

# Les espaces de nommage XML

- Qu'est-ce ?
- Nature des espaces de nommage
- Objectifs
- Principes
- Déclaration, terminologie
- Portée des déclarations
- Espace de nommage par défaut
- Cas des attributs
- Surcharge des déclarations
- Espaces de nommages spéciaux
- URIs et préfixes
- Espaces de nommages et DTD
- Espaces de nommages dans le modèle de données
- Espaces de nommages dans les contenus
- Quelques problèmes

- Exemple avec Java :

Prenons la classe `Timer`

On en a 4

Comment distinguer `Timer` de `Timer` de `Timer` de `Timer` ?

En qualifiant le nom de la classe :

```
java.util.Timer  
javax.management.timer.Timer  
javax.swing.Timer  
org.acme.foo.Timer
```

Ces 4 classes peuvent cohabiter dans le même programme, mais il faudra les distinguer par leur nom qualifié, sauf une, qui pourra être spécifiée dans une directive d'import

Chacune de ces classes `Timer` est dans son espace de nom qui la distingue des autres

- Exemple de répertoire d'espace de noms :

Voir <http://ns.inria.org>

Objectif commun :

- Donner un nom universel à l'espace de nom

Java : basé sur les noms de domaine, sauf pour `java.XXX` et `javax.XXX`

`org.w3c.dom`

`org.inria.ns.reflex`

Identification d'un document : basé sur des URNs

`urn:oasis:names:tc:entity:xmlns:xml:catalog`

`urn:isbn:2-84177-230-6`

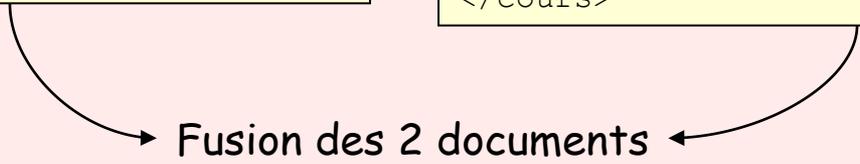
Référence d'un paramètre : basé sur des URLs

`http://xml.org/sax/features/validation`

XML : les espaces de noms sont basés sur des URIs

```
<?xml version="1.0"?>
<auteur>
  <nom>Poulard</nom>
  <prénom>Philippe</prénom>
  <titre>Baron</titre>
</auteur>
```

```
<?xml version="1.0"?>
<cours>
  <titre>Fondamentaux XML</titre>
  <contenu>
    .../...
  </contenu>
</cours>
```



Fusion des 2 documents



```
<?xml version="1.0"?>
<cours>
  <titre>Fondamentaux XML</titre>
  <auteur>
    <nom>Poulard</nom>
    <prénom>Philippe</prénom>
    <titre>Baron</titre>
  </auteur>
  <contenu>
    .../...
  </contenu>
</cours>
```

Confusion sur le sens de l'élément titre

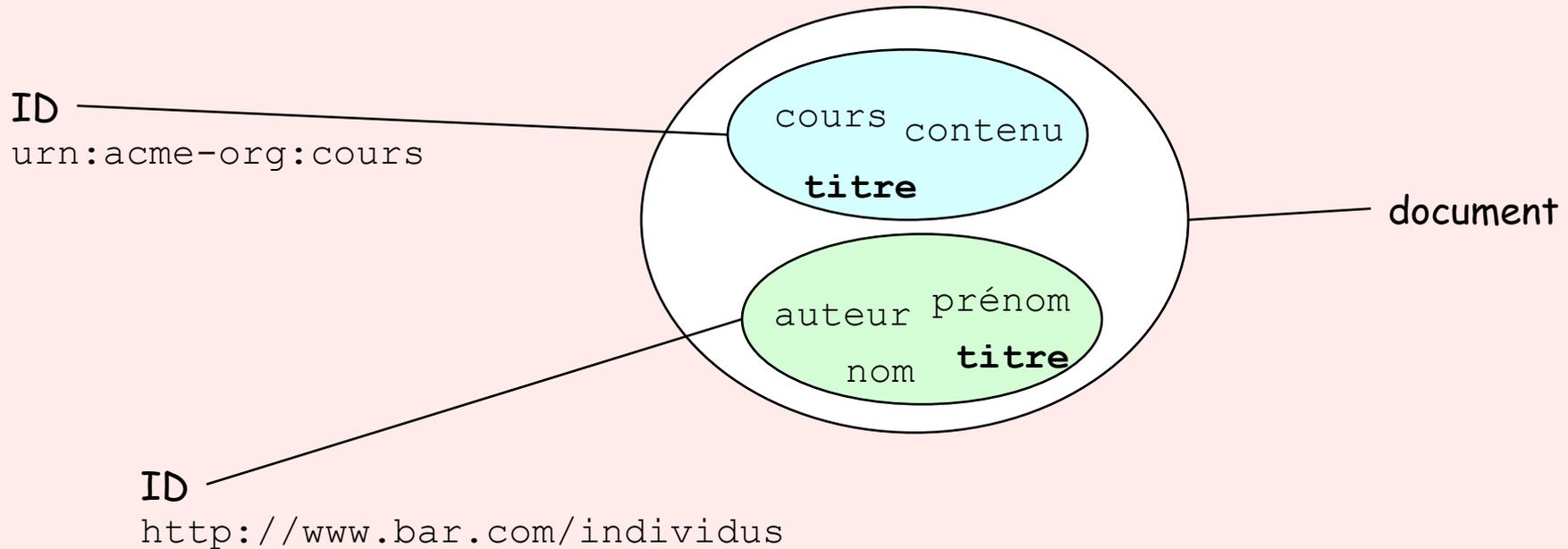
Si vous avez la maîtrise des DTD que vous utilisez, vous pouvez éviter cette confusion, mais si vous utilisez des DTD écrites par d'autres, vous y serez confronté

Objectifs des espaces de nom :

- **distinguer** les éléments et les attributs de différentes applications XML qui ont le même nom
- **grouper** les éléments et les attributs d'une même application XML pour que les logiciels puissent les reconnaître

Ambiguïté → assignation des éléments et attributs à une **clé universelle**  
(URI=ID universel)

Les URI séparent les éléments et les attributs en 2 ensembles disjoints



```
<?xml version="1.0"?>
<{urn:acme-org:cours}cours>
  <{urn:acme-org:cours}titre>Fondamentaux XML</{urn:acme-org:cours}titre>
  <{http://www.bar.com/individus}auteur>
    <{http://www.bar.com/individus}nom>Poulard</{http://www.bar.com/individus}nom>
    <{http://www.bar.com/individus}prénom>Philippe</{http://www.bar.com/individus}pré<
    <{http://www.bar.com/individus}titre>Baron</{http://www.bar.com/individus}titre>
  </{http://www.bar.com/individus}auteur>
  .../...
</{urn:acme-org:cours}cours>
```



- Un peu lourd.
- Syntactiquement incorrect.
- Mais l'idée est là.

Pour éviter une telle écriture, on a recours à une indirection :

→ On associe un alias à chaque URI d'espace de nom... {

- cours** → urn:acme-org:cours
- personne** → http://www.bar.com/individus

→ On les utilise en tant que **préfixe** :

```
<cours:cours>
  <cours:titre>Fondamentaux XML</cours:titre>
  <personne:auteur>
    <personne:nom>Poulard</personne:nom>
    <personne:prénom>Philippe</personne:préno<
    <personne:titre>Baron</personne:titre>
  </personne:auteur>
  .../...
</cours:cours>
```

Le "vrai" nom (nom expansé) est un couple : •http://www.bar.com/individus •auteur

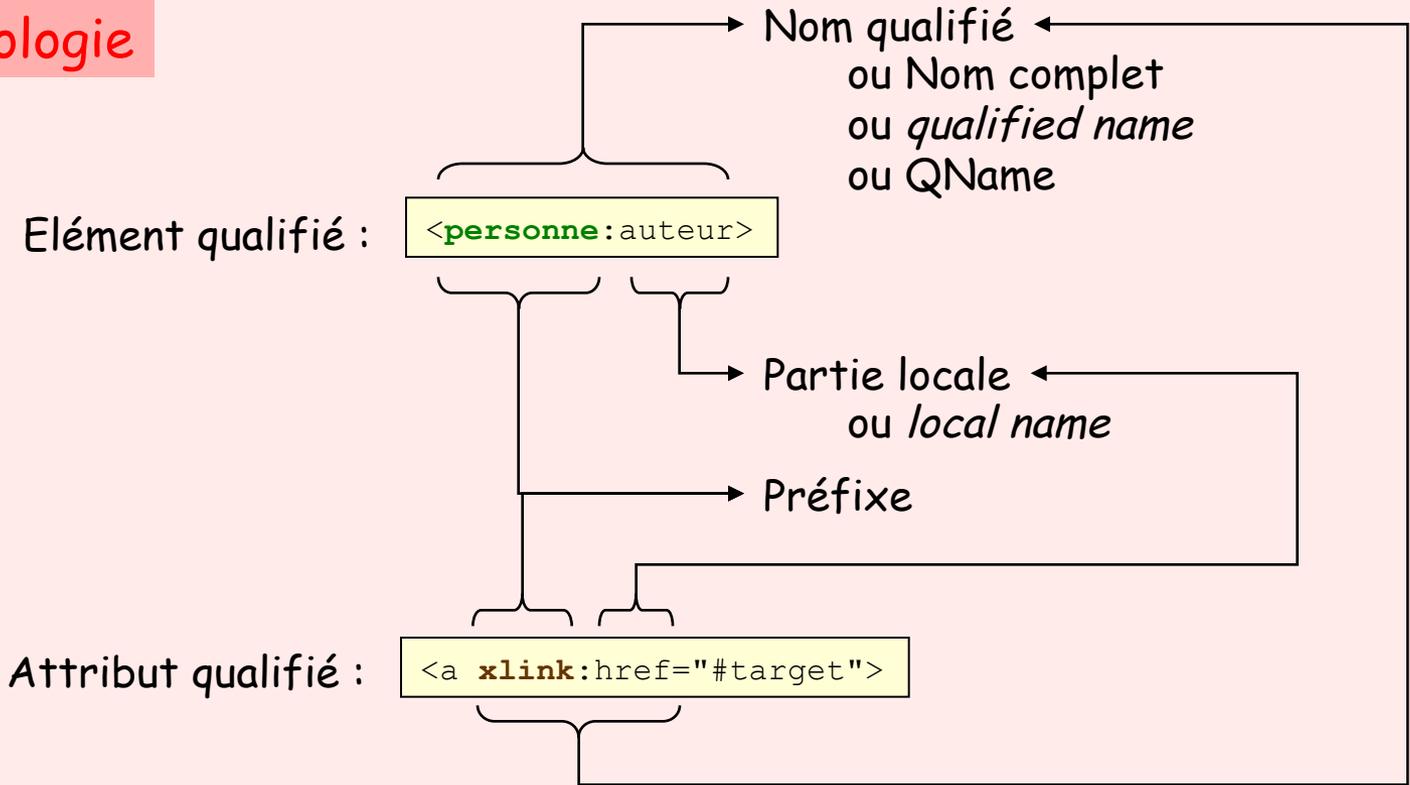
Pour référer un **nom expansé**, la notation {http://www.bar.com/individus}auteur est souvent utilisée, mais n'apparaît jamais dans un document XML.

Ils s'appliquent dès l'élément hôte de la déclaration

Ils se déclarent dans un élément, avec le préfixe spécial `xmlns`

```
<cours:cours
  xmlns:cours="urn:acme-org:cours"
  xmlns:personne="http://www.bar.com/individus">
```

## Terminologie



URI + local name = nom universel (nom expansé)

```

<?xml version="1.0"?>
<cours:cours xmlns:cours="urn:acme-org:cours">
  <cours:titre>Fondamentaux XML</cours:titre>
  <personne:auteur xmlns:personne="http://www.bar.com/individus">
    <personne:nom>Poulard</personne:nom>
    <personne:prénom>Philippe</personne:prénom>
    <personne:titre>Baron</personne:titre>
  </personne:auteur>
  <cours:contenu>
    .../...
  </cours:contenu>
</cours:cours>

```

Portée de `http://www.bar.com/individus`

Portée de `urn:acme-org:cours`

```

<?xml version="1.0"?>
<cours:cours xmlns:cours="urn:acme-org:cours"
  xmlns:personne="http://www.bar.com/individus">
  <cours:titre>Fondamentaux XML</cours:titre>
  <personne:auteur>
    <personne:nom>Poulard</personne:nom>
    <personne:prénom>Philippe</personne:prénom>
    <personne:titre>Baron</personne:titre>
  </personne:auteur>
  <cours:contenu>
    .../...
  </cours:contenu>
</cours:cours>

```

```
<?xml version="1.0"?>
<svg width="18cm" height="6cm" viewBox="0 0 200 120"
      xmlns:svg="http://www.w3.org/2000/svg">
  <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg>
```

Tous les éléments sont dans l'espace de nommage `svg`

Il est possible de ne pas spécifier de préfixe dans la déclaration

Définition d'un espace de nommage par défaut

```
<?xml version="1.0"?>
<svg width="18cm" height="6cm" viewBox="0 0 200 120"
      xmlns="http://www.w3.org/2000/svg">
  <rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg>
```

Tous les éléments non préfixés sont dans l'espace de nommage par défaut



Mais pas les attributs !

→ Ces 2 documents sont équivalents

L'espace de nommage par défaut ne s'applique pas aux attributs sans préfixe

```
<?xml version="1.0"?>
<svg:svg width="18cm" height="6cm" viewBox="0 0 200 120"
  xmlns:svg="http://www.w3.org/2000/svg">
  <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg:svg>
```

Ces attributs ne sont pas dans un espace de noms

Les attributs sans préfixes "appartiennent" à leur élément hôte

```
<?xml version="1.0"?>
<svg width="18cm" height="6cm" viewBox="0 0 200 120"
  xmlns="http://www.w3.org/2000/svg">
  <rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg>
```

Ces attributs ne sont toujours pas dans un espace de noms

Les applications :

- ne doivent pas tenter d'associer les attributs non préfixés à l'espace de nommage XML par défaut
- doivent associer les attributs non préfixés à leur élément-type (DTD)

Il est donc inutile de préfixer les attributs pour qu'ils soient dans le même espace de nommage de l'élément dont ils dépendent (souvent, les DTD l'interdisent même)

Il est possible d'associer à un élément des attributs qui font partie d'un autre espace de nommage :

```
<?xml version="1.0"?>
<document xmlns:xlink="http://www.w3.org/1999/xlink">
  <image { xlink:type="simple"
         { xlink:actuate="onLoad"
         { xlink:show="embed"
         { xlink:href="http://www.acme.com/foo/bar.jpg"
           width="123"
           height="234" />
</document>
```

attributs "étrangers"

Pour qu'un attribut soit dans un espace de nommage il DOIT avoir un préfixe

Surcharge d'une déclaration d'espace de nommage qui utilise un préfixe :

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <foo:C xmlns:foo="http://www.bar.org/">
      <foo:D>abcd</foo:D>
    </foo:C>
  </foo:B>
</foo:A>
```

Surcharge d'une déclaration d'espace de nommage par défaut :

```
<A xmlns="http://www.foo.org/">
  <B>
    <C xmlns="http://www.bar.org/">
      <D>abcd</D>
    </C>
  </B>
</A>
```

Surcharge d'une déclaration d'espace de nommage qui utilise un préfixe avec une déclaration vide :

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <foo:C xmlns:foo="">
      <foo:D>abcd</foo:D>
    </foo:C>
  </foo:B>
</foo:A>
```



Surcharge d'une déclaration d'espace de nommage par défaut avec une déclaration vide :

```
<A xmlns="http://www.foo.org/">
  <B>
    <C xmlns="">
      <D>abcd</D>
    </C>
  </B>
</A>
```

Préfixes spéciaux `xml` et `xmlns`

`xmlns:svg="http://www.w3.org/2000/svg"`

`xml:lang="fr"`

Le préfixe `xml` est associé à l'espace de nommage : `http://www.w3.org/XML/1998/namespace`

**i** Dans le modèle logique, tous les éléments disposent de l'espace de nommage `xml`

Le préfixe `xml` sert pour les attributs spéciaux `xml:space`, `xml:lang` et `xml:base`

Le préfixe `xmlns` est utilisé exclusivement pour définir les espaces de nommage

URI `http://www.w3.org/2000/xmlns/`

(utilisé dans le DOM)

URI = identifiant d'objet abstrait, sans ressource physique

Quelques déclarations célèbres :

RDF

```
<rdf:RDF xmlns:rdf="http://www.w3.org/TR/REC-rdf-syntax#">
```

Dublin-core

```
<dc:title xmlns:dc="http://purl.org/dc/">
```

SVG

```
<svg xmlns="http://www.w3.org/2000/svg">
```

XSLFO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
```

XSLT

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

XHTML

```
<xhtml:xhtml xmlns:xhtml="http://www.w3.org/1999/xhtml">
```

XLink

```
<foo xmlns:xlink="http://www.w3.org/1999/xlink">
```

SOAP

```
<SOAP-ENV xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
```

Schema

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Les URI sont purement déclaratives

Il n'y a pas nécessairement quelque chose à l'adresse indiquée

Les parseurs et les applications ne sont pas tenus d'aller y chercher quoi que ce soit

- il n'y a rien qui puisse les intéresser
- la cible n'est pas nécessairement accessible par votre réseau

Les URI des espaces de nommage ne sont que des identificateurs formels

En général, on y trouve un document explicatif de l'utilisation de l'espace de nommage, mais ce n'est pas une obligation.

Les espaces de nommage du W3C font référence aux spécifications appropriées

Les URIs servent seulement à définir des clés universelles :

```
xmlns="urn:guid:88d969c9-f192-11d4-a65f-0040963251e6"  
xmlns="urn:isbn:2-84177-230-6"  
xmlns="urn:schemas-microsoft-com:office:office"
```

Les URIs doivent être absolues ; si elles sont relatives, elles ne sont pas résolues, et perdent de ce fait leur caractère d'identificateur universel

Les préfixes doivent toujours être associés à une URI par une déclaration `xmlns`

Les préfixes n'ont **aucune** signification particulière

Seule l'URI peut avoir une signification pour une application

## Exemple

Un processeur XSLT saura traiter indifféremment les feuilles de style suivantes :

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<extensible-stylesheet-language:stylesheet
  xmlns:extensible-stylesheet-language="http://www.w3.org/1999/XSL/Transform">
```

```
<tiens-voilà-du-boudin:stylesheet
  xmlns:tiens-voilà-du-boudin="http://www.w3.org/1999/XSL/Transform">
```

prénom

noms de famille

```
<xsl:stylesheet
  xmlns:tei="http://www.tei-c.org/ns/1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
  xmlns:math="http://www.w3.org/1998/Math/MathML"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:html="http://www.w3.org/1999/xhtml"
>
```

La recommandation des espaces de nommage ne définit rien d'autre qu'un système de nommage en 2 parties pour les éléments types et les attributs

En particulier, ils ne définissent pas:

- un moyen pour fusionner 2 documents qui utilisent des DTD différentes (les ambiguïtés dans le document fusionné doivent être levées spécifiquement, et les espaces de nommage sont un moyen d'y arriver)
- un moyen pour les associer à des schémas
- un moyen pour valider les documents qui les utilise
- comment associer les éléments types et les déclarations d'attributs des DTD à un espace de nommage

Les espaces de noms sont complètement indépendants des DTD

Il peut y avoir :

- des documents sans déclaration d'espace de nom qui utilisent une DTD
- des documents avec déclaration d'espace de nom sans DTD
- des documents avec déclaration d'espace de nom ET qui utilisent une DTD

Une déclaration `xmlns` peut être omise dans le document et déclarée dans la DTD :

```
<!ATTLIST livre      xmlns:xlink CDATA #FIXED "http://www.w3.org/1999/xlink">
```

```
<!ATTLIST svg xmlns      CDATA #FIXED "http://www.w3.org/2000/svg">
```

```
<?xml version="1.0"?>
<!DOCTYPE svg:svg [
    <!ATTLIST svg:svg xmlns:svg      CDATA #FIXED "http://www.w3.org/2000/svg">
]>
<svg:svg width="18cm" height="6cm" viewBox="0 0 200 120">
    <svg:rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
</svg:svg>
```

Les DTD peuvent contenir des noms qualifiés, mais les déclarations d'espace de nommage ne s'appliquent pas aux DTD

Les DTD doivent déclarer les éléments avec leur nom qualifié

Il n'y a pas de moyen de déterminer à quel espace de nommage un préfixe utilisé dans une DTD se réfère

Les noms qualifiés dans les DTD ne peuvent pas être associés à des noms universels

Les déclarations d'éléments types et d'attributs sont exprimées en termes de noms qualifiés, pas en terme de nom universel

La validation ne peut être redéfinie en terme de noms universels, comme cela est requis

```
<!ENTITY % dc-prefix "dc">
<!ENTITY % dc-colon ":">
```

```
<!ENTITY % dc-prefix "">
<!ENTITY % dc-colon "">
```

```
<!ENTITY % dc-prefix "dublin-core">
<!ENTITY % dc-colon ":">
```

Au choix...

```
<!ENTITY % dc "%dc-prefix;%dc-colon;">

<!ENTITY % dc-title "%dc;title">
<!ENTITY % dc-creator "%dc;creator">
<!ENTITY % dc-description "%dc;description">
<!ENTITY % dc-date "%dc;date">
```

```
<!ELEMENT %dc-title; (#PCDATA)>
<!ELEMENT %dc-creator; (#PCDATA)>
<!ELEMENT %dc-description; (#PCDATA)>
<!ELEMENT %dc-date; (#PCDATA)>
```

```
<!ELEMENT rdf:Description ( (%dc-title; | %dc-creator; |
                             %dc-description; | %dc-date;)*)>
```

```
<!ENTITY % dc "%dc-prefix;%dc-colon;">
<!ELEMENT %dc;title (#PCDATA)>
```

Non

Le parseur peut  
insérer des blancs

Les éléments héritent des déclarations d'espaces de nommage définis par eux ou leurs ancêtres.

L'élément racine (et ses descendants) est automatiquement doté de la déclaration :

```
xmlns:xml="http://www.w3.org/XML/1998/namespace"
```

Ces 2 documents sont équivalents...

```
<foo:A xmlns:foo="http://www.foo.org/">
  <foo:B>
    <bar:C xmlns:bar="http://www.bar.org/">
      <bar:D>abcd</bar:D>
      <foo:E>efgh</foo:E>
    </bar:C>
  </foo:B>
</foo:A>
```

... mais celui-ci n'est qu'une vue "conceptuelle" : il est interdit de définir un préfixe avec la chaîne "xml"



```
<foo:A xmlns:foo="http://www.foo.org/"
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
  <foo:B xmlns:foo="http://www.foo.org/"
    xmlns:xml="http://www.w3.org/XML/1998/namespace">
    <bar:C xmlns:bar="http://www.bar.org/"
      xmlns:foo="http://www.foo.org/"
      xmlns:xml="http://www.w3.org/XML/1998/namespace">
      <bar:D xmlns:bar="http://www.bar.org/"
        xmlns:foo="http://www.foo.org/"
        xmlns:xml="http://www.w3.org/XML/1998/namespace">abcd</bar:D>
      <foo:E xmlns:bar="http://www.bar.org/"
        xmlns:foo="http://www.foo.org/"
        xmlns:xml="http://www.w3.org/XML/1998/namespace">efgh</foo:E>
    </bar:C>
  </foo:B>
</foo:A>
```

Non

Certaines applications utilisent les espaces de nommage dans les contenus textuels des éléments et dans les valeurs d'attributs.

Ces applications requièrent en général que les déclarations d'espace de nommage idoines ont été réalisées.

## XSLT

```
<xsl:template match="cours:cours/personne:auteur"  
  xmlns:cours="urn:acme-org:cours"  
  xmlns:personne="http://www.bar.com/individus"  
  xmlns:dsml="urn:oasis:names:tc:DSML:2:0:core"  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
>  
  <h3><xsl:value-of select="personne:titre"/>  
    <b><xsl:value-of select="personne:prénom"/></b>  
    <xsl:value-of select="personne:nom"/>  
  </h3>  
  <xsl:apply-templates select="../dsml:attr[ @name = 'telephoneNumber' ]/dsml:value"/>  
</xsl:template>
```

## W3C XML Schema

```
<xs:element name="âge"  
  xmlns:xs="http://www.w3.org/2001/XMLSchema"  
>  
  <xs:simpleType>  
    <xs:restriction base="xs:integer">  
      <xs:minInclusive value="0"/>  
      <xs:maxInclusive value="100"/>  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

- Quelles sont les chaînes permises dans les URI d'espace de nom ?

- Les déclarations d'espaces de nommage sont-elles porteuses d'information ?

→ Ces 2 documents sont-ils équivalents ?

```
<a xmlns:x="x"><b/></a>
```

```
<a xmlns:x="x"><b xmlns:x="x"/></a>
```

- Les déclarations d'espaces de nommage qui ne sont pas référencées sont-elles porteuses d'information ?

→ Ce document est-il équivalent à l'un des 2 autres ci-dessus ?

```
<a><b/></a>
```

→ Problème des QName dans le contenu