

# XML et DOM

Matériel de cours



mars 1999

version 0.3

dernière modification: 24/3/99

<http://tecfa.unige.ch/guides/xml/slides/xml.html> - (moche!)  
<http://tecfa.unige.ch/guides/xml/slides/xml.pdf> - (lecture on-line)  
<http://tecfa.unige.ch/guides/xml/slides/xml2.ps> - (2 pages sur une)

COPYRIGHT: Daniel Schneider

TECFA,

Faculté de Psychologie et des Sciences de l'Education,  
Université de Genève

9 route de Drize, CH-1227 Carouge

Tél.: +41 22 70 9694

Email: Daniel.Schneider@tecfa.unige.ch

MOO: tecfamoo.unige.ch 7777 (aka Daniel)

WWW: <http://tecfa.unige.ch/tecfa-people/schneider.html>

## **Usage**

- **Matériel pour l'enseignements de XML**
- **Attention: qualité brouillon parfois**
- **Voir aussi:**
  - <http://tecfa.unige.ch/guides/xml/pointers.html>

## Modules

### **I: XML et DOM: Les bases .....7**

---

- |    |   |           |
|----|---|-----------|
| 1. | <b>Vers un nouveau paradigme Internet</b> | <b>8</b>  |
| 2. | <b>XML dans la pratique</b>               | <b>18</b> |
| 3. | <b>Le Document Objet Model (DOM)</b>      | <b>20</b> |
| 4. | <b>Introduction technique à XML</b>       | <b>24</b> |
| 5. | <b>Les DTD</b>                            | <b>29</b> |

### **II: XML: En pratique .....37**

---

- |    |                           |           |
|----|---------------------------|-----------|
| 1. | <b>UNDER CONSTRUCTION</b> | <b>38</b> |
|----|---------------------------|-----------|

## Table des matières

<b>I: XML et DOM: Les bases</b> .....	<b>7</b>
<b>1. Vers un nouveau paradigme Internet</b>	<b>8</b>
A. Buts de cet exposé .....	8
B. Pointeurs supplémentaires: .....	8
C. Historique des langages “markup” .....	8
<b>1.1 Le problème HTML et la solution XML</b> ....	<b>9</b>
A. Overview .....	9
B. On veut beaucoup maintenant, par exemple: .....	9
C. Problèmes avec HTML .....	10
D. Avec PDF .....	10
E. Avec Word, Framemaker, etc. ....	10
F. Résumé .....	10
<b>1.2 W3C Data Formats</b> .....	<b>11</b>
A. Eléments les plus importants du framework de données: .....	11
B. Petite explication des sigles .....	12
C. Fonctionnalités de base d’un système d’information .....	13
D. Applications XML (voir surtout page suivante) .....	14
E. Quelques applications XML du W3C (consortium WWW) .....	15
<b>1.3 XLL - Vers un meilleur hypertexte</b> .....	<b>16</b>
A. 2 Composantes: XLink et Xpointer .....	16
B. Caractéristiques principales: .....	16
C. Héritage .....	16
<b>1.4 XML avec Style</b> .....	<b>17</b>
A. XSL: Extensible Stylesheet Language .....	17
B. CSS (Cascading Style Sheets) .....	17
<b>2. XML dans la pratique</b>	<b>18</b>
<b>2.1 XML Authoring</b> .....	<b>18</b>
A. Outils auteurs .....	18
B. Filtres XML -> HTML .....	18
C. Filtres XXX -> XML .....	18
<b>2.2 XML sur le Web</b> .....	<b>19</b>
A. Browsers WWW .....	19
B. Applets Java .....	19
C. Server-Side .....	19
<b>2.3 XML ailleurs</b> .....	<b>19</b>
<b>3. Le Document Objet Model (DOM)</b>	<b>20</b>
<b>3.1 Le principe du DOM avec un exemple</b> .....	<b>21</b>
<b>3.2 Conclusion (provisoire)</b> .....	<b>23</b>

<b>4.</b>	<b>Introduction technique à XML</b>	<b>24</b>
<b>4.1</b>	<b>Les notions de well-formed et valid .....</b>	<b>25</b>
A.	"Well-formed" documents .....	25
B.	"Valid documents" .....	25
C.	La définition de XML .....	26
<b>4.2</b>	<b>Exemples .....</b>	<b>27</b>
<b>5.</b>	<b>Les DTD</b>	<b>29</b>
<b>5.1</b>	<b>Introduction aux DTD de XML .....</b>	<b>29</b>
A.	HTML avec XML ? .....	29
<b>5.2</b>	<b>Le langage pour définir une grammaire DTD .....</b>	<b>30</b>
A.	Prologue et déclaration de DTD .....	30
B.	Déclaration d'éléments .....	31
C.	Déclaration d'attributs .....	34
D.	Déclaration d'Entities .....	36
<b>II:</b>	<b>XML: En pratique .....</b>	<b>37</b>
<b>1.</b>	<b>UNDER CONSTRUCTION</b>	<b>38</b>
<b>1.1</b>	<b>Autres ressources pour l'enseignement à TECFA .....</b>	<b>38</b>

## Liste des exemples

### **I: XML et DOM: Les bases .....7**

---

- Exemple 3.1: Un simple table HTML comme “DOM tree” 21
- Exemple 4.1: Quelques éléments de XML à titre d’illustration 26
- Exemple 4.2: Une veille blague en XML 27
- Exemple 4.3: Une recette en XML 28
- Exemple 5.1: Hello XML sans DTD 30
- Exemple 5.2: Hello XML avec DTD 30
- Exemple 5.3: Un DTD pour un simple Address Book 32
- Exemple 5.4: Un DTD pour une recette 33
- Exemple 5.5: Un DTD pour un moins simple Address Book 35

### **II: XML: En pratique .....37**

---

# I: XML et DOM: Les bases



Version 0.3 (mars 1999)  
Dernière modification: 24/3/99

<http://tecfa.unige.ch/guides/xml/slides/xml.html>  
<http://tecfa.unige.ch/guides/xml/slides/xml.pdf>

COPYRIGHT: Daniel Schneider

TECFA,  
Faculté de Psychologie et des Sciences de l'Education,  
Université de Genève

9 route de Drize, CH-1227 Carouge  
Tél.: +41 22 70 9694

Email: [Daniel.Schneider@tecfa.unige.ch](mailto:Daniel.Schneider@tecfa.unige.ch)  
MOO: [tecfamoo.unige.ch 7777](http://tecfamoo.unige.ch/7777) (aka Daniel)  
WWW: <http://tecfa.unige.ch/tecfa-people/schneider.html>

## 1.. Vers un nouveau paradigme Internet

.

# 1. Vers un nouveau paradigme Internet

---

## A. Buts de cet exposé

- Donner un “feeling” pour XML et le DOM “Framework”
- Montrer la place de ce nouveau framework par rapport aux autres outils (HTML, bases de données, etc.)
- Faire un “état des lieux”

## B. Pointeurs supplémentaires:

**Quelques indexes qui vous amèneront plus loin:**

- **XML@Tecfa:** <http://tecfa.unige.ch/guides/xml/pointers.html>
- **DOM@Tecfa:** <http://tecfa.unige.ch/guides/dom/pointers.html>
- **RDF @Tecfa:** <http://tecfa.unige.ch/guides/rdf/pointers.html>
- **Voir le WDWL Acronym Expander pour l’explication de la plupart des sigles:** <http://wdvl.internet.com/Authoring/Languages/XML/Overview/acronym.html>

## C. Historique des langages “markup”

**SGML (Standard Generalized Markup Language**

- ISO standard en 1986)

**HTML (application SGML + liens en 1990)**

- une application SGML avec très peu de balises (tags) fixes

**XML (1997, -)**

- un langage plus léger que SGML adapté au Web

1.. Vers un nouveau paradigme Internet

1.1. Le problème HTML et la solution XML ....

1.1 Le problème HTML et la solution XML ....

---

A. Overview

**Le monde HTML**

incompatibilités  
manque de flexibilité  
beaucoup de faiblesses  
manque de portée

facile .....  
beaucoup d'outils

**Le monde XML**

standardisation  
extensibilité  
faiblesses (?)  
ouvert

difficile .....  
peu d'outils



B. On veut beaucoup maintenant, par exemple:

- **structurer de l'information**
  - .... et la retrouver facilement
  - .... et l'imbriquer facilement dans des applications
- **faire des hypertextes puissants**
- **afficher et imprimer de façon flexible et jolie**
- **un format universel pour toute sortes de données et usages**
  - diffuser/échanger/stocker/chercher/..... pas juste afficher
- **un meta-langage qui permet de créer des langages variés, .... adaptés aux besoins, mais "propres"**

## 1.. Vers un nouveau paradigme Internet

### 1.1. Le problème HTML et la solution XML ....

#### C. Problèmes avec HTML

- **HTML (seul type de document “WWW” universel) n’est pas flexible, pas de “customisation” possible**
- **HTML est incompatible (trop de versions)**
- **HTML est faible pour décrire le contenu d’information**
  - essentiellement un langage pour structurer et présenter un “texte”
  - ne permet pas d’exprimer des hiérarchies et relations entre données
- **HTML est faible pour l’hypertexte**
  - Les pages HTML sont isolées
  - pas de liens bi-directionnels
  - pas de “fan-out”
  - pas de inclusion, .....
- **HTML est fait pour être affiché dans un browser**
  - pas pour échanger de l’information entre programmes

#### D. Avec PDF

- **C’est un format trop orienté vers le display**
- **Difficilement “lisible” et manipulable**

#### E. Avec Word, Framemaker, etc.

- **On est prisonnier d’un format propriétaire (qui change selon les vœux des fabricants)**
- **Absence de “features” typiques de HTML**
- **Formats aussi difficiles (RTF, MIF, etc) que PDF.**
- **Pas utile comme format de représentation de données**

#### F. Résumé

**Du ALL-IN-ONE et propre pour ceux qui veulent**

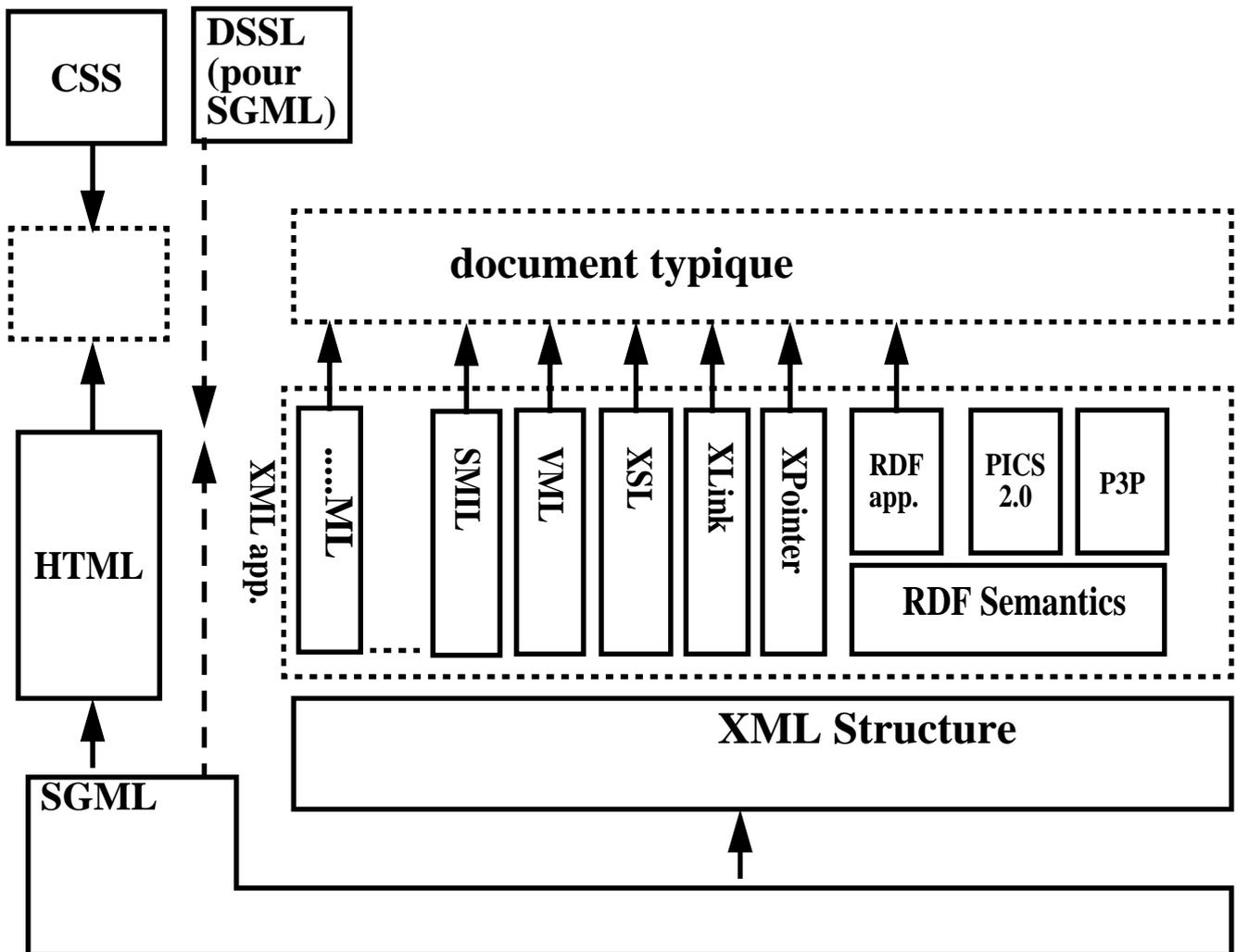
1.. Vers un nouveau paradigme Internet

1.2. W3C Data Formats

## 1.2 W3C Data Formats

- XML est une initiative du consortium WWW (W3C)
- bien accueilli par l'industrie (y compris Microsoft)

A. Eléments les plus importants du framework de données:



**1.. Vers un nouveau paradigme Internet****1.2. W3C Data Formats****B. Petite explication des sigles****Meta-Langages:**

- Un meta-langage est un langage qui permet de définir d'autres langages (appelés souvent "applications" dans ce contexte)
- **SGML: Standardized Generalized Markup Language (ISO 8879)**
- **XML (version simplifiée de SGML): Métalangage comme SGML, mais permet des documents non-valides (comme HTML)**

**Langages SGML**

- pleins (dont XML)
- **HTML (sans qq extensions sauvages): Quelques simple tags pour faire des rapports, inclure un peu de multimédia et un peu d'hypertexte.**

**Langages de Style:**

- **CSS (1/2): Langage de style pour HTML et XML**
- **DSSL: Langage de Style le plus populaire pour SGML (Scheme like)**
- **XSL (application XML): Langage de style pour XML**

## 1.. Vers un nouveau paradigme Internet

## 1.2. W3C Data Formats

C. Fonctionnalités de base d'un système d'information

- **Markup:** Langage pour caractériser des éléments d'information
- **Style:** Langage pour définir la mise en page d'une classe d'objets
- **Linking:** Langage pour représenter des liens entre éléments et objets
- **Scripting:** Interface et langages pour créer des applications

Tableau 1: "The Big Picture"

	<i>monde HTML</i>	<i>monde XML</i>	<i>monde SGML</i>
<i>Linking</i>	(<A> Tag dans HTML)	Xlink & Xpointer	HyTime & TEI
<i>Style</i>	CSS2	XSL (CSS)	DSSL
	CSS1		
<i>Markup</i>	HTML	XML	SGML
<i>Interface entre Markup et Scripting</i>	Document Object Model (DOM)		
<i>Scripting</i>	Javascript		
	JScript ECMAScript		

Inspiré de l'article "Next Generation HTML: The Big Picture"  
(<http://wdvl.internet.com/Authoring/Languages/XML/Overview>)

- **Fonctionnalités additionnelles pour le monde XML**
  - extensibilité (chaque communauté peut créer son langage adapté)
  - structure (le document peut contenir son "modèle d'information", les DTD (Document Type Definitions))
  - validation (on peut contraindre les auteurs à suivre un schéma DTD)

1.. Vers un nouveau paradigme Internet

1.2. W3C Data Formats

#### D. Applications XML (voir surtout page suivante)

- **langage pour représenter la structure d'un type d'information**
  - vous pouvez créer votre propre langage
- **“Vocabulaires”**: XML est un langage assez universel pour la représentation de contenus, exemples:
  - CML (Chemical Markup Language)
  - VRML-Xmlisé
  - NML (News Markup Language) et NITF (News Interchange Text Format)
  - .....
- **XML est un langage pour rajouter d'autre fonctionnalités aux vocabulaires**:
  - hyper-liens
  - graphisme
  - catégorisation
  - échange de données
  - échange de requêtes entre programmes
  - .... etc.
- **Il existe déjà une centaine d'applications XML....**

## 1.. Vers un nouveau paradigme Internet

### 1.2. W3C Data Formats

## E. Quelques applications XML du W3C (consortium WWW)

voir aussi Tecfa XML page: <http://tecfa.unige.ch/guides/xml/pointers.html>

- **XLL: XML hyperlinking**
  - XLink (proper): Hypertext links
  - XPointer
  - Voir section 1.3 “XLL - Vers un meilleur hypertexte” [16]
- **Applications RDF: (par exemple IMS)**
  - voir: <http://tecfa.unige.ch/guides/rdf/pointers.html>
- **PICS 2.0: Platform for Internet Content Selection**
  - <http://www.w3.org/PICS/>
- **SMIL: Synchronized Multimedia Integration Language**
  - <http://www.w3.org/AudioVideo/>
- **P3P: Platform for Privacy Preferences**
  - <http://www.w3.org/P3P/>
- **VML: Vector Markup Language**
- **MathML: Mathematical Markup Language**
  - <http://www.w3.org/Math/>
- **XHTML: (HTML 4.0 en XML)**
  - Tags fermés, pas de croisements, imbrication correcte des éléments

### **Ce n'est pas tout !**

- ... XML est le langage pour formaliser des problèmes de représentation de toutes les sortes.

## 1.. Vers un nouveau paradigme Internet

### 1.3. XLL - Vers un meilleur hypertexte

## 1.3 XLL - Vers un meilleur hypertexte

---

### A. 2 Composantes: XLink et Xpointer

- **Xlink = comment insérer un lien dans un document XML (le lien exprime une relation entre deux ou plusieurs objets)**
  - <http://www.w3.org/TR/NOTE-xlink-req/>
- **XPointer = comment identifier un fragment XML (utilisable par des liens)**
  - <http://www.w3.org/TR/NOTE-xptr-req/>

### B. Caractéristiques principales:

- **Liens multi-directionnels**
- **Liens à multiple destinations**
- **“Inlining” de contenus dans un document**
- **Remplacement de contenus dans un document**
- **Bases de données pour organiser des locations de liens**

### C. Héritage

- **HTML**
  - Ancres: A, LINK, SRC (attribut IMG et NOTE), ISMAP (attribut IMG)
  - Targets: BASE, NAME attribut (A), ID (attribut dans HTML 3)
- **HyTime**
  - standard (ISO 10744) bâti sur SGML. “It provides facilities for representing both static and dynamic information for processing and interchange by hypertext and multimedia applications.”
- **TEI Extended Pointers**
  - une extension à HyTime

## 1.. Vers un nouveau paradigme Internet

### 1.4. XML avec Style

## 1.4 XML avec Style

---

- **Les style-sheets permettent de:**
  - séparer contenu et présentation
  - rationaliser le travail (un style-sheet pour beaucoup de documents)

### A. XSL: Extensible Stylesheet Language

- <http://www.w3.org/TR/WD-xsl>
- **XSL est un langage pour définir des stylesheet pour une classe de documents XML.**
- **Features:**
  - formatage sophistiqué, aussi selon héritage, descendance, position etc.
  - génération de textes et graphiques
  - possibilité de définir des macros
  - tout ce que l'on trouve dans CSS et plus ....
- **... enfin XSL est difficile à comprendre**

### B. CSS (Cascading Style Sheets)

- **CSS1 (obsolète, mais prédominant)**
- **CSS2 (support direct pour XML aussi)**

## 2.. XML dans la pratique

### 2.1. XML Authoring

## 2. XML dans la pratique

---

- **sous développement, le marché est récent, mais en train d’exploser**
- **Voir: <http://www.xmlsoftware.com/>**
- **Tous les grands vendeurs ont des produits (la plupart en alpha, beta)**

### 2.1 XML Authoring

---

#### A. Outils auteurs

**3 genres:**

- **Outils permettant d’éditer un “arbre” (quelques programmes Java gratuits et les éditeurs de programmation comme Emacs)**
- **Outils professionnels (souvent chers, souvent SGML aussi)**
- **Plug-ins pour traitement de texte (à venir)**

#### B. Filtres XML -> HTML

- **Les mêmes qui sont utilisés “server-side”**
  - souvent écrits en Java, nécessitent des connaissances de base en programmation

#### C. Filtres XXX -> XML

- **RTF**
- **HTML**
- **Lotus Notes**
- **.....**

## 2.. XML dans la pratique

### 2.2. XML sur le Web

## 2.2 XML sur le Web

---

### A. Browsers WWW

- **IE Explorer: support XML et CSS et XSL (via Java->HTML)**
- **Mozilla (XML + CSS), annonces pour Netscape 5**

### B. Applets Java

- **Pleins de bibliothèques pour développeurs existent**
- **... et certaines applications aussi**

### C. Server-Side

- **Traducteurs XML->HTML**
  - XML + CSS
  - XML + XSL
  - XML + traduction “ad-hoc”
- **Bases de données**
  - DB -> XML -> traduction -> HTML
- **Transactions**

## 2.3 XML ailleurs

---

- **Langages pour interfacier des applications, clients-serveurs etc.**
  - ne pas négliger cet aspect, pensez par exemple aux URL-encoded strings, aux HTTP request headers etc.

### 3.. Le Document Objet Model (DOM)

#### 2.3. XML ailleurs

## 3. Le Document Objet Model (DOM)

---

- **Voir W3C Data Formats** (<http://www.w3.org/TR/NOTE-rdfarch>)
- **API (application programming interface) pour documents HTML et XML**
- **Ce API sert à:**
  - construire des documents (browsers..)
  - naviger leur structure avec un programme
  - rajouter, modifier ou détruire des éléments
- **Une sorte de DHTML++**
  - mais propre, le terme DHTML est utilisé par Netscape et Microsoft (ne se retrouve dans aucun standard)
  - permettant de manipuler toutes sortes de formats de données (comme HTML et XML) avec des scripts

De l'abstract de la spécification (<http://www.w3.org/TR/REC-DOM-Level-1/>): a platform- and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of documents. The Document Object Model provides a standard set of objects for representing HTML and XML documents, a standard model of how these objects can be combined, and a standard interface for accessing and manipulating them. Vendors can support the DOM as an interface to their proprietary data structures and APIs, and content authors can write to the standard DOM interfaces rather than product-specific APIs, thus increasing interoperability on the Web.

### 3.. Le Document Objet Model (DOM)

#### 3.1. Le principe du DOM avec un exemple

## 3.1 Le principe du DOM avec un exemple

---

### Exemple 3.1: Un simple table HTML comme “DOM tree”

- voir: REC-DOM-Level-1.19981001

#### Les données XML:

```
<TABLE>
<TBODY>
<TR> <TD>Pierre Muller</TD>
      <TD>http://pm.com/</TD> </TR>
<TR> <TD>Elisabeth Dupont</TD>
      <TD></TD> </TR>
</TBODY>
</TABLE>
```

#### Rendering typique dans un browser:

Pierre Muller	http://pm.com/
Elisabeth Dupont	

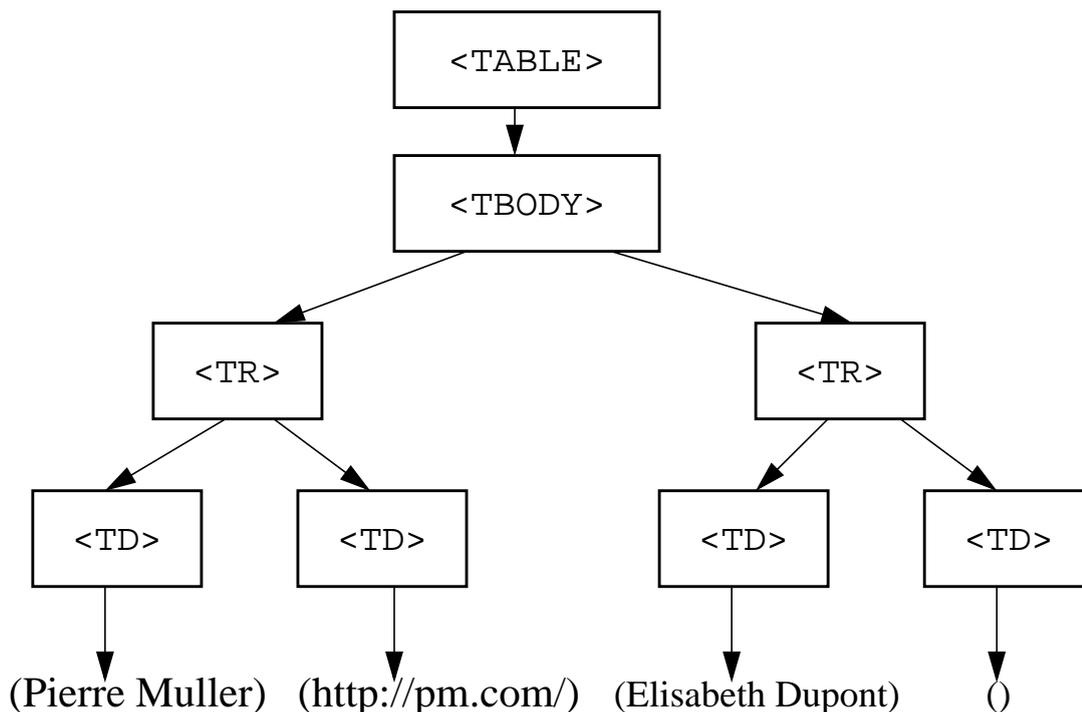
#### Représentation alternative dans un browser ou un applet:

- **line 1:**
  - Pierre Muller
  - http://pm.com/
- **line 2:**
  - Elisabeth Dupont
  - [Edit]

### 3.. Le Document Objet Model (DOM)

#### 3.1. Le principe du DOM avec un exemple

#### Représentation interne de l'arbre dans le DOM



- **Imaginez un script qui peut manipuler cette structure au niveau du contenu, de l'affichage et de l'interface utilisateur**
  - Javascript et VB dans les browsers WWW standards
  - Java (et autres) dans des "costume" browser (applets par exemple)
- **Shift de paradigme pour les pages Internet:  
Display quasi-statique => Application**

### 3.. Le Document Objet Model (DOM)

#### 3.2. Conclusion (provisoire)

## 3.2 Conclusion (provisoire)

---

- **XML et DOM ont une potentialité énorme par rapport à HTML:**
  - un seul formalisme pour (presque) tout,
  - donc un formalisme à vocation multiple,
  - un formalisme pour représenter toutes sortes de structures de données,
  - mieux adapté à l'âge de l'information et aux systèmes d'information
- **Peut-on faire confiance ?**
  - XML est le produit de vieilles et bonnes connaissances (SGML)
  - L'intérêt actuel pour XML est énorme
  - XML est beaucoup plus flexible que HTML,....  
mais beaucoup plus strict et formel ,  
..... donc ça plaît aux grandes organisations
- **Il faut investir dès maintenant:**
  - se renseigner
  - se former
  - faire des essais
- **Rien ne marche très bien en ce moment**
  - IE Explorer contient des bugs et ne supporte pas XSL en mode direct
  - Netscape 5 n'est pas encore là
  - Ecrire des applets Java coûte très cher
  - Du server-side XML-> HTML se justifie bien pour certaines applications, mais de nouveau, XSL pose un problème
  - Il n'existe pas assez de DTD et style-sheets, ceci ne pose pas de problème pour les grandes organisations, mais pour nous.
  - .....(à suivre) .....
- **PS: les outils de développement marchent très bien**
  - 3-5 bonnes libraires XML en Java par exemple

#### 4.. Introduction technique à XML

#### 3.2. Conclusion (provisoire)

## 4. Introduction technique à XML

---

- **XML = Extensible Markup Language**
- **Utilisation de tags et attributs comme dans HTML**

### **Définition un peu plus formelle:**

- **XML est un langage pour définir une classe de “data-objects”.**
- **Structure physique:**
  - Un document XML est une collection d’entités (entities)
  - Une entité peut référer à d’autres entités pour inclusion
  - Chaque document commence par une racine (“root” ou "document entity")
- **Structure logique, a document is composed of:**
  - declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup :)
- **Structure physique et logique doivent s’imbriquer proprement**
  - voir I-A. ““Well-formed” documents” [25]

## 4.. Introduction technique à XML

### 4.1. Les notions de well-formed et valid

## 4.1 Les notions de well-formed et valid

---

### A. "Well-formed" documents

- **Le document commence par une déclaration XML**
  - `<?xml version="1.0"?>`
- **Structure hiérarchique:**
  - begin-tags and end-tags doivent correspondre
  - pas de croisements de type `<i>...<b>...</i> .... </b>`
  - Case sensitivity
- **Tags vides (oui, mais ...):**
  - Les tags vides utilisent la syntaxe XML (e.g. `<empty/>`)
- **Les valeurs d'attributs sont quotés:**
  - (e.g. `<a href="http://tecfa.unige.ch:8080/xml.html">`)
- **Un seul élément root:**
  - L'élément root ne peut apparaître qu'une fois
  - Le root ne doit pas apparaître dans un autre élément (comme `<html>`)
- **Caractères spéciaux: <, &, >, ", ' ,**
  - Utilisez `&lt;`; `&amp;`; `&gt;`; `&quot;`; `&apos;`; à la place dans un texte !

### B. "Valid documents"

- **Déclaration des entités**
  - un document "valid" doit posséder un DTD
- **DTD = document type declaration:**
  - Une grammaire (ou jeu de règles) qui définit les tags utilisés, et qui spécifie leur possible imbrication.
  - le DTD peut être référencé par un URI ou inclus dans le document XML

#### 4.. Introduction technique à XML

##### 4.1. Les notions de well-formed et valid

### C. La définition de XML

- **Les grammaire en informatique sont décrits sous format EBNF (Extended Backus-Naur Form)**
- **Le langage XML utilise un autre formalisme**
  - Tous les langages XML utilisent le formalisme DTD
  - mais XML est défini en EBNF, donc il faut comprendre EBNF pour bien comprendre la spécification de XML.
  - Voir: the Annotated XML Specification, <http://www.xml.com/axml/axml.html>

#### **Exemple 4.1: Quelques éléments de XML à titre d'illustration**

```
[1] document ::= prolog element Misc
[2] element  ::= EmptyElemTag
                | STag content ETag [WFC: Element Type Match]
                                   [VC: Element Valid]
```

- **Pour construire des simples DTD, comprendre les détails de la spécification formelle de XML n'est pas nécessaire**

## 4.. Introduction technique à XML

### 4.2. Exemples

## 4.2 Exemples

---

### Exemple 4.2: Une veille blague en XML

Voir: [http://www.arbortext.com/Think Tank/XML Resources/ A Technical Intorduction to XM/a technical intorduction to xm.html](http://www.arbortext.com/Think_Tank/XML_Resources/A_Technical_Introduction_to_XML/a_technical_introduction_to_xml.html)

- **Données XML**

- On a un document de type <oldjoke>
- Avec deux personnes "burns" et "allen" qui parlent
- <quote> est un marqueur de mise en forme
- un applaudissement (vide)

```
<?XML version="1.0"?>
  <oldjoke>
    <burns>
      Say <quote>goodnight</quote>,
      Gracie.
    </burns>
    <allen><quote>Goodnight, Gracie.</quote>
    </allen>
    <applause/>
  </oldjoke>
```

- **Le DTD (à titre d'indication)**

- Voir 5. "Les DTD" [29] pour comprendre

```
<!ELEMENT oldjoke (burns+, allen, applause?)>
  <!ELEMENT burns (#PCDATA | quote)*>
  <!ELEMENT allen (#PCDATA | quote)*>
  <!ELEMENT quote (#PCDATA)*>
  <!ELEMENT applause empty>
```

## 4.. Introduction technique à XML

### 4.2. Exemples

#### Exemple 4.3: Une recette en XML

Source: Introduction to XML by Jay Greenspan,

[http://www.hotwired.com/webmonkey/98/41/index1a\\_page5.html?tw=html](http://www.hotwired.com/webmonkey/98/41/index1a_page5.html?tw=html)

```
<?xml version="1.0"?>
<list>
  <recipe>
    <author>Carol Schmidt</author>
    <recipe_name>Chocolate Chip Bars</recipe_name>
    <meal>Dinner
      <course>Dessert</course>
    </meal>
    <ingredients>
      <item>2/3 C butter</item>
      <item>2 C brown sugar</item>
      <item>1 tsp vanilla</item>
      <item>1 3/4 C unsifted all-purpose flour</item>
      <item>1 1/2 tsp baking powder</item>
      <item>1/2 tsp salt</item>
      <item>3 eggs</item>
      <item>1/2 C chopped nuts</item>
      <item>2 cups (12-oz pkg.) semi-sweet choc. chips</item>
    </ingredients>
    <directions>
      Preheat oven to 350 degrees. Melt butter;
      combine with brown sugar and vanilla in large mixing bowl.
      Set aside to cool. Combine flour, baking powder, and salt; set aside.
      Add eggs to cooled sugar mixture; beat well. Stir in reserved dry
      ingredients, nuts, and chips.
      Spread in greased 13-by-9-inch pan. Bake for 25 to 30 minutes until
      golden brown; cool. Cut into squares.
    </directions>
  </recipe>
</list>
```

- **Cette exemple montre quelques éléments de type HTML**
- **Pour le DTD, voir exemple 5.4 “Un DTD pour une recette” [33]**

## 5.. Les DTD

### 5.1. Introduction aux DTD de XML

## 5. Les DTD

---

### 5.1 Introduction aux DTD de XML

---

- **DTD = Document Type declaration**
- **il s'agit d'une grammaire qui définit:**
  - quels tags sont autorisés à l'intérieur d'autres tags
  - quels tags et attributs sont à option
- **Chaque "tag" XML est défini comme un élément dans le DTD**
  - Exemple illustratif: `<!ELEMENT title (#PCDATA)>`

#### A. HTML avec XML ?

- **HTML est formalisable comme DTD SGML**
- **La version XML de HTML s'appelle XHTML**
  - légèrement différente de HTML 4.0

## 5.. Les DTD

### 5.2. Le langage pour définir une grammaire DTD

## 5.2 Le langage pour définir une grammaire DTD

---

### A. Prologue et déclaration de DTD

- **Le DTD est inséré entre la déclaration de XML et le document lui-même.**
- **Les 2 ensembles sont appelés le prologue**

#### **Exemple 5.1: Hello XML sans DTD**

```
<?xml version="1.0" standalone="yes"?>
<hello>
Hello XML et hello cher lecteur !
</hello>
```

#### **Exemple 5.2: Hello XML avec DTD**

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE hello [
  <!ELEMENT hello (#PCDATA)>
  ]>
<hello>
Hello XML et hello cher lectrice !
</hello>
```

- **Chaque DTD commence par:**  
<!DOCTYPE
- **... et fini par:**  
]>
- **La racine (ici: hello) doit être indiquée après <!DOCTYPE**
- **Mais elle doit également être définie comme ELEMENT.**
- **(#PCDATA) veut dire qu'on peut insérer n'importe quel texte entre les tags <hello> .... </hello>**
  - sauf d'autre tags !!

## 5.. Les DTD

## 5.2. Le langage pour définir une grammaire DTD

B. Déclaration d'éléments**Syntaxe: Eléments DTD simples**

```
<!ELEMENT tag contenu>
```

**Tableau 2: Eléments pour les règles DTD**

<i>A et B = tags</i>	<i>Explication</i>
<i>A?</i>	A (un seul) est une option, (match A ou rien)
<i>A+</i>	Il faut un ou plusieurs A
<i>A*</i>	A est une option, il faut zero, un ou plusiers A
<i>A / B</i>	Il faut A ou B, mais pas les deux
<i>A , B</i>	Il faut A, suivi de B (dans l'ordre)
<i>(A, B) +</i>	Les parenthèses regroupent. Ici: un ou plusieurs (A suivi de B)

**Tableau 3: Mots clefs dans les règles DTD**

	<i>Explication</i>
<i>#PCDATA</i>	Il faut des données (non-interprétés par XML) dans le langage d'encodage courant.
<i>ANY</i>	Mot clé qui indique que tous les éléments sont autorisés
<i>EMPTY</i>	Tag sans "closing"

## 5.. Les DTD

### 5.2. Le langage pour définir une grammaire DTD

#### Exemple 5.3: Un DTD pour un simple Address Book

voir: <http://www.software.ibm.com/xml/education/tutorial-prog/writing.html>

```
<!DOCTYPE addressBook [  
  <!ELEMENT addressBook (person)+>  
  <!ELEMENT person (name,email*)>  
  <!ELEMENT name (family,given)>  
  <!ELEMENT family (#PCDATA)>  
  <!ELEMENT given (#PCDATA)>  
  <!ELEMENT email (#PCDATA)>  

```

#### Exemple juste

```
<addressBook>  
  <person>  
    <name>  
      <family>Wallace</family> <given>Bob</given>  
    </name>  

```

#### Exemple faux (3 erreurs)

```
<addressBook>  
  <address>Derrière le Salève</address>  
  <person>  
    <name>  
      <family>Schneider</family> <firstName>Nina</firstName>  
    </name>  
    <email>nina@dks.com</email>  
  </person>  
  <name>  
    <family> Muller </family> </name>  
</addressBook>
```

## 5.. Les DTD

### 5.2. Le langage pour définir une grammaire DTD

#### Exemple 5.4: Un DTD pour une recette

- Pour un exemple XML voir l'exemple 4.3 "Une recette en XML" [28]

```
<!DOCTYPE list [  
  <!ELEMENT list (recipe+)>  
  <!ELEMENT recipe (author, recipe_name, meal, ingredients, directions)>  
  <!ELEMENT ingredients (item+)>  
  <!ELEMENT meal (#PCDATA, course?)>  
  <!ELEMENT item (#PCDATA, sub_item*)>  
  <!ELEMENT recipe_name (#PCDATA)>  
  <!ELEMENT author (#PCDATA)>  
  <!ELEMENT course (#PCDATA)>  
  <!ELEMENT item (#PCDATA)>  
  <!ELEMENT subitem (#PCDATA)>  
  <!ELEMENT directions (#PCDATA)>  
>
```

- **Voir:** [http://tecfa.unige.ch/guides/xml/examples/simple-dtd/choco\\*](http://tecfa.unige.ch/guides/xml/examples/simple-dtd/choco*)
- **NOTE:** vérifier si cet exemple est vraiment légal, voir meal et item ! IE et Xed saute avec ce truc !

#### Exercice 5.1: Lecture d'un DTD

- L'élément recette contient combien d'éléments ?
- Comment aurait-on pu rendre un de ces éléments optionels ?
- Quel élément n'est pas utilisé dans l'exemple à page 28 ?
- Faites 2 exemples pour l'élément "meal" (un avec course et un sans course)

## 5.. Les DTD

## 5.2. Le langage pour définir une grammaire DTD

C. Déclaration d'attributs

- Voir la spec pour une définition "clean" !!

**Syntaxe: Attributs DTD simples**

```
<!ATTLIST tag nom TypeAttribut TypeDefault Defaut
```

**Tableau 4: Types d'attributs**

	<i>Explication</i>
<b>ID</b>	Attribut unique dans le document
<b>IDREF</b>	Doit correspondre à un ID attribut dans un des éléments
<b>IDREFS</b>	Doit correspondre à 1 ou plusieurs ID attributs (séparés par des blancs)
<b>(A,B,C,..)</b>	Liste énumérée

**Tableau 5: Mot clefs pour les défauts d'Attributs**

	<i>Explication</i>
<b>#IMPLIED</b>	Attribut à option
<b>#REQUIRED</b>	Attribut nécessaire
<b>#FIXED Value</b>	Attribut avec valeur fixe

**Illustrations:**

```
<!ATTLIST person gender (male|female) #IMPLIED>
<!ATTLIST form method #FIXED "POST">
<!ATTLIST list type (bullets|ordered) "ordered"
<!ATTLIST sibling type (brother|sister) #REQUIRED
```

**5.. Les DTD****5.2. Le langage pour définir une grammaire DTD****Exemple 5.5: Un DTD pour un moins simple Address Book**

[fichier ab.dtd]

```
<?xml encoding="UTF-8"?>
<!ELEMENT addressBook (person)+>
<!ELEMENT person (name,email*)>
<!ATTLIST person id ID #REQUIRED>
<!ATTLIST person gender (male|female) #IMPLIED>
<!ELEMENT name (#PCDATA|family|given)*>
<!ELEMENT family (#PCDATA)>
<!ELEMENT given (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT link EMPTY>
<!ATTLIST link manager IDREF #IMPLIED subordinates IDREFS #IMPLIED>
```

**Exemple:**

```
<!DOCTYPE addressBook SYSTEM "ab.dtd">
<addressBook>
  <person id="B.WALLACE" gender="male">
    <name>
      <family>Wallace</family> <given>Bob</given>
    </name>
    <email>bwallace@megacorp.com</email>
    <link manager="C.TUTTLE"/>
  </person>

  <person id="C.TUTTLE" gender="female">
    <name>
      <family>Tuttle</family> <given>Claire</given>
    </name>
    <email>ctuttle@megacorp.com</email>
    <link subordinates="B.WALLACE"/>
  </person>
</addressBook></pre>
```

## 5.. Les DTD

## 5.2. Le langage pour définir une grammaire DTD

D. Déclaration d'Entities

- Les entités prédéfinies sont: `&amp;`, `&lt;`, `&gt;`, `&quot;` et `&apos;` pour les signes `&`, `<`, `>`, `"`, `'`
- Les autres sont définies par l'auteur du DTD
- ... (section pas complète, à suivre ....)

Tableau 6: Déclaration d'Entities

	<i>Explications</i>
SYSTEM	Le contenu de l'entité est accessible par un URI
.....	.....

**Syntaxe: Structure d'une déclaration d'entity simple:**

```
<!ENTITY tag "Contenu">
```

**Illustrations:**

```
<!ENTITY tecfaUnit "Unité de technologies de formation et
apprentissage">
```

```
<!ENTITY tecfaDesc SYSTEM "http://tecfa.unige.ch/..//
tecfa_description.xml">
```

.... à suivre (XLL, XSL, etc.)

## II: XML: En pratique



Version 0.2 (novembre 1998)  
Dernière modification: 24/3/99

<http://tecfa.unige.ch/guides/xml/slides/xml.html>  
<http://tecfa.unige.ch/guides/xml/slides/xml.pdf>

COPYRIGHT: Daniel Schneider

TECFA,  
Faculté de Psychologie et des Sciences de l'Education,  
Université de Genève

9 route de Drize, CH-1227 Carouge  
Tél.: +41 22 70 9694

Email: [Daniel.Schneider@tecfa.unige.ch](mailto:Daniel.Schneider@tecfa.unige.ch)  
MOO: [tecfamoo.unige.ch](http://tecfamoo.unige.ch) 7777 (aka Daniel)  
WWW: <http://tecfa.unige.ch/tecfa-people/schneider.html>

**1.. UNDER CONSTRUCTION****1.1. Autres ressources pour l'enseignement à TECFA****1. UNDER CONSTRUCTION**

---

..... (à suivre)

**1.1 Autres ressources pour l'enseignement à TECFA**

---

- **Les XML & Java Slides sont dans les Java Slides**
  - <http://tecfa.unige.ch/guides/java/staf2x/slides/java.html>
- **Pour PHP et XML voir pour le moment:**
  - <http://tecfa2.unige.ch/guides/php/examples/simple-xml/>