

# Le langage XML

Dominique PRESENT  
I.U.T. de Marne la Vallée

## XML pour structurer des documents

**XML (*eXtensible Markup Language*) est un metalangage :**

- norme du W3C (<http://www.w3.org/TR/REC-XML>), datant du 10/02/98
- XML sert à **stocker des données structurées dans un fichier texte** :
  - données structurées = arbre (racine – branches - feuilles)
  - *Markup* = balise  $\Rightarrow$  structuration
- XML  $\neq$  HTML :
  - n'est pas limité au Web
  - doit être adapté à chaque utilisation
  - pas toujours très lisible
- XML ressemble à HTML :
  - langages à balise, mais des **balises non définies**
  - dérivés de SGML

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Un standard ouvert indépendant des applications

1. XML est un **standard ouvert**
2. XML est une solution complète de stockage, échange, manipulation, transformation, etc. de données structurées

Standard  $\Rightarrow$  économie de développement. Produits disponibles :

- éditeurs
- analyseur syntaxique
- validateur (vérification de cohérence)
- moteur de transformation
- etc.

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Un langage structuré

Exemple de fichier XML :

[hello.xml](#)

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <conversation>
3   <greeting mood="polite">Hello, world!</greeting>
4   <response mood="angry">Leave me alone!</response>
5 </conversation>
```

1. en-tête "presque" obligatoire : c'est une PI (*processing instruction*)
2. balise ouvrante (ici la racine de l'arbre) : débute un noeud (un élément)
3. mood est un attribut, polite est sa valeur (guillemets obligatoires)
5. balise fermante correspondant à la BO de la ligne 2

## XML structure, mais n'organise pas

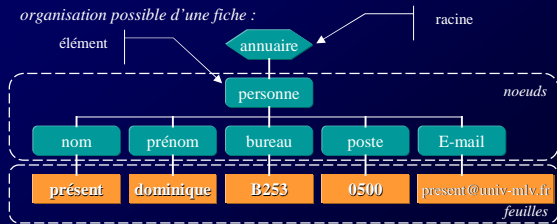
XML permet de représenter des données structurées :

- données textuelles (binaire : codage, par exemple basé sur *mime*)
- organisées :
  - on manipule un **document** constitué d'**éléments**
  - un **élément** peut être constitué simplement de **texte** ou contenir d'autres éléments (ou un mélange des deux)
  - un élément peut être associé à des informations complémentaires, les **attributs**
- la structure est celle d'un arbre :
  - un document XML = un arbre
  - un élément = un noeud de l'arbre
- le standard indique comment traduire l'arbre en un texte XML, pas comment organiser les données

## Exemple : structurer un annuaire

- structure : stocker les fiches d'un annuaire (nom, prénom, bureau, numéro de poste, *email*)
- le texte du document : les informations !
- organisation : s'arranger pour que les informations restent correctement groupées (ne pas mélanger les données !)

organisation possible d'une fiche :



## Structure du fichier XML d'une fiche

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<annuaire>
  <personne>
    <nom>PRESENT</nom>
    <prenom>Dominique</prenom>
    <bureau>B253</bureau>
    <poste>0500</poste>
    <email>present@univ-mlv.fr</email>
  </personne>
<!-- suite de l'annuaire -->
</annuaire>
```

- la racine est **unique**
- le contenu d'un élément est :
  - d'autres éléments
  - du texte : les *character data* ( < et & interdits )
- balise ouvrante ou fermante , nom d'un noeud
- texte : feuille de l'arbre
- Ne pas confondre les éléments (information) et les balises (syntaxe).

IUT de Marne la Vallée

langage XML

© D. PRESENT

## En-tête d'un fichier XML

- il est **conseillé** de commencer un document XML par :

```
< ?xml version="1.0" ?>
```
- si on donne l'en-tête, version="1.0" est **obligatoire**
- l'attribut encoding permet d'indiquer le codage des caractères du fichier :

```
< ?xml version="1.0" encoding="UTF-16" ?>
```

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

 par défaut (supporte l'ASCII)

```
< ?xml version="1.0" encoding="ISO-8859-1" ?>
```

 sous linux
- < ?nom ?> : une *Processing Instruction*. Indique aux logiciels comment traiter le document :
  - encodage
  - associer une feuille de style à un document

IUT de Marne la Vallée

langage XML

© D. PRESENT

## La structure découle de l'organisation

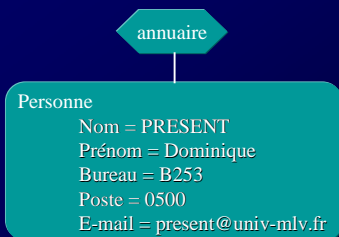
- On organise les données en décidant de la structure de l'arbre :
  - le nom des éléments
  - l'ordre des éléments
  - les relations d'inclusion
  - la position des données (c'est-à-dire du texte)
  - les contraintes sur les données (texte quelconque, valeur numérique, etc.)
  - les attributs
- L'utilisation de langages associés nécessite une organisation particulière, par exemple :
  - MathML : pour décrire des équations
  - xbel : pour décrire des *signets*
  - SVG : dessin vectoriel
  - XHTML : HTML re-spécifié en XML
  - etc.

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Un autre exemple d'organisation



IUT de Marne la Vallée

langage XML

© D. PRESENT

## Fichier utilisant des attributs

[Annuaire2.xml](#)

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <annuaire>
3   <personne
4     nom="PRESENT"
5     prenom="Dominique"
6     bureau="B253"
7     poste="0500"
8     email="present@univ6mlv.fr"/>
9 <!-- suite de l'annuaire -->
10 </annuaire>
```

- attributs ⇔ annotations d'un noeud
- élément vide ⇒ feuille

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Une bonne organisation facilite le traitement

Le choix de la structure n'est pas anodin :

- l'intérieur d'un attribut n'est pas structuré
- au maximum un seul exemplaire d'un attribut dans un élément
- impossible de faire ce qui suit avec des attributs sans introduire lourdeurs et limitations :

[annuaire3.xml](#)

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <annuaire>
3   <personne>
4     <nom>PRESENT</nom>
5     <prenom>Dominique</prenom>
6     <bureau>B253</bureau>
7     <poste>0500</poste>
8     <email><nom>present</nom>
9       <domaine>univ-mlv.fr</domaine></email>
10   </personne>
11 </annuaire>
```

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Une organisation lourde et peu modifiable

annuaire4.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <annuaire>
3   <personne nom="PRESENT"
4     prenom="Dominique"
5     bureau="B253"
7     poste="0500"
8     email-nom="present"
9     email-domaine="univ-mlv.fr" />
10 </annuaire>

```

- oblige à prévoir tous les cas (redondance dans les traitements ultérieurs) :
  - on peut toujours structurer le contenu d'un élément *a posteriori*, pas celui d'un attribut

IUT de Marne la Vallée

langage XML

© D. PRESENT

## XML dans HTML

- Les balises XML ne sont pas interprétables par les navigateurs (balises non prédéfinies)
- Possibilité d'inclure des « îlots de données » (Data Islands) dans un fichier HTML en indiquant :
  - Le fichier source XML : `<xml id="fichierxml" src="simple.xml"></xml>`
  - La balise identifiant le champ de données à insérer : `datafld=" balise_xml"`

[Emailpro.html](#)

```

<html>
<body>
E-mail professionnel
<xml id="fichierxml" src="annuaire1.xml" />
<table border="1" datasrc="#fichierxml">
  <tr>
    <td><span datafld="nom"></span></td>
    <td><span datafld="prenom"></span></td>
    <td><span datafld="email"></span></td>
  </tr>
</table>
</body>
</html>

```

```

annuaire1.xml
<?xml version="1.0"
encoding="ISO-8859-1"?>
<annuaire>
  <nom>PRESENT</nom>
  <prenom>Dominique</prenom>
  <bureau>B253</bureau>
  <poste>0500</poste>
  <email>present@univ-
mlv.fr</email>
</personne>
</annuaire>

```

IUT de Marne la Vallée

langage XML

© D. PRESENT

## XSL : la feuille de style du XML

- Le langage XSL est au XML ce que la feuille de style CSS est au HTML
- Il permet la manipulation des données : sélection, tri, filtrage, choix conditionnel
- XSL est un ensemble de 3 langages :
  - XSLT pour transformer le format du document (ex.: XML → HTML)
  - Xpath pour définir et adresser des parties de document XML
  - XML Formater pour « formater » le document à destination de « pockets »
- Le langage XSL est dérivé du XML : il reprend sa structure et sa syntaxe
- le document XSL comporte un document HTML ou XHTML reconnu par les navigateurs (version 5 ou plus pour Internet Explorer)
- XSL fonctionne avec des modèles (templates) pour définir l'affichage des éléments

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <body>
          <html>
            <xsl:template>
              <xsl:stylesheet>

```

En-tête xml

déclaration

Document HTML

Modèle et son attribut : s'applique à toutes les balises XML ("/")

IUT de Marne la Vallée

langage XML

© D. PRESENT

## mailISSI.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="orderby.xsl"?>
<annuaireISSI>
  <email>
    <nom>PRESENT</nom>
    <prenom>Dominique</prenom>
    <mailUMLV>present@univ-mlv.fr</mailUMLV>
  </email>
  <email>
    <nom>MEYER</nom>
    <prenom>Claudie</prenom>
    <mailUMLV>claudie.meyer@univ-mlv.fr</mailUMLV>
  </email>
  <email>
    <nom>CACALY</nom>
    <prenom>Serge</prenom>
    <mailUMLV>serge.cacaly@univ-mlv.fr</mailUMLV>
  </email>
</annuaireISSI>
```

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Opération de tri : orderby.xsl

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <html>
      <body>
        <table border="1" cellspacing="0" cellpadding="3">
          <tr>
            <td>nom</td>
            <td>prénom</td>
            <td>mailUMLV</td>
          </tr>
          <xsl:for-each select="annuaireISSI/email">
            <tr>
              <td><xsl:value-of select="nom"/></td>
              <td><xsl:value-of select="prenom"/></td>
              <td><xsl:value-of select="mailUMLV"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Quelques opérateurs

- := pour égal ; != pour différent ; > pour plus grand ; < pour plus petit  
<xsl:for-each select="chemin\_d'accès[balise opérateur 'valeur']">  
</xsl:for-each>
- Recherche de valeur : <xsl:if match=".[balise = 'valeur']">  
Balises HTML  
</xsl:if>
- Choix conditionnel de valeur :  
<xsl:choose>  
condition vérifiée <xsl:when test=".[balise='valeur']">  
balises HTML  
</xsl:when>  
sinon <xsl:otherwise>  
balises HTML  
</xsl:otherwise>
- Combinés à des fichiers CSS, les fichiers XSL permettent de générer des modèles de structurés de documents pour l'entreprise

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Microsoft et le XML

- La problématique des entreprises :
  - Normaliser les processus d'usage de l'informatique ;
  - Réduire les coûts de développement et de formation aux applications ;
  - Structurer les documents pour faciliter la G.E.D. (Gestion Electronique des Documents) et diminuer les saisies ;
  - Faciliter les échanges de documents entre utilisateurs, serveurs, pages Web, applications.
- Remarques :
  - La suite MS Office 2003 permet un transfert sélectif bidirectionnel entre des documents XML et les applicatifs de MS Office 2003 ;
  - La suite MS Office 2003 ne gère pas les attributs des nœuds XML (il faut passer par VBA) ;
  - MS Publisher 2003, MS OneNote 2003, MS Business Contact Manager ne proposent pas d'utilitaires pour traiter du XML (il faut programmer ces outils)

IUT de Marne la Vallée

langage XML

© D. PRESENT

## XSD pour la structuration des documents

- Pour utiliser professionnellement des documents structurés en XML, il faut établir un schéma de validation XSD des fichiers XML ;
- Les fichiers XSD utilisent la syntaxe XML.

[Email.xsd](#)

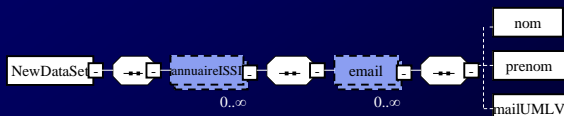
```
<?xml version="1.0" standalone="yes"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:element name="annuaireISSI">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="email" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="nom" type="xsd:string" minOccurs="0"/>
              <xsd:element name="prenom" type="xsd:string" minOccurs="0"/>
              <xsd:element name="mailUMLV" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Forme graphique d'un DTD



Présentation sous l'éditeur XML Spy

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Excel 2002 et les fichiers XML

- Excel 2002 permet d'inclure les données de fichiers XML dans leur intégralité (pas d'inclusion sélective) ;
- Excel 2002 ne permet pas l'exporter les données vers un fichier XML (contrairement à Excel 2003)

1	annuaire						
2	/personne/bureau	/personne/email	/personne/nom	/personne/poste	/personne/poste#agg	/personne/prenom	
3	B253	present@univ-mv.fr	PRESENT	500	500	Dominique	
4	B255	claudie.meyer@MEYER	523	523	Claudie		
5	B250	serge.cacaly@CACALY	512	512	Serge		
6							
7							

IUT de Marne la Vallée

langage XML

© D. PRESENT

## Excel 2003 et les fichiers XML

- Importer des données :**
  - Les fichiers XML doivent être associés à un fichier XSD. Ce fichier permet l'importation sélective de données ;
  - Onglet **Options** -> **Source XML**. Cliquer sur le bouton « **XML Maps** » dans le coin inférieur droit et choisir le fichier XSD. La structure des éléments du fichier XML apparaît dans une fenêtre ;
  - Importer les champs utiles dans la feuille Excel par un couper -glisser ;
  - Importer les données XML, cliquer sur le bouton **importer données XML** (**Import XML Data**) de la barre d'outils **Liste (List)**
- Exporter des données :**
  - Les données peuvent être triées, filtrées et manipulées comme n'importe quelles données ;
  - Les liaisons sont dans le sens fichier XML -> Excel ;
  - Pour modifier un fichier XLS à partir de la feuille Excel par la fonction **Export**

IUT de Marne la Vallée

langage XML

© D. PRESENT

## MS ACCESS 2002 et les fichiers XML

- MS Access 2002 ne traite que l'intégralité des fichiers XML (pas de traitement sélectif) ;
- Il assure l'importation et l'exportation des données à partir et vers des fichiers XML (l'exportation inclue la création des fichiers XSL et XSD, mais avec un code compliqué)

nom	prenom	mailMLV
PRESENT	Dominique	present@univ-mv.fr
MEYER	Claudie	claudie.meyer@univ-mv.fr
CACALY	Serge	serge.cacaly@univ-mv.fr

IUT de Marne la Vallée

langage XML

© D. PRESENT



## MS INFOPATH 2003 pour les formulaires

- Un des 3 nouveaux outils de MS Office System 2003 (avec *OneNote* et *Business Contact Manager*) de gestion et traitement de données au format XML ;
- Outil de création de formulaires, il permet par exemple d'ajouter des éléments à une base de données ACCESS ;

IUT de Marne la Vallée

langage XML

© D. PRESENT

## XMLRAD : un environnement de développement

- environnement de développement web, un RAD, basé sur les technologies XML/XSL ;
- lié à la plateforme Windows, génère un code au plus près des standards du W3C (les requêtes SQL n'utilisent qu'un standard compris de toutes les bases) ;
- Le propre de cet outil est l'accès à la base, et la génération automatique d'une "interface" HTML ;
- L'environnement permet de créer des écrans à partir de description d'extraction de la base de données. Très peu de code est à faire pour obtenir cela. Les applications que XMLRAD génère sont dites dirigées par les données.
- s'associe aux environnements de développement tel que Delphi, Kylix, Borland C++Builder, Visual Studio .NET, JBuilder, Eclipse, PHP et permet de générer des applications pour les plateformes Windows (Win32), .NET et Linux avec les Serveurs Web IIS ou Apache.
- XMLRAD 2006 requiert Windows 2000, Windows XP Pro ou Windows 2003 avec le Serveur IIS, Apache 1.3 ou Apache 2.0 installé pour fonctionner. Les applications générées peuvent fonctionner sur Apache ou IIS et sur les plateformes Windows, Linux, .NET, J2EE et PHP.

IUT de Marne la Vallée

langage XML

© D. PRESENT