

Développement d'Applications Web DAW

Langages de programmation
pour le Web:

XML

Présenté par :

Me Derkaoui Orkia

Université Docteur Moulay TAHAR de Saida

derkaouiorkia@gmail.com

Plan du cours

Chapitre 2 : Langages de programmation pour le Web

4 XML

4.1 Structure d'un document XML

4.2 DTD (Document Type Definition)

4.3 XML Schéma

4.4 XSLT

Du HTML vers le XML [1]

Du HTML vers le XML :

- XML (eXtensible Markup Language)
- XML = langage HTML amélioré permettant de définir de nouvelles balises HTML.
- XML : Définir de nouveaux domaines de données.
- Séparer le contenu de la présentation (plus de liberté de présentation)

Pourquoi utiliser XML ? [5]

La communauté World Wide Web Consortium (W3C):

- URL: <http://www.w3.org>
- 400 partenaires industriels, parmi lesquels les plus grand comme Oracle, IBM, Compaq, Xerox, Microsoft
- Laboratoires de recherche: MIT pour les États Unis, INRIA pour l'Europe, université Keio (Japon) pour l'Asie

XML : recommandation W3C pour

- les documents Web (généralisation de HTML),
- mais aussi pour l'échange, la transformation,
- l'intégration et l'interrogation des données sur le Web.

XML : pourquoi je l'utilise ? [5]

➤ **SGML : Standard Generalized Markup Language**

Trop compliqué, inadapté aux besoins du WWW .

➤ **HTML : HyperText Markup Language**

Simple mais ayant beaucoup de problèmes .

➤ **XML : eXtensible Markup Language (1997)**

Adaptation de SGML aux besoins du World Wide Web.

Objectifs :

➤ Séparation des données et de leur mise en forme (XML, XSL (eXtensible Style Language)).

➤ Séparation des informations structurelles et des données (DTD (Document Type Definition)).

Principe de XML

Le Principe clé de XML est de

**séparer la structure d'un document de sa
présentation.**

Historique d'XML

Descendant de SGML (ISO 8879, 1986)

SGML = Standard Generalized Markup Language

- Invention : 1970's (avant le WWW)
- Spécification : 150 pages
- Intègre tous les langages de «markup» (très puissant)
- Application SGML connue : HTML
- SGML est très complexe (pas d'implémentation unifiée)

XML = version «légère» de SGML

- Début : 1996
- Normalisation par le W3C : 02/1998
- Version : 1.0 (jusqu'à maintenant)

Les avantages de XML

- **La lisibilité** : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- **Autodescriptif et extensible** : Une structure arborescente permettant de modéliser la majorité des problèmes informatiques
- **salité et portabilité** : les différents jeux de caractères sont pris en compte
- **Déployable** : il peut être facilement distribué par n'importe quels protocoles capable de transporter du texte, comme HTTP
- **Intégrabilité** : un document XML est utilisable par toute application pourvue d'un parseur (c'est-à-dire un logiciel permettant d'analyser un code XML)
- **Extensibilité** : un document XML doit pouvoir être utilisable dans tous les domaines d'applications

Les avantages de XML

- Indépendance entre les outils de présentation (browser) et les outils de gestion de l'information.
- Différentes présentations sont possibles pour le même document (transformation) .
- Interrogation (semi-)structurée de documents.

L'extensibilité de XML

L'extensibilité de XML

L'intérêt de disposer d'un format commun d'échange d'information dépend du contexte professionnel dans lequel les utilisateurs interviennent :

VoiceXML : Description des services vocaux interactifs.

CCXML : *Call Control eXtensible Markup Language* - Programmation de SERVEUR téléphonique.

HTML : *Hyper Text Marckup Language* : Pour la description du contenu des pages web.

CML : *Chemical Markup Language* - Permet de décrire des composés chimiques.

SMIL : *Synchronized Multimedia Integration Language* - Permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte,...

SOAP : *Simple Object Access Protocol* “ Description des Services Web

Structure d'un document XML

La **norme XML** en tant que telle doit être vue comme un outil permettant de définir un langage (on dit alors qu'il s'agit d'un **métalangage**),

Une balise est une chaîne de caractère du type: <balise>

Exemple :

<annuaire>

<personne class = "etudiant">

<nom> Ajal</nom>

<prenom>Ahmed</prenom>

<telephone>(07)77 23 43 43 </telephone>

<email>webmaster;/email>

<!-- insertion de commentaires XML -->

</personne>

</annuaire>

Structure d'un document XML

Structure d'un document XML

Un document XML est structuré en 3 parties:

- **La première partie**, appelée *prologue* : permet d'indiquer la version de la norme XML utilisée pour créer le document ainsi que le jeu de caractères utilisé dans le document :

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

- **Le second élément** est une déclaration de type de document (déclaration de la DTD ou du Schéma, de la feuille de style)(optionnel)
- **Et enfin la dernière composante** d'un fichier XML est l'arbre des éléments.

Structure d'un document XML

La syntaxe des éléments en XML

-Encapsulation dans les balises:

<balise> contenu de la balise </balise>

-Utilisation des attributs :

<balise attribut_01 = "text" attribut_02= "text" >

-Pas de chevauchement de balises :

<balise_01>

<balise_02>

</balise_01>

</balise_02>

- Balise Vide

<balise> </balise> = <balise/>

Éléments de base en XML

Éléments de base sont :

DOM et Parseur

➤ **Parseur** : programme permettant le passage de la forme sérialisée à la représentation arborescente de DOM .

➤ **DOM** (**D**ocument **O**bject **M**odel) : Un arbre, constitué de nœuds typés (éléments, commentaires, valeurs, etc), Document a une forme arborescente.

Éléments de base en XML

➤ **Le document sérialisé** est analysé, et une représentation arborescente est créée :

- ❑ le nœud racine est de type Document

- ❑ les catégories syntaxiques (commentaires, balises, texte) se traduisent par différents types de noeuds (Comment, Element, Text, ...)

- ❑ les nœuds constituent un arbre qui reflète l'imbrication des éléments dans la forme sérialisée

Exemple [5]

➤ Exemple : représentation sérialisée (textuelle)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CINEMA>
  <NOM>
    Epée de Bois
  </NOM>
  <ADRESSE>
    10, rue du Cinéma
  </ADRESSE>
  <BUS>
    101:Fabron-Valrose
  </BUS>
</CINEMA>
```

Exemple [5]

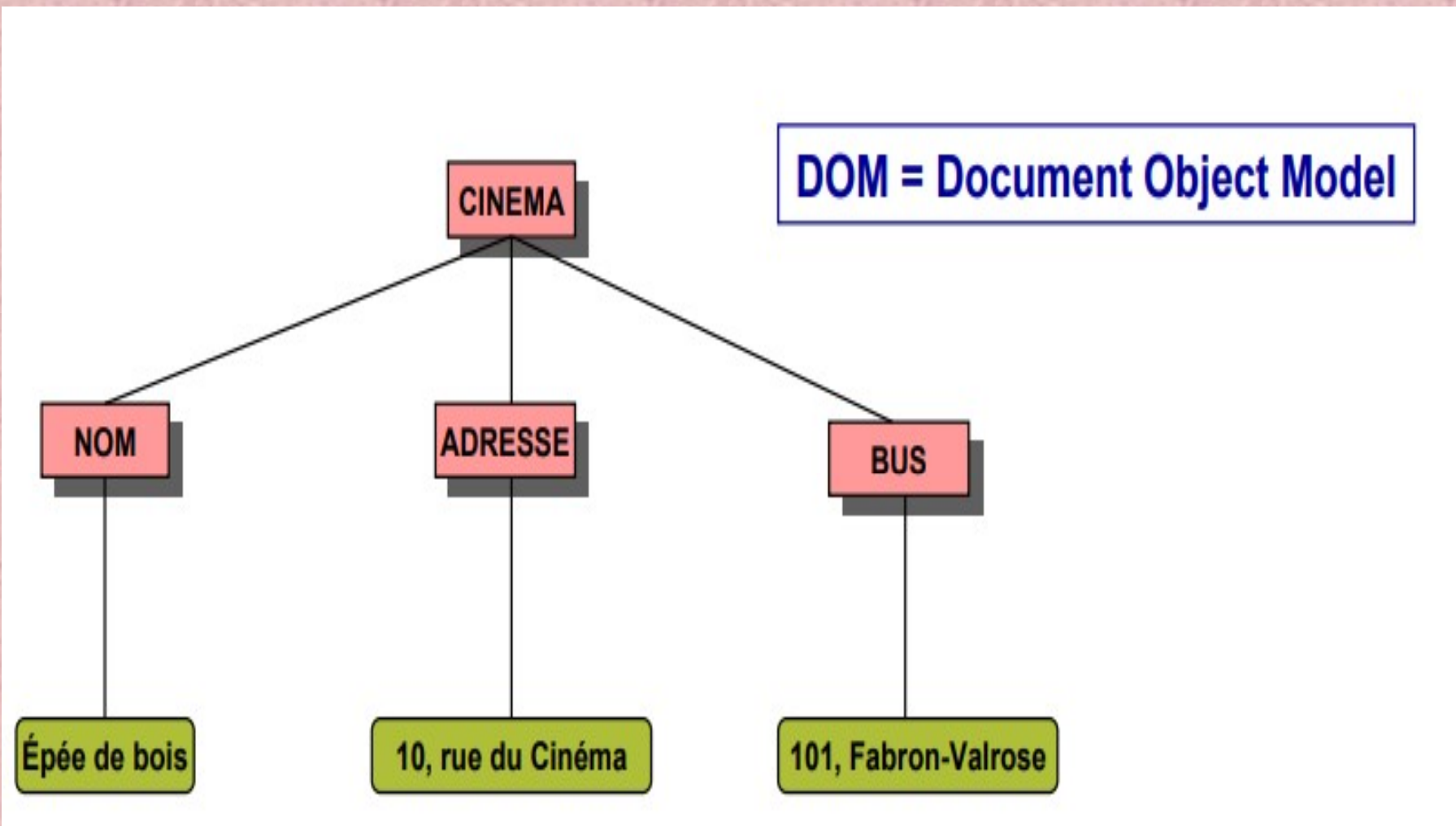


Image empruntée à [5]

Exemple [5]

➤ Exemple : représentation sérialisée (textuelle)

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<!-- Commentaire -->  
<A>Le texte de A  
<B>Le texte de B</B>  
<D attr1="1" attr2="azerty">  
<C/>  
</D>  
<![CDATA[2x < y]]>  
</A>
```

Exemple [5]

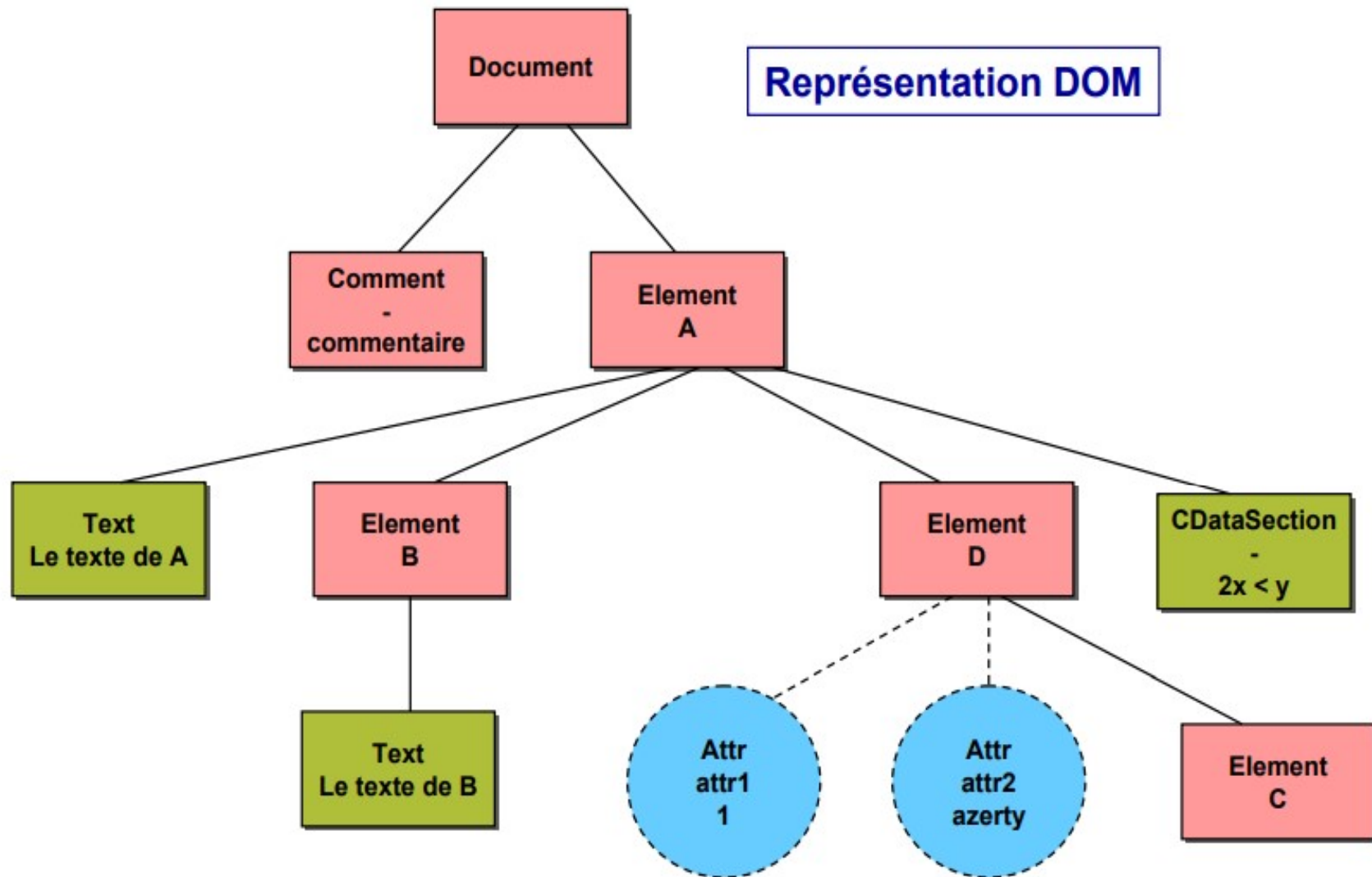


Image empruntée à [5]

XML : Une solution d'échange

La nouvelle tendance : Exploitation des réseaux et développement d'application réparties, (efficacité, réutilisation et accessibilité)

XML : Solution pour assurer l'échange de données entre systèmes distribués.

3 concurrents à XML :

- COM/DCOM (Component Object Model / Distributed COM)
- CORBA (Common Object Request Broker Architecture)
- Java RMI (Remote Method Invocation)

XML : Une solution d'échange (comparatif)

Validation des documents XML

Validation des documents XML

XML fournit un moyen de vérifier la syntaxe d'un document :

- Utilisation des **XML Schema** ou
- des **DTD (*Document Type Definition*)**
- pour décrire la structure des documents avec les imbrications des éléments possibles
- ***Document bien formé*** : Un document suivant les règles de XML.
- ***Document valide*** : Un document XML possédant une DTD (ou XML Schema) et étant conforme à celle-ci.

Références bibliographiques

[1] <https://www.cours-gratuit.com/cours-xml/cours-complet-pour-apprendre-le-xml>

[2] <https://www.toutjavascript.com/savoir/savoir12.php3#:~:text=Comme%20en%20C%2C%20les%20instructions,ouvertes%20que%20d'accolades%20ferm%C3%A9es.&text=Cette%20fonction%20inutile%20retourne%20la,fonction%20est%20entour%C3%A9%20d'accolades>

[2] [https://developer.mozilla.org/fr/docs/Learn/Getting started with the web/JavaScript basics](https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics)

[3] <https://www.chiny.me/comment-integrer-du-code-javascript-6-2.php>

[4] [https://apical.xyz/fiches/particularites de la programmation web 002/Cote client vs cote serveur](https://apical.xyz/fiches/particularites_de_la_programmation_web_002/Cote_client_vs_cote_serveur)

[5] [https://www.i3s.unice.fr/~nlt/cours/licence/xml/s1 xml slides.pdf](https://www.i3s.unice.fr/~nlt/cours/licence/xml/s1_xml_slides.pdf)