eXtensible Markup Language

XML

Alexis NEDELEC

Centre Européen de Réalité Virtuelle
Ecole Nationale d'Ingénieurs de Brest

de l'ARPA au W3C

Historique

- 1969 : **ARPA** : **A**dvanced **R**esearch **P**roject **A**gency
- 1970 : **ARPANET**: ARPA**NET**work
- 1970 : **NCP** : **N**etwork **C**ontrol **P**rotocol
- 1972 : **INWG** : **I**nter**N**etwork **W**orking **G**roup (EU + UE)
- 1973 : **TCP** : **T**ransmission **C**ontrol **P**rotocol
- 1973 : **IP** : **I**nternet **P**rotocol
- 1990 : **HTML** : **H**yper**T**ext **M**arkup **L**anguage
- 1996 : **HTTP** : **H**yper**T**ext **T**ransfert **P**rotocol

TCP/IP, HTTP, HTML : fondements du **W**orld **W**ide **W**eb

- **W3C** : le consortium (`https://www.w3.org`)
- **W3schools** : les tutoriaux (`https://www.w3schools.com`)

HTML

Une page personnelle

```
<html>
 <head>
   <title>Ma page Personnelle </title>
 </head>
 <body>
  <h1>section de ma page</h1>
  <h2>sous-section de la page</h2>
  <p>premier paragraphe</p>
  <p>deuxieme paragraphe</p>
 </body>
</html>
```

HTML

La même page personnelle

```
<html>
 <head>
  <title> Ma page Personnelle </title>
 </head>
 <body>
  <b>
   <font size="6">section de ma page</font>
   </br></br>
   <font size="5">sous-section de la page</font>
   </br></br>
  </b>
   <font size="3"> premier paragraphe </font></br></br>
   <font size="3"> deuxieme paragraphe </font></br>
</body></html>
```

SGML

Standard Generalized Markup Language

- séparer la structure logique d'un document
  - titres, chapitres,paragraphes ...
- de sa mise en page
  - livre, journal, écran ...

Normalisation de documents

- **GED** : **G**estion **E**lectronique des **D**ocuments
- **EDI** : **E**change de **D**onnées **I**nformatisées
- **XML** : e**X**tensible **M**arkup **L**anguage

XML

Description

- textuel simple à écrire, raisonnablement lisible
- auto-descriptif, règles strictes de validation
- multilingue (Unicode), programmable (arbre)

Parseur XML : MSXML, Xerces, libXML ...

- cohérence des documents XML (syntaxe, conformité)
- chargement de l'arbre du document en mémoire
- accès aux élements, noeuds du document (tokens)

Processeur XML : MSXML, Xalan, libXSLT ...

- modifier, transformer un document XML
- XSLT (XML Stylesheet Language Transformation)

XML

Objectifs

- **HTML** : présentation (affichage) de données,
- **XML** : représentation (description) de données

Structuration de l'information

- **DTD** : règles de validation XML
- **XML Schemas**: spécifications W3C pour remplacer les DTD
- **XSLT**: transformation de documents XML
- **XPath**: associé à XSLT pour naviguer dans un arbre XML

W3 Consortium

Manipulation, contrôle, liaison de documents XML

- **X Query**: langage de requêtes pour accéder au contenu
- **DOM** : XML objet manipuler le document XML
- **SAX** : XML événementiel, pour identifier des parties
- **XLink**: navigation hypertexte entre documents XML
- **XPointer**: accès aux sous-ensembles de documents XML
- **XForms**: équivalent des formulaires HTML
- **XSL-FO**: pour publication HTML,PDF,SVG ,MathML ...
- **SVG** : **S**calable **V**ector **G**raphics (dessins vectoriels 2D)
- **X3D** : XML pour faire de la 3D

standards XML, tutoriel SVG

Structuration

Structuration de documents

- prologue, instructions de traitement
- arbre d'éléments

Exemple de document

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl"?>
<html>
  <head>
    <title> Ma page Personnelle </title>
  </head>
  <body>
    ...
  </body>
</html>
```

Structuration

Briques de construction

- Déclarations: `<?xml version="1.0"?>`
- Balises: `<tag>...</tag>`, marquage des éléments
- Eléments: `<tag>balise avec son contenu</tag>`
- Attributs: information sur les éléments:
  `<img src="xml.gif">...</img>`
- Instructions de traitement : `<?cible valeur?>`
- Entités (référence sur): variables communes aux documents XML (`  &gt; &quot; &apos; &amp...`)
- PCDATA: données à traiter par le parseur XML
- CDATA: données brutes, non-traitées par le parseur XML

Structuration

Entités XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE monDocument [
<!ENTITY monEntite "toto">
]>
<monDocument>
  mon entite est : &monEntite;
</monDocument>
```

Données brutes

- `<inegalite> deux<trois>un </inegalite>` : KO
- `<inegalite><![CDATA[deux<trois>un]]></inegalite>` : OK

Structuration

Exemple : messages.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<message>
  <expediteur>Nedelec</expediteur>
  <destinataire> EdT </destinataire>
   <sujet>A propos de XML</sujet>
   <contenu> quel jour ?  </contenu>
</message>
```

Remarques

- utilisation de XML version 1.0
- codage de jeux de caractères :
  - `ASCII`, `Unicode`, `UTF-8` ...
- définir les balises `<message>`, `<destinataire>`, ...

Element XML

Un élément XML contient

- uniquement des éléments (element content)
- des éléments et des données (mixed content)
- des données simples, donc du texte (simple content)
- être vide (empty content)
- avoir des attributs (ex: `<message date="01/02/03">`)

Sur l'exemple précédent

- `<destinataire>Nedelec</destinataire>` : est un élément
- `destinataire` : est l'appelation de l' élément
- `Nedelec` : est la donnée de l'élément
- `PCDATA` (Parsed Character DATA) :
  le contenu d'un élément qui ne contient pas d'autres balises

Element XML

Element ou attribut ?

```
<message date="01/02/03"> ... </message>
<message>
  <date>01/02/03</date>
  ...
</message>
<message>
  <day>01</day>
  <month>02</month>
  <year>03</year>
   ...
</message>
```

Attribut XML

Element ou attribut ?

- les attributs ne peuvent pas contenir de valeurs multiples
- les attributs ne peuvent pas décrire des structures
- on peut enrichir les éléments pour les faire évoluer
- les attributs sont difficiles à manipuler par des programmes

Contre-Exemple

```
<message day="01" month="02" year="03"
  expediteur="Nedelec" destinataire="EdT"
  sujet="A propos de XML"
  contenu="quel jour ?"
</message>
```

Attributs XML

Exemple d'utilisation d'attributs

```
<messages>
 <message id="001">
  <expediteur>Nedelec</expediteur>
  <destinataire>EdT</destinataire>
  <sujet>A propos de XML</sujet>
  <contenu> quel jour ?</contenu>
 </message>
 <message id="002">
  <expediteur>EdT</expediteur>
  <destinataire>Nedelec</destinataire>
  <sujet>Re: A propos de XML</sujet>
  <contenu>le: <day>01</day>...<year>03</year></contenu>
 </message>
</messages>
```

Syntaxe XML

Règles syntaxiques

- tout document a un élément racine :
  `<root></root>`
- toute balise doit-être ouverte et fermée :
  `<tag>...</tag>` ou `<tag... />`
- les balises sont sensibles à la "Casse" :
  `<tag>, <Tag>, <TAG>`
- tout sous-élément (fils) doit-être correctement imbriqué:

```
<root>
 <child><subchild>
    ...
 </subchild></child>
</root>
```

Syntaxe XML

Règles syntaxiques

- définition des attributs dans la balise ouvrante :
  `<Chapitre numero="1">`
- valeurs d'attributs entre guillemets :
  `<Chapitre numero="1" media="paper">`
- neutraliser les caractères spéciaux par `CDATA` :
  - `<inegalite><![CDATA[deux<trois>un]]></inegalite>`
- conventions de nommage de balise, éviter les .,-
  `<nom_de_balise>` : OK, `<nom.de-balise>` : KO
- commentaire: `<!- Ceci est un commentaire ->`

Syntaxe XML

Document XML syntaxiquement correct

Well formed XML document

- le document XML a une balise racine
- les éléments XML sont fermés
- ils sont sensible à la "case"
- ils sont bien imbriqués
- la valeur des attributs XML est entre "guillemets"

Validation des documents XML

valid XML document

- Document Type Definition (DTD)
- XML schema (XSD)

Document Type Definition

Structuration de document XML

- le document `<!DOCTYPE ...>`
- les éléments `<!ELEMENT ...>`
- les attributs `<!ATTLIST ...>`
- les entités `<!ENTITY ...>`
- les données `CDATA, #PCDATA`

Document XML valide

- document XML bien **formé**
- respecte les règles de DTD interne et/ou externe

Document Type Definition

DTD interne

```
<!DOCTYPE messages [
<!ELEMENT messages (message+)>
<!ELEMENT message (expediteur,destinataire,sujet,contenu)>
<!ELEMENT expediteur (#PCDATA)>
<!ELEMENT destinataire (#PCDATA)>
<!ELEMENT sujet (#PCDATA)>
<!ELEMENT contenu (#PCDATA)>
]>
<messages><message>
 <expediteur>Nedelec</expediteur>
 <destinataire> EdT </destinataire>
 <sujet>A propos de XML</sujet>
 <contenu>quel jour ? </contenu>
</message></messages>
```

Document Type Definition

DTD externe (message.dtd)

```
<!DOCTYPE messages [
<!ELEMENT messages (message+)>
<!ELEMENT message (expediteur,destinataire,sujet,contenu)
<!ATTLIST message id CDATA #REQUIRED>
<!ELEMENT expediteur (PCDATA)>
...
]>
```

DTD externe (message.xml)

```
<?xml version="1.0"?>
<!DOCTYPE messages SYSTEM "message.dtd">
<messages><message id="001">
  ...
</message></messages>
```

Cascading Style Sheets

Feuille de style : message.css

```
message {
background-color: #fffff; width: 100%; }
expediteur  {
font-size: 20pt; margin-bottom: 30pt; margin-left: 0; }
destinataire {
font-size: 20pt;margin-bottom: 30pt; margin-left: 0; }
sujet {
display: block;
color: #FF0000; font-size: 15pt;}
contenu {
color: #0000FF; font-size: 20pt;}
```

Cascading Style Sheets

Document XML : message.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="message.css"?>
<messages>
 <message>
  <expediteur>Nedelec</expediteur>
  <destinataire>EdT</destinataire>
  <sujet>A propos de XML</sujet>
  <contenu> Quel jour ?</contenu>
 </message>
 ...
</messages>
```

eXtensible Stylesheet Language

Document XML : les données à transformer

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="message.xsl"?>
<!-- message.xml -->
<messages>
 <message>
  <expediteur>Nedelec</expediteur>
  <destinataire>EdT</destinataire>
  <sujet>A propos de XML</sujet>
  <contenu> Quel jour ?</contenu>
 </message>
 ...
</messages>
```

XML et XSL

Document XSL : les transformations à appliquer

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
<!-- message.xsl -->
```

Règles de transformation : nœud racine ("/")

```
<xsl:template match="/">
  <html><body>
    <h2>Mes messages</h2>
    <xsl:apply-templates/>
  </body></html>
</xsl:template>
```

Transformation XSL

Règles de transformation : élément racine ("messages")

```
<xsl:template match="messages">
   <xsl:apply-templates select="message"/>
</xsl:template>
```

Règles de transformation : nœuds ("message")

```
<xsl:template match="message">
  <p>
  <xsl:apply-templates select="expediteur"/>
  <xsl:apply-templates select="destinataire"/>
  <xsl:apply-templates select="sujet"/>
  <xsl:apply-templates select="contenu"/>
  </p>
</xsl:template>
```

XML et XSL

Règles de transformation: nœuds ("expediteur")

```
<xsl:template match="expediteur">
  Expediteur: <span style="color:#ff0000">
  <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```

Règles de transformation: nœuds ("destinataire")

```
<xsl:template match="destinataire">
  Destinataire: <span style="color:#000fff">
  <xsl:value-of select="."/></span>
<br />
</xsl:template>
```

eXtensible Stylesheet Language

Règles de transformation nœuds ("sujet")

```
</xsl:template>
<xsl:template match="sujet">
  Sujet: <span style="color:#ff0000">
  <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```

Règles de transformation: nœuds ("contenu")

```
<xsl:template match="contenu">
Contenu: <span style="color:#ff0000">
<xsl:value-of select="."/></span>
<br />
</xsl:template>
```

XML Schema Definition

XML Schema

- alternative au DTD
- recommandations W3C (mai 2001)
- langage de schéma XSD

XML Schema vs DTD

- ils sont écrits en XML
- auto-descriptifs (même parseur, même règles)
- types plus riches (booléens,numériques, dates ...)
- 1 document XML / plusieurs schémas (espace de nommage)

XML Schema Definition

Données XML : message.xml

```
<?xml version="1.0"?>
<messages
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="messages.xsd">
 <message>
  <expediteur>Nedelec</expediteur>
  <destinataire>EdT</destinataire>
  <sujet>A propos de XML</sujet>
  <contenu> Quel jour ?</contenu>
 </message>
</messages>
```

XML Schema Definition

Schéma XSD : message.xsd

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="messages">
  <xsd:complexType>
   <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <!--  TODO : message element definition -->
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

XML Schema Definition

Schéma XSD : élément <message>

```
<xsd:element name="message">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="expediteur" type="xsd:string"/>
   <xsd:element name="destinataire" type="xsd:string"/>
   <xsd:element name="sujet" type="xsd:string"/>
   <xsd:element name="contenu" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attribute name="id" type="xsd:string"/>
 </xsd:complexType>
</xsd:element>
```

XML Schema Definition

Schéma XSD : élément racine <schema>

```
<?xml version="1.0"?>
<xsd:schema
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 targetNamespace="http://www.monschema.org"
 xmlns="http://www.monschema.org"
 elementFormDefault="qualified">
 ...
</xsd:schema>
```

Attributs de l'élément <schema>

- `xmlns:...` : espace de nommage d'éléments
- `targetNamespace` : espace de nommage du schéma
- `elementFormDefault` : nommage des élements du schéma

XML Schema Definition

Données XML : message.xml

```
<?xml version="1.0"?>
<messages xmlns:"http://www.monschema.org"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
 xsi:schemaLocation=
  "http://www.monschema.org message.xsd">
 <message>
  <expediteur>Nedelec</expediteur>
  <destinataire>EdT</destinataire>
  <sujet>A propos de XML</sujet>
  <contenu> Quel jour ?</contenu>
 </message>
</messages>
```

Eléments, attributs

<element> : définition d'un élément

- `<element name="aName" type="built-in-type"/>`

Types prédéfinis

- `decimal, integer ,boolean`
- `string, date, time`

<element> : autres attributs

- `default` : une valeur par défaut modifiable
- `fixed` : une valeur par défaut non-modifiable

<element> : exemple

```
<xsd:element name="color"
             type="xsd:string" default="red"/>
```

Eléments, attributs

<attribute> : définition d'un attribut

- `<attribute name="aName" type="built-in-type"/>`

Types prédéfinis

- `decimal, integer ,boolean`
- `string, date, time`

<attribute> : autres attributs

- `default` : une valeur par défaut modifiable
- `fixed` : une valeur par défaut non-modifiable

<attribute> : exemple

```
<xsd:attribute name="color" type="xsd:string"
               default="red" use="required"/>
```

Eléments simples : <simpleType>

<restriction> : sur une plage de valeur

```
<xsd:element name="age">
 <xsd:simpleType>
  <xsd:restriction base="xsd:integer">
   <xsd:minInclusive value="0"/>
   <xsd:maxInclusive value="120"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

Eléments simples : <simpleType>

<enumeration> : sur un ensemble de valeurs

```
<xsd:element name="voiture">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:enumeration value="Renault"/>
   <xsd:enumeration value="Citroen"/>
   <xsd:enumeration value="Peugeot"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

Eléments simples : <simpleType>

<pattern> : sur une série de valeurs

```
<xsd:element name="letter">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
   <xsd:pattern value="[a-z]"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

Eléments simples : <simpleType>

<length> : sur la longueur d'une châine

```
<xsd:element name="password">
 <xsd:simpleType>
  <xsd:restriction base="xsd:string">
<!--    <xs:length value="8"/> -->
   <xsd:minLength value="5"/>
   <xsd:maxLength value="8"/>
  </xsd:restriction>
 </xsd:simpleType>
</xsd:element>
```

Eléments complexes : <complexType>

<complexType> : élément structuré

- élément vide
  - <xsd:complexType>, <xsd:attribute>
- ne contient que des éléments
  - <xsd:sequence> <xsd:element>
- ne contient que du texte
  - <xsd:simpleContent>
- contient des éléments et du texte
  - <xsd:complexType mixed="true">

Element complexes : <complexType>

schéma <messages>

```
<?xml version="1.0"?>
<xsd:schema
 xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="messages">
  <xsd:complexType>
   <xsd:sequence minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="message">
     <!-- TODO message -->
    </xsd:element>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>
</xsd:schema>
```

Element complexes : <complexType>

schéma <message>

```
<xsd:element name="message">
 <xsd:complexType>
  <xsd:sequence>
   <xsd:element name="expediteur" type="xsd:string"/>
   <xsd:element name="destinataire" type="xsd:string"/>
   <xsd:element name="sujet" type="xsd:string"/>
   <!-- TODO contenu -->
  </xsd:sequence>
  <xsd:attribute name="id"
                 type="xsd:string" use="required"/>
 </xsd:complexType>
</xsd:element>
```

Element complexes : <complexType>

schéma <contenu>

```
<xsd:element name="contenu">
 <xsd:complexType mixed="true">
  <xsd:sequence minOccurs="0" maxOccurs="unbounded">
   <xsd:element name="jour" type="xsd:date"/>
  </xsd:sequence>
 </xsd:complexType>
</xsd:element>
```

Element complexes : <complexType>

Document XML messages.xml

```
<messages
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="messages.xsd">
  <message id="001">
     <expediteur>Nedelec</expediteur>
     <destinataire>EdT</destinataire>
     <sujet>A propos de XML</sujet>
     <contenu> Quel jour ?</contenu>
  </message>
```

Element complexes : <complexType>

Document XML messages.xml

```
  <message id="002">
    <expediteur>EdT</expediteur>
    <destinataire>Nedelec</destinataire>
    <sujet>Re: A propos de XML</sujet>
   <contenu>Le : <jour>2001-02-03</jour></contenu>
  </message>
</messages>
```

Element complexes : <complexType>

Réutilisation : type

```
<xsd:element name="personalMsg" type="message"/>
<xsd:element name="publicMsg" type="message"/>

<xsd:complexType name="message">
 <xsd:sequence>
  <xsd:element name="expediteur" type="xsd:string"/>
    ....
 </xsd:sequence>
 <xsd:attribute name="id" type="xsd:string" />
</xsd:complexType>
```

Réutilisation d'éléments complexes

Héritage : <extension>, <complexContent>

```
<xsd:element name="publicMsg" type="message"/>
<xsd:element name="privateMsg" type="fullMessage"/>
<xsd:complexType name="message">
  ...
</xsd:complexType>

<xsd:complexType name="fullMessage">
 <xsd:complexContent>
  <xsd:extension base="message">
   <xsd:element name="affinity" type="xsd:string"/>
  </xsd:extension>
 </xsd:complexContent>
</xsd:complexType>
```

Bibliographie

Ouvrages

- **D. Hunter** "Initiation à XML"
  ed. Eyrolles Wrox Press 2000
- **S. Lecomte** "XML par la pratique"
  ed. ENI 2005
- **K. Williams et. al.** "XML et les Bases de Données"
  ed. Eyrolles Wrox Press 2000
- **A. Brillant** "XML Cours et exercices"
  ed. Eyrolles Wrox Press 2007
- **G. Chagnon, F. Nolot** "XML"
  ed. Pearson Education, coll. Synthex 2007

Bibliographie

Liens Au Net

- le consortium W3C : `www.w3.org`
- l'organisme : `www.xml.org`
- les tutoriaux : `www.w3schools.com`
- cours en ligne de Gilles Chagnon : `www.licence.elec.upmc.fr/S_tec/coursEnLigne/xml/index.html`
- Le club d'entraide des développeurs :`www.developpez.com`