

Fiche XML n°7

XLink/Xpointer & XInclude : référencer et composer

Cours

1. XLink

1.1 Présentation

Utiliser XML pour mémoriser des données est une approche intéressante et utile. Seulement, les données à manipuler sont parfois très volumineuses, voire sur des serveurs distants, fragmentées. Aussi, le W3C propose un mécanisme inspiré des hyperliens HTML : XLink¹. Ce « langage » (en fait une collection d'attributs) permet de mettre en relation une ou plusieurs ressources avec une ou plusieurs autres ressources. Couplé avec XPointer, que nous verrons plus loin, les ressources peuvent être des éléments XML.

Ce standard permet de mettre en relation plusieurs documents (ou ressources), pas nécessairement en XML. De plus, ces liens sont indépendant des documents mis en relation. XLink est une syntaxe basée sur des attributs XML définis sur l'espace de noms : «<http://www.w3.org/1999/xlink>».

D'un point de vue théorique, XLink est une syntaxe XML pour décrire des graphes orientés et étiquetés. Dans ces graphes, les sommets sont les ressources et les arêtes les liens entre les ressources.

XLink propose deux versions de liens : les liens simples et les liens étendus. Il permet de représenter : des liens classiques « à la HTML » (liens simples), des tables des matières, des index...

1.2 Liens simples

Le lien simple XLink est un ensemble d'attributs permettant de décrire un lien vers une ressource depuis un élément XML quelconque et son utilisation. Cette catégorie de lien est de même type que l'hyperlien HTML. Au minimum, deux attributs sont nécessaires :

- «type» avec la valeur «simple» (c'est un lien simple) ;
- «href» qui reçoit une URI (URL ou autre URI comme par exemple «urn:isbn:15513615698») ; c'est le lien proprement dit.

Dans l'exemple ci-contre, les parties XLink sont colorées en rouge. Dans l'élément «identité», l'espace de nom XLink est défini et associé au préfixe «xlink». Ensuite, dans les éléments «site-web», l'attribut XLink «href», identifié par le préfixe spécifique, permet de référencer l'URL de ce site Web.

```
<identité xmlns:xlink="http://www.w3.org/1999/xlink">
  <nom>Asimov</nom>
  <prenom>Isaac</prenom>
  <date-naissance>1920-01-02</date-naissance>
  <date-deces>1992-04-06</date-deces>
  <nationalite>Russe/Américain</nationalite>
  <webographie>
    <site-web lang="fr"
      xlink:href="http://www.asimov.fr/"
      xlink:type="simple"/>
    <site-web lang="en"
      xlink:href="http://www.asimovonline.com/"
      xlink:type="simple"/>
  </webographie>
</identité>
```

¹ <http://www.w3.org/XML/Linking>, <http://www.w3.org/TR/xlink/>

Il est possible de compléter les attributs précédents par d'autres décrivant le comportement des liens. Ainsi, il existe les deux attributs suivants :

- «show» qui indique la manière d'exploiter le lien quand il est activé :
 - «new» (par défaut) : affiche le contenu de l'URI dans une nouvelle fenêtre,
 - «replace» : affiche dans la fenêtre courante (remplace le contenu),
 - «embed» : inclut la ressource dans le document courant au niveau du lien,
 - «other» : autre comportement défini dans une balise dédiée,
 - «none» : ne rien faire ;
- «actuate» qui indique à quel moment activer le lien :
 - «onLoad» : lien suivi dès que l'application le voit,
 - «onRequest» (par défaut) : suivi sur demande de l'utilisateur,
 - «other» : autre comportement défini dans une balise dédiée,
 - «none» : pas d'explication.

Un lien HTML classique sera : «actuate='onRequest'» et «show='replace'». Ainsi, l'exemple précédent peut devenir le document XML ci-dessous. D'autres attributs, plus orientés vers la sémantique existent :

- «title» qui est un court texte décrivant la ressource ;
- «role» donnant l'URI où trouver la description ou l'annotation de la ressource distante.

```
<identité xmlns:xlink="http://www.w3.org/1999/xlink">
  <nom>Asimov</nom>
  <prenom>Isaac</prenom>
  <date-naissance>1920-01-02</date-naissance>
  <date-deces>1992-04-06</date-deces>
  <nationalite>Russe/Américain</nationalite>
  <webographie>
    <site-web
      xlink:role="Principal site sur I. Asimov en français"
      xlink:title="ISAAC ASIMOV"
      xlink:href="http://www.asimov.fr/"
      xlink:type="simple"/>
    <site-web
      xlink:role="Site majeur sur I. Asimov en anglais"
      xlink:title="Isaac Asimov Home Page"
      xlink:href="http://www.asimovonline.com/"
      xlink:type="simple"/>
  </webographie>
</identité>
```

2. XLink étendu

XLink propose aussi des liens beaucoup plus complexes. Ici, nul besoin de se trouver «dans» une ressource. Le lien peut être indépendant du document de départ et du document visé. De plus, il permet de décrire une collection de ressources et des chemins entre ces ressources (chemins entre une ou plusieurs ressources). Pour cela, il propose de nouveaux attributs :

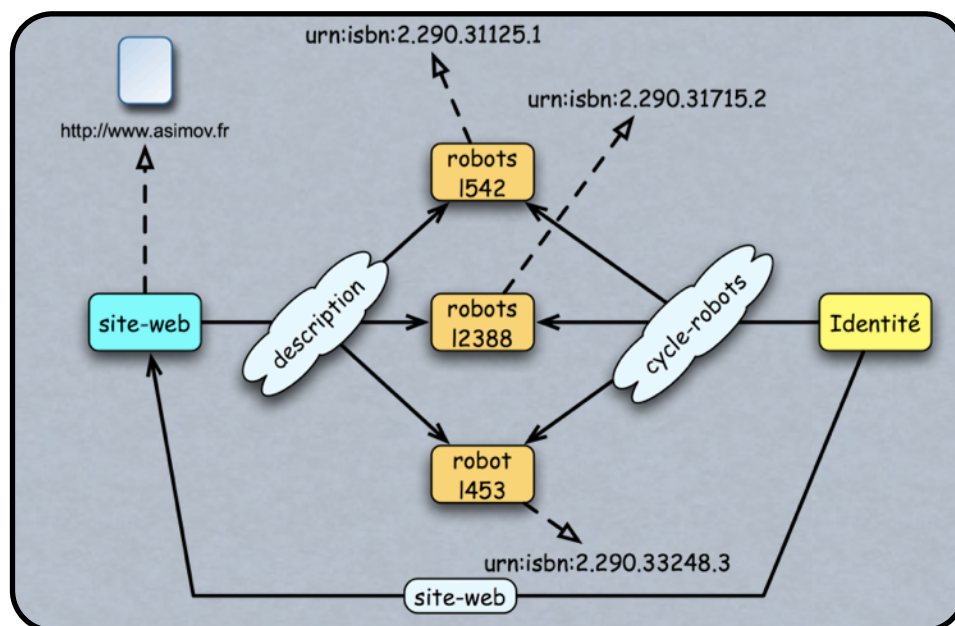
- «title» et «role» sont possibles pour tous ces types ;
- «type» prend alors «extended» sur l'élément englobant la description des liens et des ressources ;
- les ressources sont décrites dans l'élément principal avec les attributs suivants :
 - «type» vaut «ressource» (signifie une référence à une ressource locale) ou «locator» (signifie une référence à une ressource distante),
 - «href» donne l'URI de la ressource,
 - «label» sert à l'identification de la ressource, plusieurs ressources peuvent avoir le même label si elles sont prises ensembles dans le lien ;
- les liens sont aussi décrits dans l'élément principal avec les attributs suivants :
 - «type» vaut «arc» détermine une connexion entre deux ressources,
 - «from» et «to» détermine les extrémités de l'arc (orienté), respectivement l'origine et l'extrémité.

L'exemple ci-dessous présente un cas de liens étendus. On y retrouve :

- (1) des ressources locales ;
- (2) des ressources distantes sous forme d'URL ;
- (3) des ressources distantes sous forme de numéro ISBN (URI) ;
- (4) des liens entre les ressources.

<pre><?xml version="1.0" encoding="UTF-8"?> <auteur xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended"> <identité xlink:type="ressource" xlink:label="identite"> <nom>Asimov</nom><prenom>Isaac</prenom> <date-naissance>1920-01-02</date-naissance> <date-deces>1992-04-06</date-deces> <nationalite>Russe/Américain</nationalite> </identité></pre>	①
<pre> <site xlink:title="site d'un fan français" xlink:label="site-web" xlink:href="http://www.asimov.fr" xlink:type="locator"/></pre>	②
<pre> <livre annee="1964" reference="l542" editeur="J'ai lu" xlink:type="locator" xlink:href="urn:isbn:2.290.31125.1" xlink:label="robots"> <titre>Un défilé de robots</titre> </livre> <livre annee="1950" reference="l453" editeur="J'ai lu" xlink:type="locator" xlink:href="urn:isbn:2.290.34248.3" xlink:label="robots"><titre>I, robot</titre> </livre> <livre annee="2002" reference="l2388" editeur="J'ai lu" xlink:type="locator" xlink:href="urn:isbn:2.290.31715.2" xlink:label="robots"><titre>Le robot qui rêvait</titre> </livre></pre>	③
<pre> <cycle-robots xlink:type="arc" xlink:from="identite" xlink:to="robots" xlink:title="début du cycle des robots"/> <site-web xlink:type="arc" xlink:from="identite" xlink:to="site-web"/> <description xlink:type="arc" xlink:from="site-web" xlink:to="robots"/></pre>	④
<pre></auteur></pre>	

Graphiquement, cet exemple peut donner le schéma ci-dessous. Les flèches en pointillé indiquent les ressources externes. Les flèches pleines sont les liens XLink.



3. XPointer

En complément de XLink et utilisant XPath, il existe un standard de syntaxe non XML qui localise des noeuds, des points ou des régions dans des documents : XPointer². Utilisé conjointement avec XPath, XPointer permet de localiser des structures XML dans un document distant. Attaché à la fin des URI, après un "#", comme un identifiant de fragment HTML, il permet d'indiquer une partie spécifique d'un document XML, vue d'un autre document.

Par exemple «[http://www.mon-site.fr/mon_fic.xml](http://www.mon-site.fr/mon_fic.xml#xpointer(/html/head/title)» indique l'élément «title» de l'entête XHTML dans le document «<a href=)». Ce standard complète XLink au niveau de l'attribut «href» (voir section précédente). Dans le cadre de XPointer, XPath est étendu pour affiner la localisation des informations (référence par la position, points de référence dans les noeuds textes, etc.).

La syntaxe de base est la suivante : «en*xp+». Soit d'abord, éventuellement, des définitions d'espaces de nom («en»). Les espaces de noms sont décrits de la manière suivante : «xmlns(pref=URI)» où «pref» est le préfixe qui sera utilisé et «URI» décrit l'espace de noms. Puis une suite de références («xp»), la première dans l'ordre de déclaration retournant un résultat non vide est alors conservée. Les différents «xp» possibles sont :

- «xpointer(*chemins_xpath_étendu*)» ;
- «element(*position_arbre*)» : c'est une définition d'une séquence d'enfants à la place d'un chemin XPath qui est soit à partir de la racine ("/"), soit à partir d'un ID. Ensuite, les enfants sont notés par niveaux, puis, éventuellement, par positionnement dans le noeud texte (point ou région).

Voici quelques exemples de descriptions de contenu en XPointer :

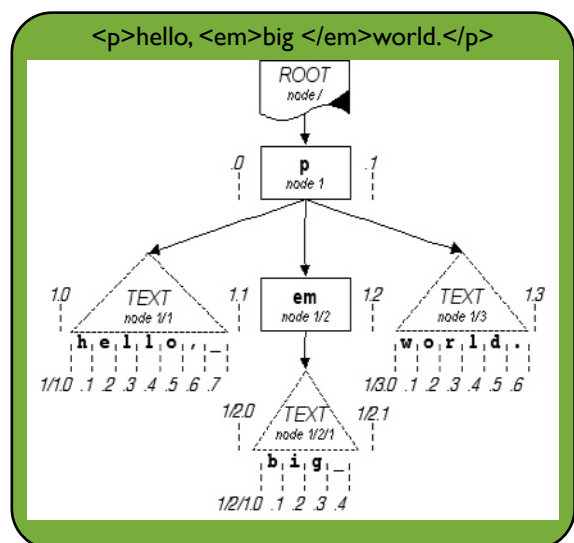
- «xmlns(m="http://.../.../MathML")xpointer(/m:math[2])»
- «element(/1/2)» retourne l'élément «em» sur le fragment de XML ci-dessous et est équivalente à la requête utilisant XPath suivante :
«xpointer(/child::*[position()=1]/child::*[position()=2])»

XPointer permet la localisation de parties XML (emplacements, lieux...) indépendamment des structures XML. Comme exemple d'utilisation, il peut servir à la localisation d'annotations après sélection (sur-lignage) par un utilisateur. Cela passe par la numérotation des noeuds et de points entre les noeuds et dans les textes. Par contre, il n'y a pas de localisation dans les espaces de noms et les attributs, car ils ne sont pas ordonnés.

La fonction permettant de repérer un emplacement dans un document XML est : «point(localisation)», qui pointe un emplacement dans le fichier XML. Dans l'exemple précédent, voici des exemples de points «①<p>②hello, b③ig ④world.</p>»

qui sont définis par :

- ① «point(.0)» ;
- ② «point(1.0)» ;
- ③ «point(1/2/1.1)» ;
- ④ «point(1.2)».



A partir des emplacements, il est possible de définir des intervalles («range») qui sont des zones XML entre deux «points» : un point de départ (ou «start-point») et un point d'arrivée (ou «end-point»). Ainsi, il est possible de trouver les fonctions suivantes :

- «range(XPath)» : retourne un ensemble d'emplacements (y compris les balises de début et de fin) ;
- «range-inside(XPath)» : ensemble de lieux sans les balises de début et fin ;
- «range-to(XPath)» : détermine le point de départ et le point d'arrivée de l'élément référencé pour chaque sous-arbre par rapport au noeud contextuel ;

² <http://www.w3.org/XML/Linking>, <http://www.w3.org/TR/xptr/>

- «string-range(*XPath*, *mot?*, *int?*, *int?*)» : retourne un «range» au niveau du texte, contenant le mot avec éventuellement un décalage et un nombre de caractères ;
- «start-point(*XPath*)» : détermine le point de départ d'un «range» ;
- «end-point(*XPath*)» : détermine le point d'arrivée d'un «range» ;
- «here()» : dans un document, permet de référencer le noeud contextuel (contenant le XPointer) ;
- «origin()» : dans le cadre de XLink, permet de référencer des éléments ayant même «label».

4. XInclude

XInclude³ est un standard XML pour inclure des fragments de fichiers (texte ou XML) dans un document XML. L'espace de noms associé à ce dialecte est «<http://www.w3.org/2001/XInclude>». Il permet d'introduire la notion de modularité étendue à un document XML, car il est possible prendre des morceaux de document (de les fusionner), même si un document n'est pas modifiable ! En termes de bases de données, il permet de mettre en place des bases de données réparties (ou distribuées). Pour déterminer les structures XML à récupérer, il s'appuie sur XPointer. Le schéma DTD de XInclude est ci-dessous.

Dans ce schéma :

- «href» est l'URI du document à inclure ;
- «xpointer» spécifie la ou les parties à inclure ;
- «parse» détermine la méthode d'inclusion (si le fichier est en XML, mais inclus en texte, les symboles sont échappés) ;
- «fallback» détermine le XML ou le texte à positionner en cas d'échec de l'inclusion.

```
<!ELEMENT xi:include (xi:fallback?)>
<!ATTLIST xi:include
  xmlns:xi CDATA #FIXED "http://www.w3.org/2001/XInclude"
  href CDATA #IMPLIED
  parse (xml|text) "xml"
  xpointer CDATA #IMPLIED
  encoding CDATA #IMPLIED
  accept CDATA #IMPLIED
  accept-language CDATA #IMPLIED
>

<!ELEMENT xi:fallback ANY>
<!ATTLIST xi:fallback
  xmlns:xi CDATA #FIXED "http://www.w3.org/2001/XInclude"
>
```

Soit un document ci-dessous nommé «ex_asimov.xml».

```
<?xml version="1.0" encoding="UTF-8"?>
<auteur>
  <identité>
    <nom>Asimov</nom><prenom>Isaac</prenom>
    <date-naissance>1920-01-02</date-naissance>
    <date-deces>1992-04-06</date-deces>
    <nationalite>Russe/Américain</nationalite>
  </identité>
  <livre annee="1964" editeur="J'ai lu" reference="l542">
    <titre>Un défilé de robots</titre>
  </livre>
  <livre annee="1950" editeur="J'ai lu" reference="l453">
    <titre>I, robot</titre>
  </livre>
  <livre annee="2002" editeur="J'ai lu" reference="l2388">
    <titre>Le robot qui rêvait</titre>
  </livre>
</auteur>
```

Si un document XML, basé sur XInclude est construit comme celui ci-dessous.

³ <http://www.w3.org/TR/xinclude/>

```
<?xml version="1.0" encoding="UTF-8"?>
<bibliotheque xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="ex_asimov.xml" parse="xml" xpointer="xpointer(//livre)"/>
</>
```

Alors le résultat de l'application de XInclude, par exemple avec la commande Linux⁴

«\$ xmllint --xinclude --noent --encode iso-8859-1 ex_xinclude.xml» donnera comme résultat le document ci-dessous.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<bibliotheque xmlns:xi="http://www.w3.org/2001/XInclude">
  <livre annee="1964" editeur="J'ai lu" reference="l542">
    <titre>Un défilé de robots</titre>
  </livre><livre annee="1950" editeur="J'ai lu" reference="l453">
    <titre>I, robot</titre>
  </livre><livre annee="2002" editeur="J'ai lu" reference="l2388">
    <titre>Le robot qui rêvait</titre>
  </livre>
</bibliotheque>
```

Exercices

Exercice 1. Plan de livre

Soit la page du plan du livre "An Introduction to XML and Web Technologies" de Anders Moller et Michael Schwartzbach (Addison Wesley) ci-contre.

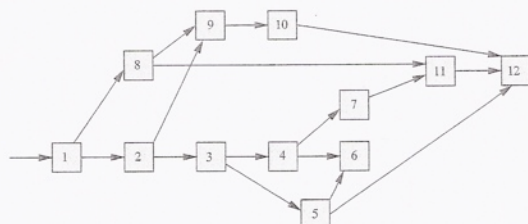
Dans ce livre, la lecture des différents chapitres suit le graphe ci-dessous.

Question. Construire un document XML décrivant ce graphe en utilisant XLink. Proposer plusieurs solutions qui n'utilisent pas nécessairement XLink.

BRIEF CONTENTS

	Foreword	xvii
	Preface	xix
PART I	XML TECHNOLOGIES	1
1	HTML and Web Pages	3
2	XML Documents	32
3	Navigating XML Trees with XPath	58
4	Schema Languages	92
5	Transforming XML Documents with XSLT	188
6	Querying XML Documents with XQuery	240
7	XML Programming	285
PART II	WEB TECHNOLOGIES	341
8	The HTTP Protocol	343
9	Programming Web Applications with Servlets	390
10	Programming Web Applications with JSP	429
11	Web Services	466
12	A Complete Application	496
	Bibliography	529
	Index	535

The chapter dependencies may be illustrated by the following diagram:



BRIEF CONTENTS

v

⁴ «xmllint» est une application Linux en ligne de commande permettant d'analyser des documents XML. <http://xmlsoft.org/xmllint.html>

Exercice 2. Agenda partagé (version 2)

Une entreprise propose à chacun de ses collaborateurs un agenda personnel disponible sur l'intranet. La DTD de cet agenda est exposée ci-dessous.

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

<!ENTITY % contenu_journée " (rdv)*" >

<!ELEMENT agenda (année)* >
<!-- ATTLIST agenda ref ID #REQUIRED -->

<!-- ELEMENT année (semaine)* -->
<!-- ATTLIST année no NMTOKEN #REQUIRED -->

<!-- ELEMENT semaine (lundi,mardi,mercredi,jeudi,vendredi,samedi?,dimanche?) -->
<!-- ATTLIST semaine no_semaine NMTOKEN #REQUIRED -->

<!-- ELEMENT lundi %contenu_journée; -->
<!-- ELEMENT mardi %contenu_journée; -->
<!-- ELEMENT mercredi %contenu_journée; -->
<!-- ELEMENT jeudi %contenu_journée; -->
<!-- ELEMENT vendredi %contenu_journée; -->
<!-- ELEMENT samedi %contenu_journée; -->
<!-- ELEMENT dimanche %contenu_journée; -->

<!-- ELEMENT rdv (destination,répétition?,commentaire?) -->
<!-- ATTLIST rdv
no ID #REQUIRED propriétaire IDREF #REQUIRED
horaire-début CDATA #REQUIRED horaire-fin CDATA #REQUIRED
titre CDATA #REQUIRED importance (1|2|3|4) '1' -->

<!-- ELEMENT destination EMPTY -->
<!-- ATTLIST destination
statut (privé|public|liste) 'public'
destinataire NMTOKENS 'anonyme' -->

<!-- ELEMENT répétition EMPTY -->
<!-- ATTLIST répétition
répétition (journalière|hebdomadaire|mensuelle|annuelle) #REQUIRED
début-répétition CDATA #REQUIRED
fin-répétition CDATA #REQUIRED -->

<!-- ELEMENT commentaire (#PCDATA) -->
```

De plus, les différents collaborateurs sont recensés dans un document lié à l'authentification des personnes, dont la DTD est présentée ci-contre.

Cette entreprise se propose de construire un document central qui regroupe l'ensemble des rendez-vous des collaborateurs pour une semaine donnée. Seuls les rendez-vous qui ne sont pas privés sont présentés.

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

<!-- ELEMENT liste-utilisateurs (utilisateur)* -->

<!-- ELEMENT utilisateur EMPTY -->
<!-- ATTLIST utilisateur
login ID #REQUIRED
nom CDATA #REQUIRED
prénom CDATA #REQUIRED -->
```

Question 1. Écrire une DTD permettant de décrire un tel document. Celle-ci doit s'appuyer sur la DTD des agendas personnels.

Question 2. Écrire un document XML/XInclude permettant de construire l'agenda centralisé, sans reprendre directement les informations et dont le résultat respecte la DTD de la question précédente. L'année et la semaine seront "en dur" dans le document.

Question 3. Proposer une solution utilisant XLink. Discuter de l'utilisation de XLink à la place de XInclude. Quelles sont les différences ? Quelle est l'impact sur l'utilisation du document ? etc.



Cette création est mise à disposition selon le Contrat Paternité-Pas d'Utilisation Commerciale-Pas de Modification 2.0 France disponible en ligne <http://creativecommons.org/licenses/by-nc-nd/2.0/fr/> ou par courrier postal à Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.