Introduction à XML [DVIR3]

Informatique et systèmes finalité Réseaux & télécommunication

2016-2017

XML

1. Introduction
2. Langage XML
3. Document XML
4. DTD : Déclaration de la structure du document
5. XML Schema
6. Adressage de fragments xml : Xpath
7. Contenu et présentation : CSS et XSL
8. Langage de transformation XSLT
9. Langage de formatage XSL-FO
10. PHP - XML

XML

1. Introduction
2. Langage XML
3. Document xml
4. DTD : Déclaration de la structure du document
5. XML Schema
6. **Adressage de fragments xml : Xpath**
7. Contenu et présentation : CSS et XSL
8. Langage de transformation XSLT
9. Langage de formatage XSL-FO
10. PHP - XML

Sélectionner des Fragments XML

- XPath est utilisé par XML Schéma pour créer des clés et des références

- XSL pour transformer des documents XML

XPath

- Fondé sur une représentation arborescente (DOM) du document XML
- Objectif : référencer des nœuds (éléments, attributs, commentaires, ...) dans un document XML afin d'être utilisés par le langage XSLT.

La norme de référence

- W3C recommandation :

  http://www.w3.org/TR/xpath

- Xpath modélise un document XML sous la forme d'un arbre composé de nœuds de plusieurs types

Types de nœuds

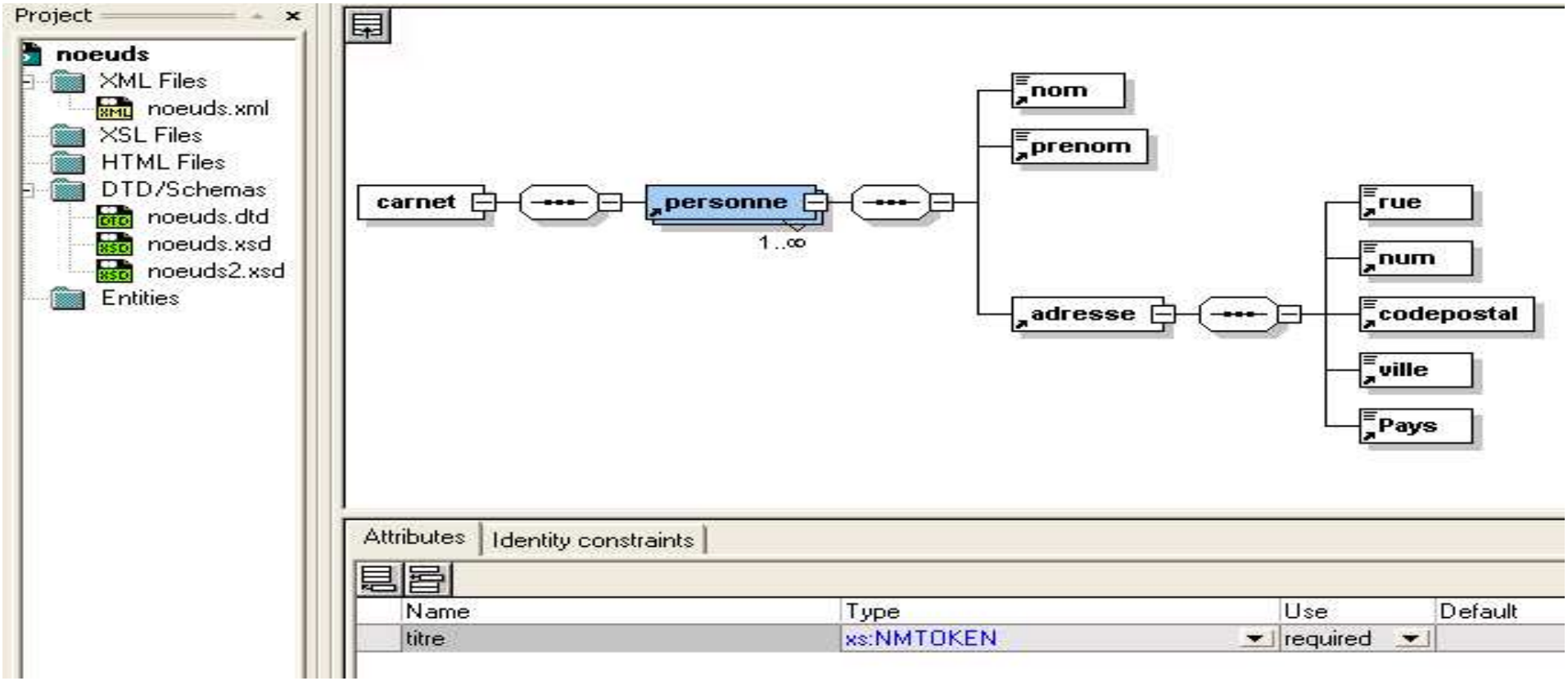
- *Racine* : / représente l'entièreté du document
- *Element* : pour chaque élément du document
- *Attribut* : pour chaque attribut d'un élément
- *Texte*: pour chaque donnée textuelle ou la valeur d'un attribut
- *Commentaire* : pour chaque commentaire dans le document
- *Instruction de traitement* : dans le document
- *Namespace* : pour chaque namespace utilisé dans le document

Objets renvoyés par les expressions XPath

Résultat après évaluation :

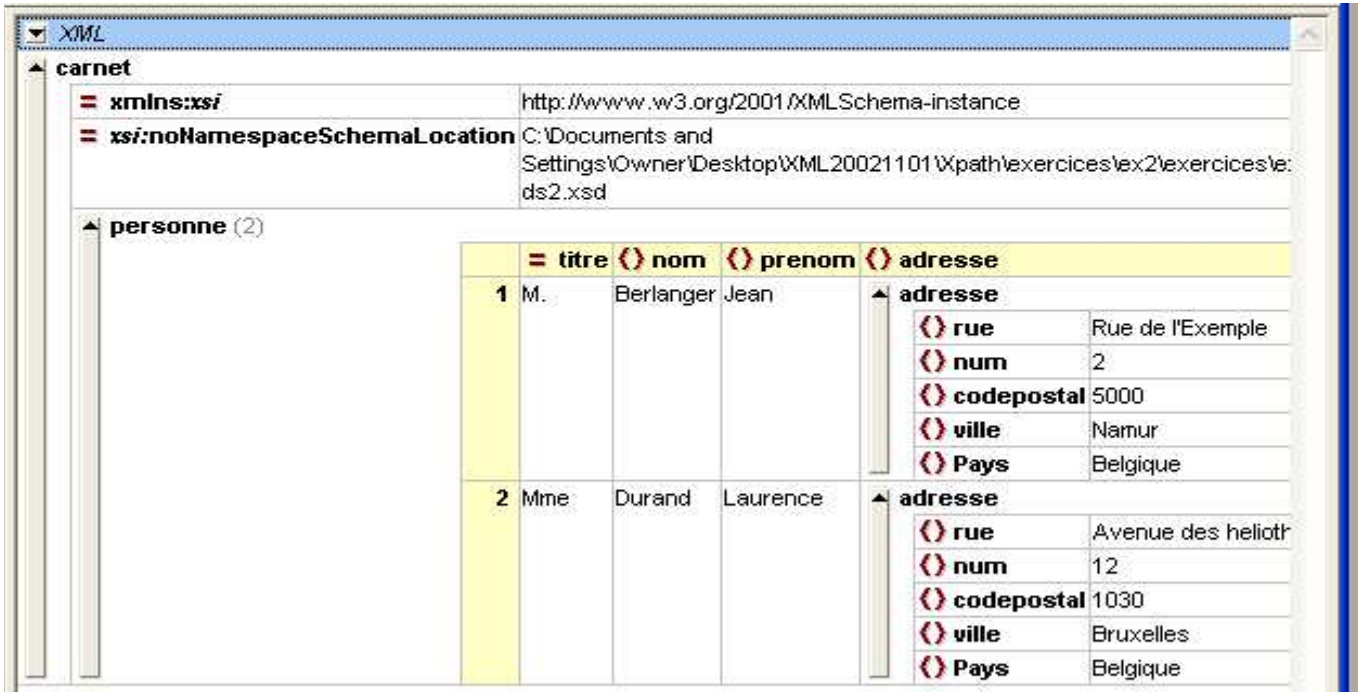
- Un ensemble de noeuds sans doublons
- Un booléen (*true* ou *false*)
- Un nombre en virgule flottante
- Une chaîne de caractères

Exemple


Project
noeuds
XML Files
noeuds.xml
XSL Files
HTML Files
DTD/Schemas
noeuds.dtd
noeuds.xsd
noeuds2.xsd
Entities
carnet
personne
1..∞
nom
prenom
adresse
rue
num
codepostal
ville
Pays
Attributes
Identity constraints
Name
Type
Use
Default
titre
xs:NMTOKEN
required

Exemple

Titre	nom	prénom	rue	num	codepostal	ville	Pays
M.	Berlanger	Jean	Rue de l'Exemple	2	5000	Namur	Belgique
Mme	Durand	Laurence	Avenue des heliothropes	12	1030	Bruxelles	Belgique

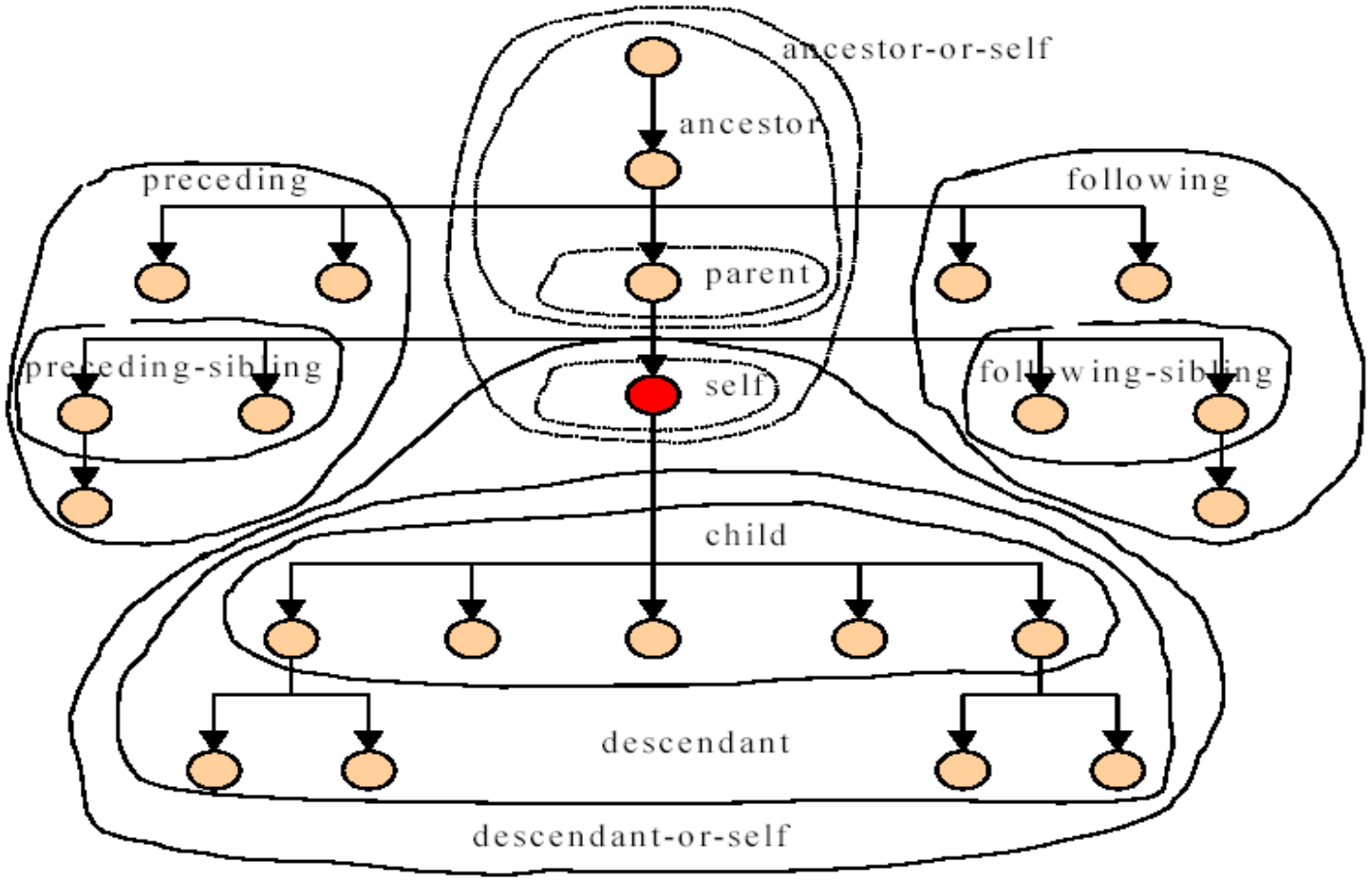


Contexte d'Évaluation

Un contexte d'évaluation dans le calcul d'une expression est défini par :

- un nœud (*context node*) dans cette liste, ex.: <personne>
- une taille de contexte (2) et la position du contexte (1 ou 2),
- un ensemble de variables (ensemble de paires nom = valeur),
- un ensemble de fonctions (fonctions de XPath et XPointer),
- un ensemble de déclarations de domaines de nommage.

Axes de déplacement



Source : http://homepage.ensica.fr/~ernesto/xml/coursXML_3.pdf

Chemin de positionnement

- Un *chemin d'accès* (de positionnement) est une séquence d'étapes de positionnement (location step) par rapport à un ensemble de nœuds données (contexte) :
  `axe::test/axe::test/.../axe::test`

  ex.: ancestor::personne/adresse/ville[parent::adresse/child::codepostal/text()='1030']

  ancestor::personne/child::adresse/child::ville[../codepostal/text()='1030']

  ancestor::personne/descendant::ville[parent::adresse/codepostal/text()='1030']

  condition=prédicat

1. L'axe sélectionne un ensemble de nœuds par rapport à leur position dans un document (arbre) ou relative à un contexte.

   ex.: ../codepostal

2. Le test est évalué pour chaque nœud dans la sélection

   ex.: sélection de tous les nœuds <personne> qui sont enfants du nœud racine <carnet>
   /carnet/personne

   ex.: sélection de tous les nœuds <personne> dont l'adresse est a Schaerbeek :
   carnet/personne[adresse/codepostal="1030"]

3. Chaque étape crée un nouveau contexte pour l'étape suivante

Chemins absolus

Le contexte d'évaluation est choisi d'une manière absolue

- /: sélectionne toute la ressource (avec le prologue)
- *id(val)* : élément avec `ID=val`
- *here*() : élément qui contient le XPointer
- *origin*() : utilisation avec liens externes XLink

Chemins relatifs

- Si le contexte d'évaluation est, par exemple, <personne>, on peut écrire pour accéder au nom de la ville :

  adresse/ville ou *child::*adresse/ville

  (*child::* est l'axe de recherche par défaut)

- Si le contexte d'évaluation est, par exemple, <adresse> du 1er nœud <personne>, on peut écrire pour accéder au nom de cette personne :

  *parent::*nom ou *ancestor::*personne/nom

  (*parent::* axe de recherche qui sélectionne le nœud parent)

  (*ancestor::* axe de recherche qui sélectionne tous les nœuds ancêtres)

XPath: Exemples

- La racine du document:
  /

- Tous les fils de type film de la racine du document XML :
  /child::film

- Tous les fils de type film (d'un contexte donné) :
  child::film

- Tous les (premiers) fils de type film qui suivent un fils de type seance :
  child::seance/following-sibling::film[position()=1]

Axes de recherche

(élément, PI, commentaire, texte, CDATA):

Reverse axe :

ancestor, ancestor-or-self, parent, preceding-sibling, preceding , self

Forward axe :

child, descendant, descendant-or-self, following-sibling,

following, attribute, namespace, self

Dans un élément XML, l'ensemble `{self::*, preceding::*, following::*, descendant::*, ancestor::*}` forme une partition de tous les éléments du document XML.

Nœuds test 1/2

- ***** : sélectionne tous les nœuds éléments dans l'axe de recherche

  child::*

- Un ***nom*** : sélectionne tous les nœuds appelés *nom* dans l'axe de recherche

  child::codepostal descendant::ville attribute::titre

- **text()** : sélectionne tous les nœuds textes dans l'axe de recherche

  descendant::text()

Nœuds test 2/2

- **comment()** : sélectionne tous les nœuds commentaires situés dans l'axe de recherche

- **processing-instruction(** [**nom**] **)** : sélectionne tous les nœuds "instructions de traitement" situés dans l'axe de recherche [et dont le nom est mentionné entre parenthèses]

- **node()** : sélectionne tous les nœuds situés dans l'axe de recherche, quel que soit leur type (sauf attributs et namespaces)
  child::node()

Prédicats

- Filtrent l'ensemble des nœuds sélectionnés par l'axe de recherche et le nœud test.

- Pour chaque nœud filtré par un prédicat, le prédicat est évalué comme une expression Xpath (retournant *true* ou *false*) pour laquelle :

  - le nœud contexte est le nœud en question
  - La taille de contexte est le nbre de nœuds de l'ensemble initial
  - La position du contexte est la position de ce nœud parmi cet ensemble

Exemple : prédicats

- Pour sélectionner les localités dont le code postal est '1030' :

1. personne/adresse [child::codepostal='1030'] /ville
2. personne/adresse/ville[parent::adresse/codepostal='1030']
3. personne/adresse/ville[../codepostal='1030']
4. personne/adresse/ville[ancestor::personne/descendant::codepostal='1030']

XPath : Tests

Les prédicats font appel à des tests sur

- les noms des éléments et des attributs
- le type du nœud (texte, comment, PI, *)
- la position : `[position()=N]`
- le nombre d'occurrences : `[count(child::acteur) > 1]`
- la structure locale : chemins imbriqués avec connecteurs logiques (qualifiers)

XPath : Exemples

- Film : tous les éléments `film`
- * : tous les éléments (pour l'axe de recherche child)
- @titre : tous les attributs `titre`
- @*: tous les attributs
- personne[position( ) =2]/adresse/rue
- acteur|actrice : tous les éléments `acteur` et `actrice`
- text() : tous les éléments qui contiennent du texte
- id("bart") : l'élément avec l'identificateur *bart* (chemin absolu)

XPath : Exemples

- `film[1]`: le premier élément film
- `*[position()=1 and self::film]` : l'élément film s'il est le premier élément
- `seance[position()>1]` : tous les éléments `seance` sauf le premier
- `film[@titre="Shining"]` : les éléments film avec l'attribut `@titre` égal à "Shining"

XPath : Syntaxe simplifiée

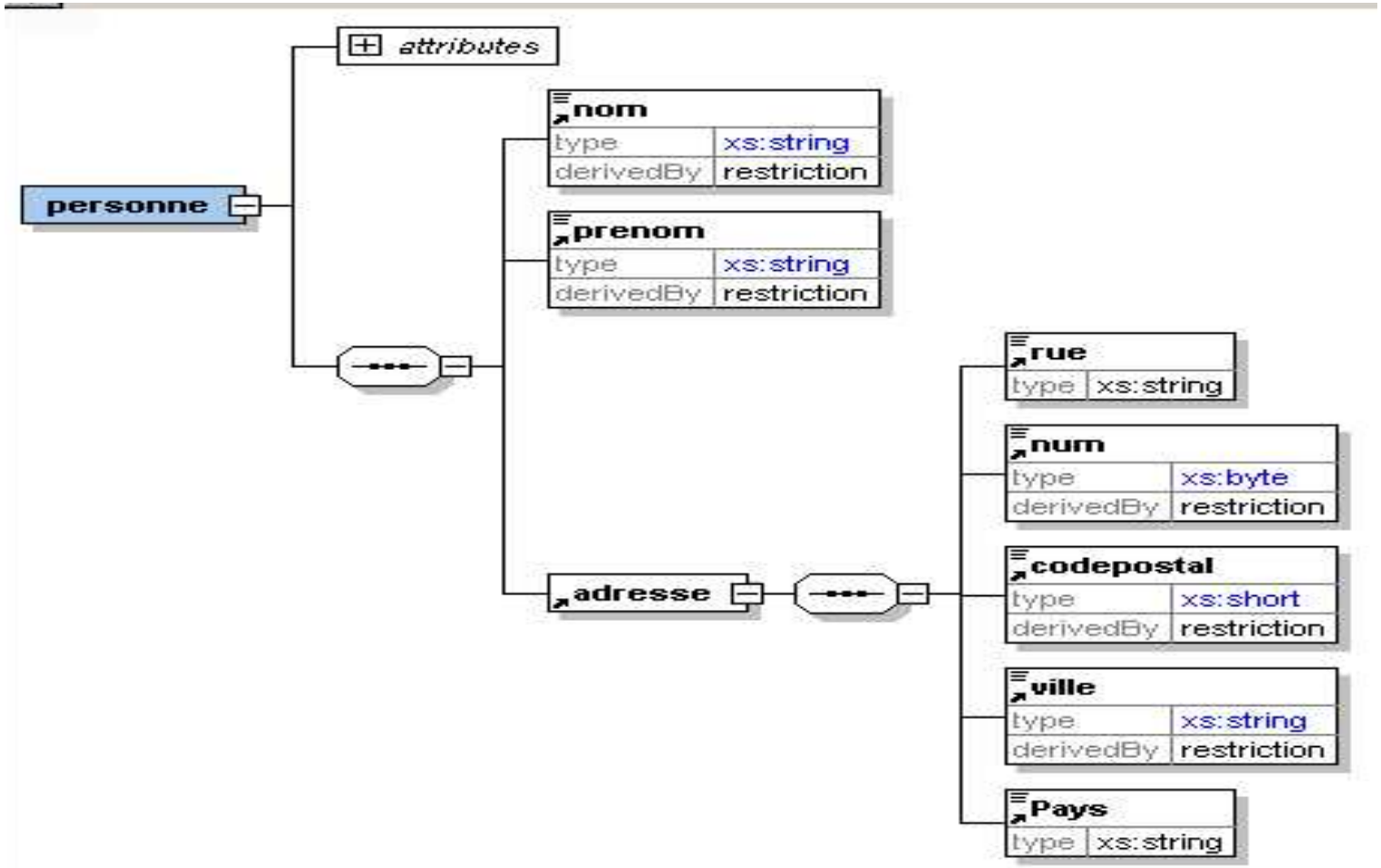
XPATH

syntaxe simplifiée

/child::film/child::acteur	/film/acteur
/child::cinema/descendant::acteur	/cinema//acteur
/descendant-or-self::node( )/	//
/descendant-or-self::node( )/*	//*
/descendant-or-self::node( )/film[@année=’2003’]	//film[@année=’2003’]
/attribute::nom	/@nom
self::node()	.
parent::node()	..
[position()=nombre]	[nombre]

Exercice

a) Construire le schéma xml suivant :



Evaluate Xpath

b) Construire un document xml associé au schéma

c) Evaluer les expressions XPATH suivantes :

/

/carnet/personne

//personne

count(child::personne)>=0

/carnet *puis* /personne[@titre='M.']

//personne[position()=1]

//personne[1]/nom

//personne[1]/nom/text()

puis à partir de ce contexte //parent::node()/child::adresse/child::ville

//personne/../personne[2]/nom

Les expressions Xpath

=

combinaison d'opérateurs et d’opérandes

opérandes

- Chaînes de caractères
- Nombres (constantes numériques)
- Appels de fonctions
- Autres expressions entre ( )
- Ensembles de nœuds
- Références à des variables : $*cpt* (cf.xslt)

opérateurs

- Ensemble de nœuds :
  '|' réalise l'union de nœuds
  nom|*prénom*
- Op. booléens : 'or' 'and'
- Op. de comparaison : '=' '!=' '>' '<' '<=' '>='
- Op. arithmétiques : '+' '-' '*' 'div' 'mod'

Fonctions

- *number* **last**() : retourne la taille du contexte
- *number* **position**() : retourne la position du contexte
- *number* **count**(*nodeset*) : retourne le nombre de nœuds se trouvant dans l'ensemble donné en paramamètre
- *nodeset* **id**(*name*) : retourne un ensemble de nœuds, composé d'un seul nœud donné par l'identificateur *name* et dont le type déclaré doit être un ID (cf. DTD)

Fonctions

- *string* **local-name**(*nodeset?*) : retourne le nom local (sans namespace) du 1er nœud de l'ensemble des nœuds donné en paramètre ou, si le paramètre est omis, du nœud contexte
- *string* **name**(*nodeset?*) : retourne le nom (avec namespace) du 1er nœud de l'ensemble des nœuds donné en paramètre ou, si le paramètre est omis, du nœud contexte
- *string* **namespace-uri**(*nodeset?*) : retourne l'URI du namespace du 1er nœud de l'ensemble des nœuds donné en paramètre ou, si le paramètre est omis, du nœud contexte

  Retourne une chaîne vide s'il n'y a pas de namespace.

Fonctions

- *string* **string**(*object?*) : retourne une chaîne de caractères qui, après conversion, représente l'objet donné en paramètre ou, si celui-ci est absent, le nœud contexte
- *number* **number**(*object?*) : retourne un nombre qui, après conversion, représente l'objet donné en paramètre ou, si celui-ci est absent, le nœud contexte

  NaN si représentation non numérique
- *boolean* **boolean**(*object?*) : retourne un booléen qui, après conversion, représente l'objet donné en paramètre ou, si celui-ci est absent, le nœud contexte

  false si l'ensemble des nœuds est vide ou si le nbre est=0

Fonctions

- *string* **concat**(*string1,string2,…*) : retourne la concaténation des chaines données en paramètre
- *boolean* **starts-with**(*string1,string2)* : retourne t*rue* si *string1* commence par *string2*
- *boolean* **contains**(*string1,string2*) : retourne *true* si *string1* contient *string2*
- *string* **substring-before**(*string1,string2*) : retourne la chaîne de car. qui se trouve avant *string2* dans *string1*
- *string* **substring-after**(*string1,string2)* : retourne la chaîne de car. qui se trouve après *string2* dans *string1*

Fonctions

- *string* **substring**(*string, i ,n?)* : retourne la chaîne de caractères qui se trouve dans *string* à partir du *i* ème, éventuellement limitée à *n* caractères
- *number* **string-length**(*string)* : retourne le nombre de caractères de la chaîne de caractères
- *string* **normalize-space**(*string?)* : retourne une chaîne de caractères égale à *string* où les "blancs" ont été normalisés (groupe d'espaces, tabulation, ligne blanche ~ un seul espace) ou conversion du nœud contexte si le paramètre est absent
- *string* **translate**(*string1,string2,string3)* : retourne une chaîne de caractères égale à *string1*, mais où tous les caractères spécifiés par *string2* sont remplacés par les caractères correspondant de *string3*

Fonctions

- *boolean* **not**(*boolean*) : inverse un booléen
- *boolean* **true**() : retourne *true*
- *boolean* **false**() : retourne *false*
- *boolean* **lang**(*string?*) : retourne *true* si *xml:lang* est spécifié pour le nœud contexte

  Si le paramètre est présent,
  renvoie *true* si *xml:lang* est égal à la langue donnée par le paramètre

Fonctions

- *number* **sum**(*nodeset*) : convertit les nœuds de l'ensemble en numérique et renvoie la somme
- *number* **floor**(*number*) : renvoie le plus grand entier <= au nombre donné
- *number* **ceiling**(*number*) : renvoie le plus petit entier >= au nombre donné
- *number* **round**(*number*) : renvoie la valeur arrondie à l'entier le plus proche du nombre donné

Exercice

Dans le projet relatif aux cours organisés,
rechercher à l'aide d'expression Xpath et de fonctions :

- Les « noms » (des profs, des ue,…)
- Le nombre total d'heures de toutes les AA données au 3ème quadri
- Le nombre de cours donnés par le prof « xyz »
- Les ids des AA des UE avec une seule AA
- Les ids des AA des UE avec maximum 2 AA
- Les ids des UE avec maximum 2 AA
- Les AA dont l'id commence par "R"
- les UE dont la dernière lettre de l'id ne correspond pas au quadrimestre

Exercice

Certains plug-in peuvent être utiles tels

- Les utilitaires pour Netbeans : XPathUtil ou XPath Evaluator
- Pour les autres expressions, l'add-on XPath Checker de Firefox.