Développer des projets en XML

La gestion des liens en XML
XLink, XPointer et XFragment

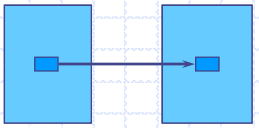
Etablir des liens entre des documents XML

- Norme proposée par le consortium W3C pour l'établissement de relations entre des ressources. (statut : "recommandation" depuis 27 Juin 2001)
- Ressources référencées peuvent être :
  - Document XML
  - Parties de document XML
  - Fichier HTML, images, …
- Normes définit mécanisme/concepts mais pas le mode de rendu par les navigateurs.

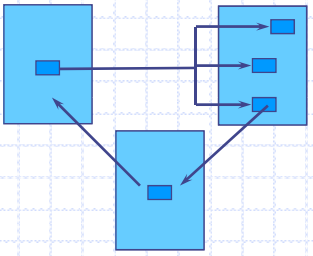
Liens en HTML

- Pour rappel en HTML, liens :
  - <a href="" (avec <a name="")
  - <img src=""
  - <link href=""
  - <object classid="" codebase=""
  - <applet code="" codebase=""
  - <script src=""
- Le lien :
  - Activé automatiquement
  - Activé par l'utilisateur.
  - Remplace ou augmente la ressource locale

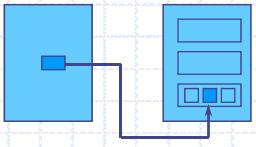
XLink, différents type de liens



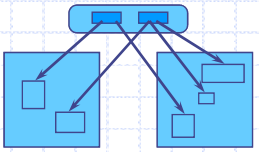
Lien simple



Lien étendu



Lien vers parties de document

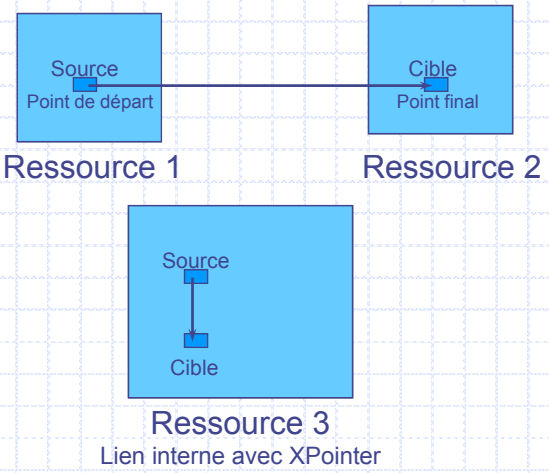


Base externe de liens

Langage XLink

- Le langage XLink ne prévoit pas d'éléments mais uniquement des attributs (globaux) utilisés avec d'autres éléments.
- Namespace :
  - http://www.w3.org/1999/xlink
  - Préfixe : xlink

XLink, lien simple



XLink, exemple de lien simple

- Exemple de lien simple, extrait d'un document XML (par ex. biblio.xml):

```
<auteurs
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xlink:href="auteurs.xml"
   xlink:role="Liste des auteurs"
   xlink:title="Expert-IT - Liste des auteurs"
   xlink:show="replace"
   xlink:actuate="onRequest"
/>
```


XLink, exemple de lien simple

- Liens avec le document "auteurs.xml"
- Activé à la requête de l'utilisateur
- Sera visualisé dans la même instance de l'agent utilisateur (par ex.: la fenêtre du navigateur)

XLink, exemple de lien simple

- Autre exemple de lien simple:

```
<auteurs
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xlink:href="auteurs.xml"
  xlink:role="Liste des auteurs"
  xlink:title="Expert-IT - Liste des auteurs"
  xlink:show="embedded"
  xlink:actuate="onLoad"
/>
```


XLink, exemple de lien simple

- Liens avec le document "auteurs.xml"
- Activé au chargement du document
- Sera intégré dans le document courant

XLink et DTD

- Si document validé par DTD, les attributs doivent être définis dans le DTD.

```
<!ATTLIST auteurs
xmlns:xlink     CDATA  #FIXED "http://www.w3.org/1999/xlink"
xlink:type      CDATA          #FIXED "simple"
xlink:href      CDATA          #IMPLIED
xlink:role      CDATA          #IMPLIED
xlink:arcrole   CDATA          #IMPLIED
xlink:title     CDATA          #IMPLIED
xlink:show      (new|replace|embed|other|none)      #IMPLIED
xlink:actuate   (onLoad|onRequest|other|none)       #IMPLIED>
```


Attributs pour type simple

- xlink:type = simple
- xlink:title : information à l'intention de l'utilisateur
- xlink:href : l'URI pour localiser la ressource (en général URL avec ou sans expression XPointer)
- xlink:role : URI (simple texte possible) décrivant la fonction du contenu du lien
- xlink:arcrole : URI (simple texte possible) décrivant la fonction du lien lui-même (par ex.: Page Suivante)

Attributs pour type simple

- xlink:show : où visualiser la ressource référencée
  - new : nouvelle instance agent utilisateur
  - embedded : inclus dans le document source à la place du point de départ
  - replace : remplace le document source (défaut)
  - other : non spécifié dans le lien, mais par d'autres moyens (éléments spécifiques dans document cible)
  - none : pas spécifié dans le lien
- xlink:actuate : quand activer le lien
  - onRequest : à la demande de l'utilisateur
  - onLoad : au chargement du document
  - none et other : idem xlink:show

Liens étendus

- xlink:type = extended
- Permet de relier plusieurs ressources entre elles
- L'expression de ces relations peut être externalisée dans un document qui ne fait pas partie des documents sources.

Définition d'un lien de type étendu

- Elément de type "lien étendu" dispose d'éléments enfants ou "participants" qui spécifient les liens.
- Les types sont :
  - locator
  - resource
  - arc
  - title

XLink, Lien simple et lien étendu

- Lien simple:

```
<item xlink:type="simple" xlink:href="bio.xml">Biographie de
    Brassens</item
```

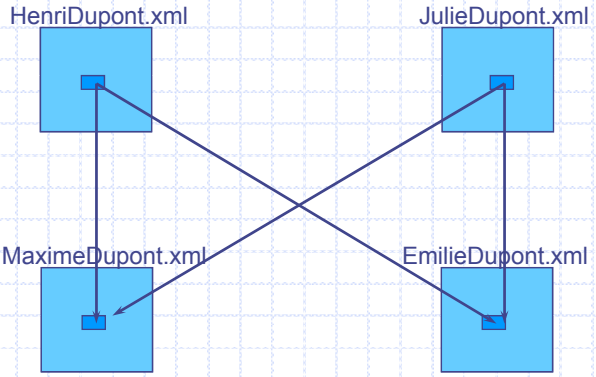
- Lien étendu :

```
<item xlink:type="extended"
    xmlns:xlink="http://www.w3.org/1999/xlink">
<loc xlink:type="resource" xlink:label="local" xlink:title="Brassens">
    Biographie de Brassens</loc>
<rem xlink:type="locator" xlink:href="bio.xml" xlink:label="remote"
xlink:title="Biographie" />
<arc xlink:type="arc" xlink:from="local" xlink:to="remote"
xlink:show="replace" xlink:actuate="onRequest" />
</item>
```

Exemple de lien étendu

```
<famille xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="extended" xlink:role="Famille"
    xlink:title="La famille Dupont">
<personne xlink:type="locator" xlink:href="HenriDupont.xml"
    xlink:label="parent" xlink:title="Henri Dupont" />
<personne xlink:type="locator" xlink:href="JulieDupont.xml"
    xlink:label="parent" xlink:title="Julie Dupont" />
<personne xlink:type="locator" xlink:href="MaximeDupont.xml"
    xlink:label="enfant" xlink:title="Maxime Dupont" />
<personne xlink:type="locator" xlink:href="EmilieDupont.xml"
    xlink:label="enfant" xlink:title="Emilie Dupont" />
<relations xlink:type="arc" xlink:from="parent" xlink:to="enfant"
    xlink:title="Voir les enfants" xlink:show="replace"
    xlink:actuate="onRequest" />
</famille>
```

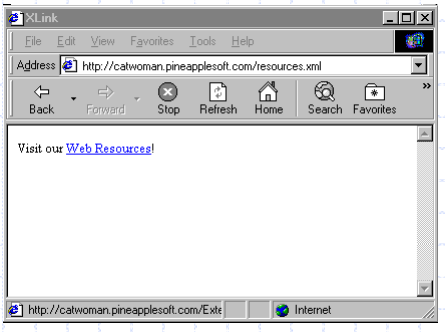

Exemple de lien étendu



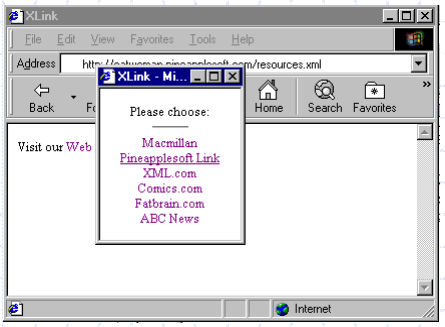
Liste des types d'attributs XLink et autres attributs nécessaires

	simple	extended	locator	arc	resource	title
type	obligatoire	obligatoire	obligatoire	obligatoire	obligatoire	obligatoire
href	optionnel		obligatoire			
role	optionnel	optionnel	optionnel		optionnel	
arcrole	optionnel			optionnel		
title	optionnel	optionnel	optionnel	optionnel	optionnel	
show	optionnel			optionnel		
actuate	optionnel			optionnel		
label			optionnel		optionnel	
from				optionnel		
to				optionnel		

Liens étendus, simulation (I)



Liens étendus, simulation (II)



Externalisation des liens

- Fichier "france.xml" :

```
<ext xmlns:xlink="http://www.w3.org/1999/xlink"
    xlink:type="extended">
<lien xlink:type="resource" xlink:label="pays"
xlink:title="Pays">France</lien>
<répertoire xlink:type="locator" xlink:href="departements.xml"
xlink:label="départements" xlink:title="Départements" />
<arc xlink:type="arc" xlink:from="pays" xlink:to="départements"
xlink:arcrole="http://www.w3.org/1999/xlink/properties
    /linkbase"
xlink:show="replace" xlink:actuate="onRequest" />
</ext>
```

Externalisation des liens

- Document "departements.xml":

```
<departements xmlns:xlink="http://www.w3.org/1999/xlink"
   xlink:type="extended" xlink:title="Liens départements">
<source xlink:type="locator" xlink:href="france.xml"
   xlink:label="france" xlink:title="France"/>
<cible xlink:type="locator" xlink:href="paris.xml"
   xlink:label="paris" xlink:title="Paris"/>
<lien xlink:type="arc" xlink:from="france" xlink:to="paris"
   xlink:show="new" xlink:actuate="onRequest" />
...
</departements>
```


XPointer, langage d'adressage

- XPointer est un langage **non-XML** qui définit un schéma d'adressage pour accéder aux différentes parties d'un document XML.
- Version 1.0, W3C Candidate Recommendation 11 September 2001

XPointer, langage d'adressage

- Un lien XLink référence via une URI (URL) qui permet de localiser une ressource.
- Cette URL peut inclure un expression XPointer permettant d'identifier une partie, un élément du document cible.
- XPointer est basé sur XPath et utilise la même syntaxe ajoutée de quelques fonctionnalités complémentaires.

Utilisation de XPointer

- Adresser des éléments ou groupes d'éléments dans un document :
  - Un élément avec identifiant (ID)
  - Tous les éléments qui possèdent un certain attribut
  - Le premier élément d'un certain type
  - Le dernier élément ayant une certaine valeur
  - Le septième élément d'un certain type
  - Le premier enfant d'un certain élément
  - ...

Exemple d'expression XPointer

- xpointer(id("ebnf"))
- xpointer(descendant::language[position()=2])
- xpointer(/child::spec/child::body/child::*/child::language[position()=2]) /1/14/2
- xpointer(id("EBNF"))

XPointer et XLink

- XPointer est combiné à une URL, à la suite du caractère #, comme identificateur de "fragment" :
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#xpointer(id("ebnf"))
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#xpointer(descendant::language[position()=2])
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#ebnf
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#xpointer(/child::spec/child::body/child::*/child::language[position()=2])
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#/1/14/2
  - http://www.w3.org/TR/1998/REC-xml-19980210.xml#xpointer(id("ebnf"))xpointer(id("EBNF"))

XPointer et XLink

```
<SPECIFICATION
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xlink:type="simple"
   xlink:href="http://www.w3.org/TR/1998/REC-
   xml-19980210.xml#xpointer(id('ebnf'))">
   xlink:actuate="onRequest"
   xlink:show="replace">
   Extensible Markup Language (XML) 1.0
   </SPECIFICATION>
```


Document exemple

```
<?xml version="1.0"?>
<!DOCTYPE FAMILYTREE [
      <!ELEMENT FAMILYTREE (PERSON | FAMILY)*>
      <!-- PERSON elements -->
      <!ELEMENT PERSON (NAME*, BORN*, DIED*, SPOUSE*)>
      <!ATTLIST PERSON ID ID #REQUIRED FATHER CDATA #IMPLIED MOTHER
         CDATA #IMPLIED >
      <!ELEMENT NAME (#PCDATA)>
      <!ELEMENT BORN (#PCDATA)>
      <!ELEMENT DIED (#PCDATA)>
      <!ELEMENT SPOUSE EMPTY>
      <!ATTLIST SPOUSE IDREF IDREF #REQUIRED>
      <!--FAMILY-->
      <!ELEMENT FAMILY (HUSBAND?, WIFE?, CHILD*) >
      <!ATTLIST FAMILY ID ID #REQUIRED>
      <!ELEMENT HUSBAND EMPTY>
```

Document exemple

```
        <!ATTLIST HUSBAND IDREF IDREF #REQUIRED>
        <!ELEMENT WIFE EMPTY>
        <!ATTLIST WIFE IDREF IDREF #REQUIRED>
        <!ELEMENT CHILD EMPTY>
        <!ATTLIST CHILD IDREF IDREF #REQUIRED> ]>
<FAMILYTREE>
<PERSON ID="p1">
<NAME>Domeniquette Celeste Baudean</NAME>
<BORN>21 Apr 1836</BORN> <DIED>Unknown</DIED>
<SPOUSE IDREF="p2"/></PERSON>
<PERSON ID="p2">
<NAME>Jean Francois Bellau</NAME>
<SPOUSE IDREF="p1"/> </PERSON>
<PERSON ID="p3" FATHER="p2" MOTHER="p1">
<NAME>Elodie Bellau</NAME>
```


Document exemple

```
<BORN>11 Feb 1858</BORN> <DIED>12 Apr 1898</DIED>
<SPOUSE IDREF="p4"/> </PERSON>
<PERSON ID="p4" FATHER="p2" MOTHER="p1">
<NAME>John P. Muller</NAME> <SPOUSE IDREF="p3"/> </PERSON>
<PERSON ID="p7"> <NAME>Adolf Eno</NAME>
<SPOUSE IDREF="p6"/> </PERSON>
<PERSON ID="p6" FATHER="p2" MOTHER="p1"> <NAME>Maria Bellau</NAME>
    <SPOUSE IDREF="p7"/> </PERSON>
<PERSON ID="p5" FATHER="p2" MOTHER="p1">
<NAME>Eugene Bellau</NAME> </PERSON>
<PERSON ID="p8" FATHER="p2" MOTHER="p1">
<NAME>Louise Pauline Bellau</NAME> <BORN>29 Oct 1868</BORN>
<DIED>3 May 1938</DIED> <SPOUSE IDREF="p9"/> </PERSON>
<PERSON ID="p9"> <NAME>Charles Walter Harold</NAME>
<BORN>about 1861</BORN> <DIED>about 1938</DIED>
```

Document exemple

```
<SPOUSE IDREF="p8"/> </PERSON>
<PERSON ID="p10" FATHER="p2" MOTHER="p1">
<NAME>Victor Joseph Bellau</NAME> <SPOUSE IDREF="p11"/> </PERSON>
<PERSON ID="p11"> <NAME>Ellen Gilmore</NAME>
<SPOUSE IDREF="p10"/> </PERSON>
<PERSON ID="p12" FATHER="p2" MOTHER="p1">
<NAME>Honore Bellau</NAME> </PERSON>
<FAMILY ID="f1">
<HUSBAND IDREF="p2"/> <WIFE IDREF="p1"/> <CHILD IDREF="p3"/>
<CHILD IDREF="p5"/> <CHILD IDREF="p6"/> <CHILD IDREF="p8"/>
<CHILD IDREF="p10"/> <CHILD IDREF="p12"/>
</FAMILY>
<FAMILY ID="f2">
<HUSBAND IDREF="p7"/> <WIFE IDREF="p6"/>
</FAMILY>
</FAMILYTREE>
```


Expression XPointer, chemin de localisation

- Une expression XPointer est un chemin de localisation (idem à XPath).
- Un chemin de localisation est composé d'étapes (niveau) de localisation.
- Chaque étape spécifie un point dans le document cible.
- Le chemin est calculé à partir d'un point de départ, le nœud contextuel :
  - La racine du document
  - Une autre étape (nœud) de localisation

Expression XPointer

- Une étape de localisation est composée trois parties:
  - L'axe de sélection
  - Le nœud-test
  - Un prédicat optionnel
    - *axe*::*noeud-test*[*prédicat*]
    - child::PERSON[position()=2]
- L'axe indique dans quelle direction chercher à partir du nœud contextuel.
- Le nœud de test indique quel(s) nœud(s) (jeu) à considérer sur l'axe.
- Le prédicat est un test appliqué à chaque nœud du jeu, si la valeur retour est "faux", le nœud est retiré du jeu.

Expression XPointer

- Expression :
  xpointer(/child::FAMILYTREE/child::PERSON[position()=3])
- Le chemin de localisation est :
  /child::FAMILYTREE/child::PERSON[position()=3].
- Composé de deux étapes de localisation :
  /child::FAMILYTREE
  child::PERSON[position()=3]
- Un seul nœud identifié :

```
<PERSON ID="p3" FATHER="p2" MOTHER="p1">
<NAME>Elodie Bellau</NAME> <BORN>11 Feb 1858</BORN>
<DIED>12 Apr 1898</DIED> <SPOUSE IDREF="p4"/>
</PERSON>
```

Expression XPointer

- xpointer(/child::FAMILYTREE/child::PERSON[position()>3])
- Identifie tous les éléments PERSON dont la position est supérieure à 3

Les axes de sélection

- Douze axes de sélection sont définis dans XPath et donc XPointer : child (défaut), ancestor, descendant, self, ancestor-or-self, descendant-or-self, attribute
- Exemple :
  - id("p6")/child::NAME
- Fonction id() retourne un jeu de nœuds avec l'élément ayant un attribut "id" à la valeur "p6"
- Cet élément sert de nœud contextuel pour l'étape de localisation suivante qui démarre par un certain axe de localisation.

Les nœuds-test

- Dix nœuds-test dans XPointer (8 de XPath) :
  - Valeurs :
    - *nomélément*
    - *
    - *préfixe*:*
    - @*nomattribut*
  - Fonctions :
    - node()
    - text()
    - comment()
    - processing-instruction()
    - point()
    - range()
    - here(), origin()
- /descendant::body/child::*/attribute::xlink:*

Prédicats

- Chaque étape de localisation peut contenir zéro ou plusieurs prédicats.
- Un prédicat précise et réduit les nœuds résultats en éliminant les nœuds ne répondant pas aux prédicats.
- Chaque prédicat contient une expression booléenne exprimée entre [ ]. Exemples :
  - Tous les éléments ayant une certaine valeur d'attribut
  - Le premier éléments contenant un certain élément enfant
  - Un élément contenant une certaine chaîne de caractère.
  - Tous les éléments avec valeur numérique.
  - ...

Fonctions pour expression XPointer

- **position()** : retourne l'index du nœud dans le jeu de nœud contextuel
  - xpointer(/child::FAMILYTREE/child::*[position()=1])

  ou autre version
  - xpointer(/child::FAMILYTREE/child::*[1])

Fonctions dans expression XPointer

- **id(*valeur*)** : sélectionne l'élément dont l'attribut de type ID à la valeur spécifiée
- Par exemple:
  - http://serveur/genealogy.xml#xpointer(id("p12"))

  Ou version simplifiée
  - http://serveur/genealogy.xml#p12

  Résultat :

  <PERSON ID="p12" FATHER="p2" MOTHER="p1">
  <NAME>Honore Bellau</NAME> </PERSON>

Fonctions dans expression XPointer

- **here()** : retourne l'élément courant (spécifique XPointer)

```
<?xml version="1.0"?>
<SLIDESHOW xmlns:xlink="http://www.w3.org/1999/xlink">
    <SLIDE> <H1>Welcome to the slide show!</H1> <BUTTON
    xlink:type="simple" xlink:href="here()/following::SLIDE[1]"> Next
    </BUTTON> </SLIDE> <SLIDE> <H1>This is the second
    slide</H1> <BUTTON xlink:type="simple"
    xlink:href="here()/preceding::SLIDE[1]"> Previous </BUTTON>
    <BUTTON xlink:type="simple"
    xlink:href="here()/following::SLIDE[1]"> Next </BUTTON>
    </SLIDE>
...
```


Fonctions dans expression XPointer

- **origin()** : cette fonction référence l'élément dans le document source à partir duquel l'utilisateur a activé le lien.
- Peut être utilisé dans le document cible, là où l'élément source du lien n'est plus présent, pour implémenter des liens de "retour".

Notion de "point" en XPointer

- Un point est la localisation
  - entre un nœud et un autre,
  - ou entre un caractère et un autre caractère dans un contenu textuel parsé d'élément
- Pour le document suivant:

  <salutation>Bonjour vous</salutation>
  - Il existe trois nœuds : nœud racine, nœud élément salutation, nœud textuel

Notion de "point" en XPointer

- Pour le document suivant (suite):

  <salutation>Bonjour vous</salutation>
  - Il existe 18 points :
    - Le point avant le nœud racine
    - Le point avant le nœud élément "salutation"
    - Le point avant le nœud textuel
    - Le point avant la lettre "B"
    - Le point entre "B" et "o" (et ainsi de suite, plusieurs espaces ramené à un espace)
    - Le point après l'élément "salutation
    - Le point après l'élément racine.

Notion d'ensemble, "range" en XPointer

- Un "range" débute à un "point" (*start-point*) dans le document XML jusqu'à un autre "point" (*end-point*).
- Les "points" sont référencés par des chemins de localisation.
- Utile pour délimiter, par exemple, la sélection "souris" faite par un utilisateur qui ne débute et ne se termine pas précisément sur des éléments, au sens XML du terme.

Notion d'ensemble, "range" en XPointer

- Pour spécifier un *range*, on utilise /range-to(*end-point*) qui est ajouté à un chemin de localisation qui est lui le *start-point*
- Autres fonctions utiles pour "range" :
  - range()
  - range-inside()
  - start-point()
  - end-point()

Notion d'ensemble, "range" en XPointer

- Exemple:

  xpointer(/child::PERSON[position() = 1]/range-to(/child::PERSON[position() = last()]))

  - ensemble de toutes les personnes

  start-point(range-inside(/child::FAMILYTREE/descendant::*[position()=1]/child::NAME/))

  - Point juste avant la lettre D du texte du premier élément

```
<FAMILYTREE>
<PERSON ID="p1">
<NAME>Domeniquette Celeste Baudean</NAME>
<BORN>…
```


Ensemble chaînes de caractères

- **string-range()** : retourne un jeu de nœuds contenant un ensemble pour chaque occurrence d'une chaîne de caractères
- Syntaxe :

  string-range(jeu-noeuds,chaîne,index,longueur)
- Exemple:

  xpointer(string-range(/,"Harold"))

  - Retrouve toutes les occurrences de "Harold" dans le document XML, tout élément confondu

Ensemble chaînes de caractères

- Exemple:

  xpointer(string-range(//NAME,"Harold"))
  - Retrouve toutes les occurrences de "Harold" dans le document XML, tout élément confondu

  xpointer(string-range(/,"Harold",6)[position()=1])
  - Pointe immédiatement après la première occurrence de "Harold"

Ensemble chaînes de caractères

- Exemple:

  xpointer(string-range(/,"Harold",4,3)[position()=1])
  - Sélectionne la chaîne "old" de la première occurrence de "Harold"

XPointer et les namespaces

- Pour utiliser un namespace dans une expression XPointer, on doit utiliser la fonction xmlns()
- Exemple :

  xmlns(svg=http://www.w3.org/2000/svg)

  xpointer(//svg:polygon[3])

XPointer et les séquences d'enfants

- XPointer offre une syntaxe abrégée pour référencer des éléments en spécifiant un séquence d'élément et d'éléments enfants par leur position

  http://serveur/genealogy.xml#/1/4

  - /1/4 est une séquence précisant la sélection du 4ème enfant du 1er enfant du nœud racine
- Une séquence peut débuter par un ID
  - p4/1 pointe sur le 1er enfant de l'élément avec ID "p4"
- Uniquement utile pour référencer des éléments par leur position.

XPointer et la codification des caractères dans l'URL

- Les caractères supportés dans une expression XPointer sont Unicode, comme XML
- URI (et donc URL) nécessite la codification des caractères non-ascii et d'autres caractères spéciaux
- Le caractère doit être remplacé par la séquence % suivie de la valeur hexadécimale du caractère en UTF-8.

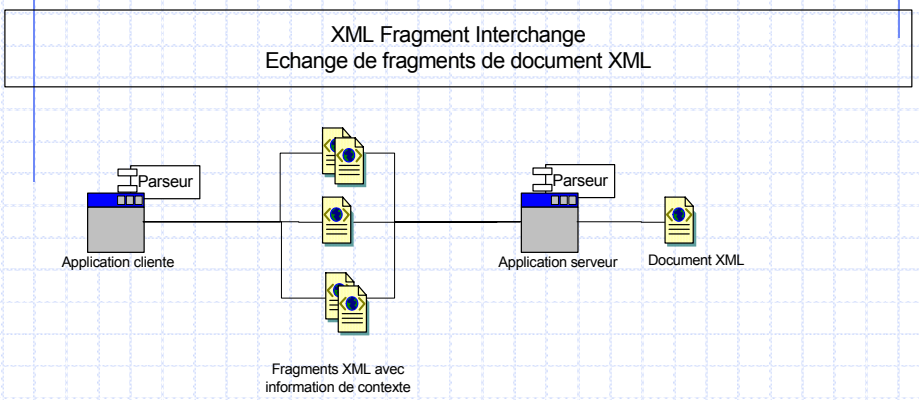
XPointer et la codification des caractères dans l'URL

- Exemple:
  - doc.xml#xpointer(id('résumé'))
    devient
    doc.xml#xpointer(id('r%C3%A9sum%C3%A9'))
  - doc.xml#xpointer(string-range(//P,"le début"))
    devient
    doc.xml#xpointer(string-range(//P,%22le%20d%C3%A9but%22))

Echange entre applications de fragments XML

- XML Fragment Interchange: proposition du W3C (non encore normalisée)
- Objectifs :
  - Fragment = partie de document XML
  - Permet de normaliser l'identification de partie de document XML
  - Permettre à un programme serveur de transmettre une partie d'un document XML à un programme client
  - Le fragment est transmis de manière formalisée avec tout le contexte nécessaire pour l'exploitation

Echange de fragments XML



Fragment XML

- Un fragment :
  - Parties bien-balancées et contiguës d'un document XML
  - Bien balancé : un élément doit être ouvert et fermé au sein du même fragment
  - Non bien-formé
  - Contenu textuel à la racine possible
  - Contenu uniquement texte supporté
  - Multiples éléments racines possible
  - Contigu : voisinage géographique dans le fichier original

Exemples de fragments

- Document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
          <Mot-clé>Internet<Mot-clé>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
          <Mot-clé>Publication Web<Mot-clé>
          <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```

Exemples de fragments valides

- Fragment valide du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```


Exemples de fragments valides

- Fragment valide du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```

Exemples de fragments valides

- Fragment valide du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```


Exemples de fragments non-valides

- Fragment **non-valide** du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```

Exemples de fragments non-valides

- Fragment **non-valide** du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```


Exemples de fragments non-valides

- Fragment **non-valide** du document original

```
<Catalogue>
    <Livre couleur="rouge">
    <Titre>Référence programmeur XML</Titre>
    <Pages>480</Pages>
    <ISBN>1-23456-66-55</ISBN>
    <Mots-clés>
            <Mot-clé>Internet<Mot-clé>
            <Mot-clé>Publication Web<Mot-clé>
            <Mot-clé>XML<Mot-clé>
    </Mots-clés>
    <Prix monnaie="euro">45</Prix>
    </Livre>
</Catalogue>
```

Intérêts des fragments XML

- Conservation des ressources :
  - Transmission/exploitation uniquement de l'information nécessaire
  - Économie mémoire, bande passante, …
- Récolter les parties pertinentes d'information
  - Principalement pour recherche textuelle
  - Recherche de jeu de nœuds (éléments) pas toujours possible car un fragment contient uniquement des éléments contigus.

Intérêts des fragments XML

- Edition en parallèle de document XML volumineux :
  - Parties contiguës du document sont éditées en parallèle avec :
    - Système de check-in, check-out
    - Gestion de versions
    - Certains serveurs XML se base déjà sur cette norme

Utilisation des fragments

- Le programme réceptionnant les fragments doit pouvoir disposer d'information contextuelle:
  - Restituer le fragment dans son contexte
  - Connaître la parenté
  - Valider le fragment
  - Utilisation des identifiants pour ID et IDREF

Information contextuelle

- La norme XML Fragment Interchange prévoit la constitution d'informations de contexte transmises avec le fragment XML
- Données de contexte transmises dans un fichier à part ou dans le même fichier que le fragment

Information contextuelle

- Données de contexte possibles:
  - URI du document original
  - URI du DTD du document original
  - Spécification concernant la localisation du fragment dans le document original
  - Informations concernant :
    - Ancêtres et frères des ancêtres du fragment
    - Éléments frères
  - Tous les éléments de la structure à l'exception de ceux du fragment

Information contextuelle

- Le choix et la création des données de contexte sont décidées par le programme serveur
- Objectif : aider le programme récepteur à interpréter et exploiter le fragment
- Pour la spécification du contexte:
  - Namespace : http://www.w3.org/2001/02/xml-fragment
  - Élément "fcs" (fragment context specification) contient les données du contexte

Exemple de fragment avec données contextuelles

- Données de contexte :

```
<f:fcs xmlns:f="http://www.w3.org/2001/02/xml-
   fragment" parentref="http://www.expert-
   it.com/Catalogue/Catalogue.xml">
<Catalogue>
<Livre>
   <f:fragbody fragbodyref=" http://www.expert-
   it.com/Catalogue/ISBN.xml " />
</Livre>
</Catalogue>
```


Exemple de fragment avec données contextuelles

- Fragment (ISBN.XML) :

```
<ISBN>1-23456-66-55</ISBN>
```