REC-html40">
<xsl:template match="/">
   <HTML><HEAD><TITLE>Article</TITLE></HEAD>
   <BODY><xsl:apply-templates/></BODY></HTML>
</xsl:template>
<xsl:template match="title">
   <P><B><xsl:apply-templates/></B></P>
</xsl:template>
<xsl:template match="p">
   <P><xsl:apply-templates/></P>
</xsl:template>
</xsl:stylesheet>
```


Exemple : liens

- il s'agit d'insérer dans un document XML, des élément définis par ailleurs
  - la sémantique de ces éléments est définies par le W3C
  - un logiciel doit être capable de les reconnaître et d'appliquer la sémantique même s'il ne reconnaît pas le reste du document
- solution : XML namespace

Exemple : XLink

```
<?xml version="1.0"?>
<info>
<para>XLink links XML documents. It supports simple
  links, which are very similar to HTML links, but it
  also supports more advanced links.</para>
<para>For more information on XLink, you can visit
  <xlink:simple
  xmlns:xlink="http://www.w3.org/XML/XLink/0.9"
  href="http://www.w3.org/TR/xlink"
  role="recommendation" title="XML Linking Language
  (XLink)" show="replace" actuate="user">the W3C
  site</xlink:simple></para>
</info>
```