Université Ahmed Zabaneh - Relizane
Faculté des Sciences et de la Technologie
Département d'Informatique

Données Semi Structurées

Chapitre 2: Noyau XML
Partie 1

S. Bella
Bella.salyma@gmail.com

L3 – Systèmes Informatiques

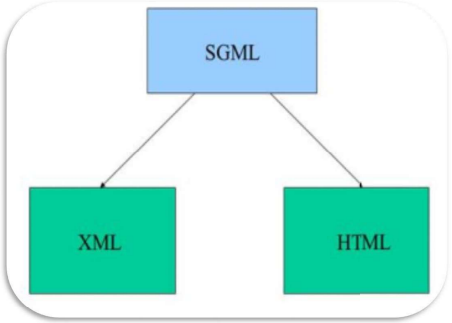
2021/2022

Plan

Introduction à XML

Origine de XML (1/6)

Le langage XML dérive de **SGML (Standard Generalized Markup Language)** et de **HTML (HyperText Markup Language)**.



Origine de XML (2/6)

SGML = Standard Generalized Markup Language, 1986.

- Langage normalisé pour la **génération de langages de balises,**
- Conçu pour des documentations techniques de **grande ampleur**,
- **Riche en sémantique** mais relativement **lourd à mettre en œuvre et inadapté au traitement des documents pour le Web,**
- Une **grande complexité** a freiné son utilisation en dehors des projets de grande envergure,
- Application SGML connue: **HTML**.

Origine de XML (3/6)

HTML = HyperText Markup Language, 1991.

- Langage **simple** de balises comme SGML,
- Balises sont normalisées d'un **ensemble figé** (stable) de symboles encadrés par "<" et ">",
- Balises définissent des **directives de mise en page** d'un texte encadré par un couple balise ouvrante, balise fermante: <H2> xxxx </H2>,
- Langage **standard** reconnu par tous les navigateurs Web.

Origine de XML (4/6)

Limites HTML:

- **HTML est un langage non extensible :** le nombre et la signification des balises sont **limités**. Ex: Il n'existe pas de balisage pour la représentation des données en chimie (molécules formules, valeurs numériques,…).
- **Langage HTML est figé :** la norme HTML fixe les balises pouvant apparaître dans un document du que les constructeurs **ne peuvent pas ajouter d'autres balises**.
- **HTML définit en fait un univers de documents plats :** une recherche doit considérer un document HTML comme **une chaîne de caractères**. Pas de moyen de **partager** entre les communicants une structure de document préétablie.

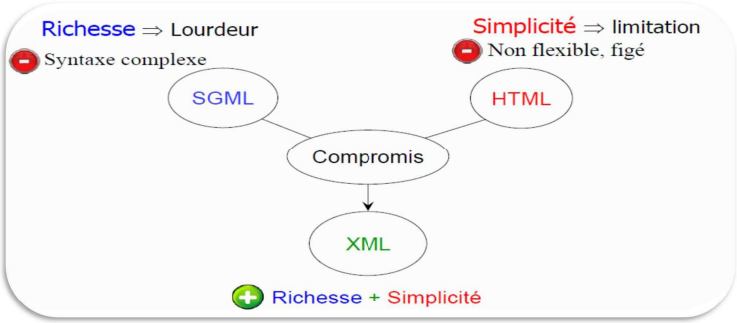
Origine de XML (5/6)

Limites HTML:

- **Langage HTML de description de documents non structurés :** HTML permet de définir de façon beaucoup trop **limitée** la structure d'un document. Il n'y a en fait pas de **vérification** d'une structure pour le document que l'on peut définir. Ex: on peut créer un document commençant par une tête de chapitre H2 et poursuivant par une tête de chapitre H1.
- **Non séparation du document et de sa présentation graphique** : HTML a été et reste un succès fantastique conçu pour afficher du texte dans un browser Web.
- **Inadapté à l'échange de données entre les applications.**

Origine de XML (6/6)

Le W3C (**W**orld **W**ide **W**eb **C**onsortium) a permis de définir un langage qui ait la facilité de **mise en œuvre de HTML** tout en offrant la **richesse sémantique de SGML**. C'est la raison d'être de XML.



W3C (communauté internationale, 1994): développer les **standards du web** (HTML5, XML, CSS, PNG,…), assurer **l'interopérabilité** sur le Web entre les différents systèmes, etc. Il compte 452 membres industriels ou académiques (Adobe, Amazon, Facebook, Microsoft, ...) au 01/10/2021.

XML (eXtensible Markup Language)

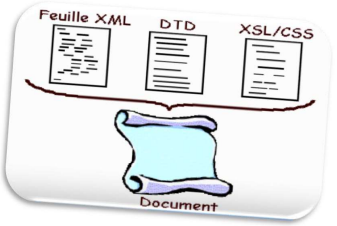
- **XML** est une spécification proposée par la communauté internationale **W3C** en 1998 et qui s'est imposé comme un **standard incontournable**.
- Un langage **orienté texte** formé de **balises** qui permet **d'écrire** et **organiser** les données de manière **structurée** et **non pas leurs affichages**.
- Le but de XML est de **faciliter le traitement automatisé** de documents et de données. L'idée est de pouvoir structurer les informations de telle manière qu'elles puissent être à la fois lues par des personnes sur le web et traitées par des applications qui exploiteront les informations en question.

Caractéristiques du XML

- XML utilise un **langage humain** et non informatique. Donc, c'est un langage **lisible** et **compréhensible**,
- Xml est un format **standardisé** ouvert ne nécessitant aucune licence (open source), intégralement **basé texte** et qui peut être associé à n'importe quel jeu de caractères,
- XML très utilisé soit pour le **stockage de document**, pour **l'échange** et le **partage de données** entre applications. L'exemple le plus connu est le format LibreOffice qui permet d'ouvrir et d'enregistrer dans les types de fichiers Microsoft courants tels que .docx, .xlsx et .pptx,…
- XML permet de gérer le problème d'**interopérabilité** entre les applications **hétérogènes**.

Caractéristiques du XML

- XML permet de **séparer strictement entre le contenu et la présentation** (feuille de style). Cette séparation fait d'un document XML un objet **portable** et **réutilisable** par un nombre illimité d'applications.
- XML est **simple, universel, extensible et méta langages** : permet de définir **des propres balises et attributs** adapté à un domaine donné.
- Document XML peut être **validé par des règles strictes**, contenues par des **modèles de documents** (DTD ou des Schémas), décrivant sa structure et la hiérarchisation de ses données. Sans ces règles, il n'est pas possible d'échanger et de traiter de manière automatique ces documents.



Caractéristiques du XML

- XML permet de définir **une structuration forte du document**

```
<html>
  <head>
     <title>
        Livre
     </title>
  </head>
  <body>
     'XML : Cours et exercices' par  Alexandre
Brillant. Editions EYROLLES, 2007, 299 pages.
  </body>
</html>
```

```
<?xml version="1.0" encoding ="ISO 8859 1"?>
<livre>
     <titre> XML : Cours et exercices </titre>
     <auteur>
        <nom> Brillant </nom>
        <prenom> Alexandre </prenom>
     </auteur>
     <editeur> EYROLLES </editeur>
     <annee>  2007    </annee>
     <taille> 299 </taille>
</livre>
```

Le contenu en HTML est vu comme un texte **(chaine de caractères)**, les balises (<title>,<body> …) sont des éléments **visuels destinés à l'affichage**. Contrairement, en XML les données sont bien structurées (détaillées), les balises (<livre> ,<auteur> …) sont des **éléments syntaxiques** destinés à **structurer le contenu**.

Principaux langages apparentés

- **Xlink** pour ajouter des **liens hypertextes** entre des éléments d'un document XML.
- **Xpointer** pour **pointer** sur des éléments de données d'un document XML.
- **XPath** pour **sélectionner** des éléments dans un document XML.
- **XQuery** pour **extraire** des données des documents XML et de synthétiser de nouvelles données à partir de celles extraites (joue le rôle de SQL).
- **XSD (Schémas XML)** remplace le **DTD (Document Type Declaration)** pour **décrire des modèles de documents.**
- **XSLT** pour **transformer** facilement des documents XML en d'autres formats (PDF, HTML, CSV,…) → la définition de feuilles de style propre à XML.

Structure XML de base

Structure d'un document XML

Un document XML est **correct**:

- Bien **formé** (correcte **syntaxiquement**), mais pas nécessairement valide.
- **Valide** (correcte **structurellement**); le document doit respecter le modèle d'organisation (DTD/XSD).

Un document XML est généralement contenu dans un fichier texte dont l'extension est **'.xml'.**

Un document XML est structuré en 3 parties:

- Un **prologue** contient des déclarations facultatives,
- Un **corps du document** est constitué de son contenu et organisé de façon hiérarchique représenté par un **arbre d'éléments**,
- **Des commentaires et instructions de traitement** sont librement insérés avant, après et à l'intérieur du prologue et du corps.

Structure d'un document XML

```
<?xml version="1.0"  encoding="iso-8859-1" standalone="yes"?>
<!-- Time-stamp: "bibliography.xml 3 Mar 2008 16:24:04" -->
<!DOCTYPE bibliography SYSTEM "bibliography.dtd">
<bibliography>
<livre key="Michard01" lang="fr"> <!-- Livre 1 -->
      <titre>XML langage et applications</titre>
      <auteur>Alain Michard</auteur>
      <annee>2001</annee>
      <editeur>Eyrolles</ editeur>
      <isbn>2-212-09206-7</isbn>
      <url>http://www.editions-eyrolles/livres/michard/</url>
  </livre>
  <livre key="Zeldman03" lang="en"> <!-- Livre 2 -->
     <titre>Designing with web standards</titre>
     <auteur>Jeffrey Zeldman</auteur>
     <annee>2003</annee>
     <editeur>New Riders</editeur>
     <isbn>0-7357-1201-8</isbn>
  </livre>
</bibliography>
```

Prologue

Corps

Prologue

- Le prologue contient **deux déclarations facultatives** mais fortement conseillées ainsi que des commentaires et des instructions de traitement.
- **Déclaration 1:** L'entête XML (la ligne d'introduction du document)

<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>

- **version XML (**obligatoire): soit 1.0 ou 1.1 (plus utilisé 1.0 et par défaut).
- **encoding** (facultatif, UTF-8 par défaut): jeu de caractères utilisé pour le codage du document (Unicode, ISO).
- **standalone** (facultatif) : désigne l'indépendance du document (yes: libre, no: dépend d'autres déclarations externes pour former la forme finale).

Prologue

- Le prologue contient deux déclarations facultatives mais fortement conseillées ainsi que des commentaires et des instructions de traitement.
- **Déclaration 2:** Déclaration de type de document (DTD)

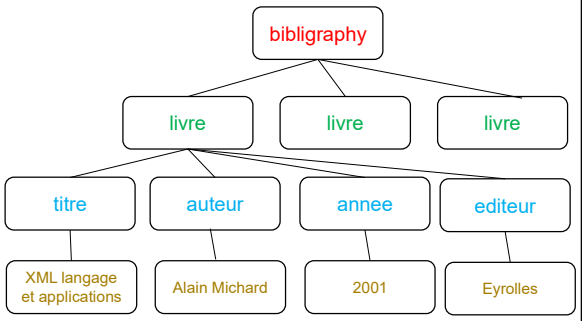
<!DOCTYPE bibliography SYSTEM "bibliography.dtd">

- **<!DOCTYPE…>:** cette déclaration optionnelle sert à attacher une grammaire de type DTD (Document Type Definition) à un document XML. Elle est introduite avant la première balise (racine) du document (voir cours validation par DTD).

Arbre d'éléments

- Les documents écrits avec XML ont une structure arborescente **d'éléments** dont les nœuds feuilles de l'arbre contiennent des **données** (texte simple, d'autres éléments ou un mélange des deux).
- L'arbre d'éléments contient un élément **racine unique** et nœuds **éléments;** en plus les attributs et les entités, les textes.

```
<bibliography>
  <livre key="Michard01" lang="fr"> <!-- Livre 1 -->
     <titre>XML langage et applications</titre>
     <auteur>Alain Michard</auteur>
     <annee>2001</annee>
     <editeur>Eyrolles</editeur>
  </livre>
  <livre key="Zeldman03" lang="en"> <!-- Livre 2 -->
  ……
  </livre>
  <livre key="Zeldman03" lang="en"> <!-- Livre 3-->
  ……
  </livre>
</bibliography>
```

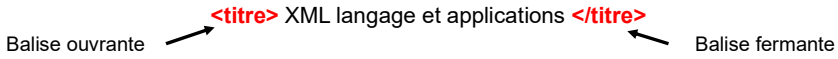


Elément racine

- L'élément racine est obligatoire. Il est le premier élément déclaré dans un document XML; il est **unique** et englobe tous les autres éléments.
- L'élément *bibliography* est l'élément racine de l'exemple donné.

Eléments

- Les éléments (tags, nœuds) gèrent la **structuration des données** d'un document XML. Ils sont les **branches et les feuilles** de l'arborescence.
- Ils constituent la majorité du contenu d'un document XML.
- Tout document a **un et un seul élément racine**.
- Un élément XML se compose d'une **balise ouvrante**, d'un contenu et d'une **balise fermante**.



Eléments

Un élément peut être:

- **Vide** (pas de contenu):

```
<couverture/>
```

- Contient du **texte**:

```
<auteur>Alain Michard</auteur>
```

- Peut avoir plusieurs **éléments fils** (enfants) : (l'imbrication des balises doit être **correcte**)

```
<livre>
  <titre>XML langage et applications</titre>
  <auteur>Alain Michard</auteur>
  <annee>2001</annee>
</livre>
```


Attributs

- Les attributs **caractérisent** les éléments.
- Tous les éléments peuvent contenir **un ou plusieurs attributs**.
- **L'ordre** des attributs **n'a pas d'importance** au sein d'un élément.
- Chaque élément ne peut contenir qu'**une fois le même attribut**.
- Un attribut est composé d'un **nom** et d'une **valeur** entre **guillemets** (" ou "").
- Il ne peut être présent que dans la balise **ouvrante** de l'élément.
- Exemple d'utilisation d'un attribut dans un élément:

```
<auteur nationalite="USA"> Alain Michard</auteur>
```

- Exemple d'utilisation d'un élément vide avec attributs:

```
<img src="ours.gif" width="56" height="100"/>
```

Entités

- XML propose une représentation appelée **référence d'entité** ou **entité**, pour que les caractères ne peuvent pas poser des problèmes à l'affichage.
- La déclaration des entités **s'effectue au sein de la DTD**. Elles peuvent être utilisées aussi bien dans la DTD que dans le document XML.
- L'appel d'une entité dans un document : **&nom_entite;**
- Tous les caractères peuvent être remplacés par une **entité numérique** à travers un **code décimal &#code_car;** (par ex. A pour le A, le caractère *é* pour é).
- Certains caractères, lorsqu'ils sont insérés dans un document XML, doivent être remplacés par des **entités prédéfinies** (les caractères spéciaux) pour qu'ils puissent être affichés.
- Les entités prédéfinies sont:

Caractère	Entité
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;

Texte

- Les textes font partie du **contenu des éléments** et sont vus comme des nœuds enfants. Il faut bien comprendre que la **totalité du texte** situé entre les balises, y compris **les espaces et retour à la ligne** font partie du texte.

```
<livre key="Michard01" lang="fr"> <!-- Livre 1 -->
    C'est le premier livre

    <titre>XML langage et applications</titre>
    <auteur>Alain Michard</auteur>
    <annee>2001</annee>
</livre>
```


Commentaires

- Ils commencent par les chaînes de caractères **<!--** et **-->** comme en HTML.
- Ils ne peuvent pas contenir la chaîne **--** formée de deux tirets **-** et ils ne peuvent donc pas être imbriqués.
- Ils peuvent être placés à n'importe quel endroit tant qu'ils se trouvent à l'extérieur d'une balise.
- Ils peuvent figurer sur plusieurs lignes.
- Exemple:

```
<livre key="Michard01" lang="fr">
<!-- Livre 1 -->
<!-- Livre 1 -->
    <titre> <!-- Livre 1 --> XML langage et applications</titre>
    <auteur>Alain Michard</auteur>
    <annee>2001</annee>
</livre>
```


Instructions de traitement

(processing instruction ou PI)

- Elles sont destinées aux applications qui traitent les documents XML. Elles servent à donner à l'application qui utilise le document XML des informations.
- Elles commencent par **<?** et se terminent par **?>** .
- On les positionne **à n'importe quel endroit du document** (après le prologue, bien entendu).
- La plus utilisée de ces instructions est celle constituant le prologue d'un document XML :

```
<?xml version="1.0"?>
```

Section CDATA

Character Data (données caractères)

- C'est une section pouvant contenir toute sorte de **chaîne de caractères**.
- Le texte contenu dans la section CDATA n'est pas **parsé** (analysé par le processeur XML) et ses caractères de balisage ignorés.
- Ceci permet entre autres de garder dans un bloc de texte un exemple de code à afficher tel quel.
- Pas besoin de recourir à des entités pour afficher les caractères réservés de XML.
- Exemple d'utilisation de CDATA:

<![CDATA[Une balise commence par un < et se termine par un >.**]]>**

Quelques règles de syntaxe

Ces règles de syntaxe sont à respecter impérativement pour qu'un document XML soit **bien formé**:

- Le nom d'un élément ne peut commencer par un chiffre :
  - Si le nom n'est composé que d'un seul caractère, ce doit être une lettre comprise entre 'a' et 'z' pour les minuscules, 'A' et 'Z' pour les majuscules. Ex: <a>, <B>,…
  - S'il est composé d'au moins deux caractères, le premier peut être '_', '-' ou ':' plus les caractères alphanumériques. Ex: <nom_prenom>, <prenom:fille45>,…
- Toutes les balises portant un **contenu non vide** doivent être **fermées**.
- Les balises **n'ayant pas de contenu** doivent se terminer par **/>**.
- Les valeurs d'attributs doivent être entre **guillemets**.
- Les éléments doivent être **correctement imbriqués**.
- L'élément racine doit être **unique**.

Document bien formé

- Un document respectant les règles de XML citées précédemment est appelé **document bien formé** («*well-formed*»).
- Seuls les documents "bien formés" seront affichés correctement par un navigateur web.
- A la moindre erreur de syntaxe, **le document ne sera pas ou ne sera que partiellement affiché**.

```
<list>
  <item> Voiture </item>
  <ITEM> Avion </ITEM>
  <Item> Train </Item>
</list>
```

Bien formé

```
<list>
  <item> Voiture </itm>
  <item> Avion </ITEM>
  <Item> Train </Item>
</list>
```

Mal formé

```
<text>
  <bold><italic> XML </bold></italic>
</text>
```

Mal formé

```
<text>
  <bold><italic> XML </italic> </bold>
</text>
```


Merci pour votre Attention

Vos Questions !!