Introduction à XML et DOM

Code: xml-dom

Originaux

url: **http://tecfa.unige.ch/guides/tie/html/xml-dom/xml-dom.html**
url: **http://tecfa.unige.ch/guides/tie/pdf/files/xml-dom.pdf**

Auteurs et version

- Daniel K. Schneider

- Version: 1.2 (modifé le 10/8/01 par DKS)

Prérequis

- Internet et WWW de base, HTML de base

Module technique précédent: internet
Module technique précédent: www-tech
Module technique précédent: html-intro

Modules

Module technique suivant: xml-tech

Objectifs

- Avoir une idée du XML et DOM framework du consortium WWW
- Avoir une idée de la définition technique de XML (méta-langage)
- Avoir envie de faire du XML :)

1. Table des matières détaillée

1. Table des matières détaillée 3
2. Vers un nouveau paradigme Internet 4
   - **2.1 Le problème HTML et la solution XML 7**
   - **2.2 W3C Data Formats 10**
   - **2.3 Xlink - Vers un meilleur hypertexte 15**
   - **2.4 XML avec Style 17**
3. XML dans la pratique 19
   - **3.1 XML Authoring 19**
   - **3.2 XML sur Internet 20**
4. Le Document Objet Model (DOM) 21
   - **4.1 Le principe du DOM avec un exemple 22**
5. Conclusion (provisoire) 24
   - **5.1 Questions techniques 24**
   - **5.2 XML dans le monde de l'éducation 26**

2. 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
- XSLT@Tecfa: http://tecfa.unige.ch/guides/xml/xsl-pointers.html
- DOM@Tecfa: http://tecfa.unige.ch/guides/dom/pointers.html
- RDF @Tecfa: http://tecfa.unige.ch/guides/rdf/pointers.html
- Voir le WDVL Acronym Expander pour l'explication de la plupart des sigles: http://wdvl.internet.com/Authoring/Languages/XML/Overview/acronym.html

C. XML sert à créer des "markup languages"

- un langage "markup" (langage de balisage") sert à encoder/structurer un texte.
- XML fournit le formalisme + des mécanismes pour définir des langages

Historique des langages "markup" Internet:

SGML (Standard Generalized Markup Language, ISO standard en 1986)

- meta-langage pour définir des langages de "markup"
- HTML (1990) est une application SGML ayant assez peu de balises (tags)

XML (1997, -)

- un meta-langage plus léger que SGML adapté au Web
- permet la définition de langages adaptés à des besoins très variés
- XML est un langage pour décrire des structures de données
  - sert à organiser l'échange d'informations
  - sert à remplacer HTML pour certaine tâches (mais fonctionne d'une autre façon !
  - donc: XML ***n'existe pas*** au même sens que HTML, ce n'est qu'un formalisme !

D. 2 façons d'aborder XML

(1) XML comme formalisme

(un formalisme pour créer des grammaires)

```
<!ELEMENT page  (title, content,
comment?)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT content (#PCDATA)>
<!ELEMENT comment (#PCDATA)>
```

```
 <title>Hello Cocoon friend</
title>
 <content>
      Here is some content :)
 </content>
 <comment>
      Written by DKS/Tecfa,
 </comment>
</page>
```

(2) Le "XML framework" du W3C

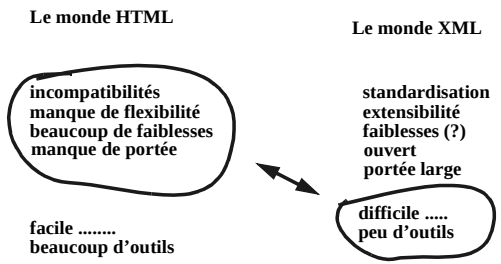
(un ensemble de langages/grammaires pour créer un web plus puissant. Ces formalismes sont définis avec XML !)

graphisme: SVG
formattage: XSL/FO
hypertexte: Xlink, Xpointer
recherche: XQL
transformations: XSLT
commerce: SOAP
.....
.....

Pour faire vivre ces langages il faut des outils (se mettent peu à peu en place)

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

A. Overview



Note:

- Aborder XML à travers les déficiences de HTML est une chose, faire croire que XML se discute au même niveau est faux !

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"

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 l’impression ou l’affichage
- Le code PDF est très difficilement “lisible” et manipulable

E. Avec Word, Framemaker, etc.

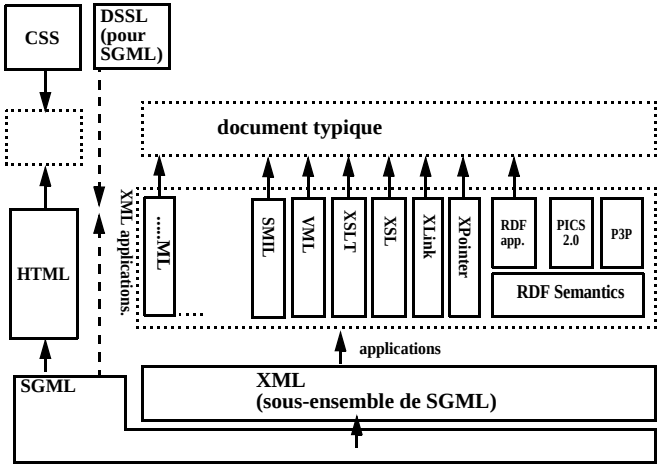
- On est prisonnier d’un format propriétaire (qui change selon les veux des fabricants)
- Absence de “features” typiques de HTML
- Formats aussi difficiles (RTF, MIF, etc) que PDF.
- Pas utile comme format de représentation de données

F. Résumé

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

2.2 W3C Data Formats

A. Eléments importants du "XML framework":



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 (notamment dans le domaine du "document processing")
- 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/FO (application XML): Langage de style pour XML
- XSLT (application XML): Langage de transformation pour XML

C. Fonctionalités 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

	monde HTML	monde XML	monde SGML
Linking de documents	Balise <A>	Xlink (+ Xpointer & Xpath)	HyTime & TEI
Assemblage de documents	"calculs server-side"	XInclude (+ Xpointer & Xpath) ou entités ou "calculs server-side"	Entités SGML
Style	CSS2 CSS1	XSL (CSS)	DSSL
Markup	HTML	applications XML (XHTML, Docbook)	applications SGML (Docbook, TEI, ...)
Multimédia	formats "exotiques" (Flash, Gif, Jpeg)	formalismes XML (SVG, SMIL, MathML)	
Interface entre Markup et Scripting	Document Object Model (DOM)		
Scripting	Javascript, JScript, ECMAScript, ...		

url: **Voir (http://wdvl.internet.com/Authoring/Languages/XML/Overview)**

Fonctionalités 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), XSchema ou autres grammaires.
- validation (on peut contraindre les auteurs à suivre un schéma DTD)

D. Applications XML (voir surtout page suivante)

- "Vocabulaires": XML est un langage assez universel pour la représentation de contenus, exemples:
  - CML (Chemical Markup Language)
  - X3D (VRML-Xmlisé)
  - NML (News Markup Language) et NITF (News Interchange Text Format)
  - vous pouvez créer votre propre langage.....
- XML est un langage pour rajouter d'autres fonctionalités aux vocabulaires:
  - hyper-liens
  - graphisme
  - catégorisation
  - échange de données
  - échange de requêtes entre programmes
  - .... etc.
- Il existe déjà une centaine d'applications XML....

E. Quelques applications XML du W3C (consortium WWW)

- XSL/FO (application XML): Langage de style pour XML
- XSLT (application XML): Langage de transformation pour XML
- XLink: Hypertext links
- XPointer (pointeurs vers une ressource) et XPath (chemins dans la structure)
  - (utilisés par XSLT, XInclude, XLink, etc.)
- 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/**
- SVG: Scalable Vector Graphics
- MathML: Mathematical Markup Language
  - **http://www.w3.org/Math/**
- XHMTL: (HTML 4.0 en XML)
  - Tags fermés, pas de croisements, imbrication correcte des éléments

2.3 Xlink - Vers un meilleur hypertexte

- reste une proposition, pas d'implémentation complète pour le moment (5/2001).

A. Composantes

- Xlink = comment insérer un lien dans un document XML (le lien exprime une relation entre deux ou plusieurs objets)

  url: **http://www.w3.org/TR/xlink**

Xlink repose sur d'autres standards (partagés avec XSLT par exemple)

- XPointer = comment identifier un fragment XML (utilisable par des liens)
- XPointer repose sur XPath = comment identifier un chemin vers une ressource

url: **http://www.w3.org/TR/xptr**

url: **http://www.w3.org/TR/xpath**

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 de quelques caractéristiques Xlink

- 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

2.4 XML avec Style

Les style-sheets permettent de:

- préparer/arranger un contenu pour une "présentation"
- définir le "layout" (mise en forme, formattage) d'un "texte" écrit en XML

L'utilité des style-sheets est donc de:

- séparer contenu et présentation
- rationaliser le travail (un style-sheet pour beaucoup de documents)

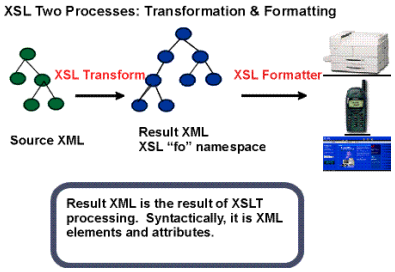
A. XSL: Extensible Stylesheet Language

- XSL est le langage recommandé par le W3C
  url: **http://www.w3.org/TR/xsl**

XSL possède 2 fonctions principales

1. langage de transformation (XSLT) d'éléments XML
   - Par exemple: création de tables de matières, traduction de XML vers HTML
     url: http://www.w3.org/TR/xslt
2. langage de mise en page (formattage) (XSL/FO)
   url: http://www.w3.org/TR/xsl/

Avec un GIF (tiré du working draft de la spécification)



Features de formattage (XslFO)

- 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 ....

B. CSS (Cascading Style Sheets)

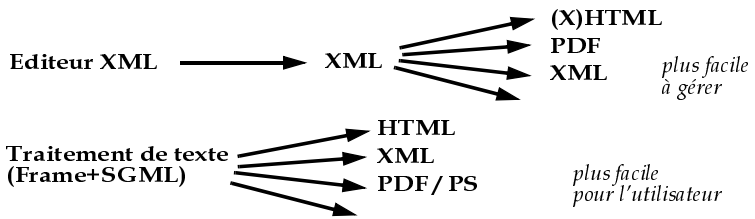
- CSS1 (pour HTML seulement, mais prédominant)
- CSS2 (support pour XML, Mozilla 0.9x & IE 5.5 partiellement)

3. XML dans la pratique

3.1 Edition de textes avec XML

- Outils permettant d'éditer un "arbre" (quelques programmes Java gratuits)
- Outils d'édition de texte structuré (éditeurs de programmation comme Emacs)
- Outils semi-professionnels (comme XMetal ou EpcEdit): assez chers.
- Outils professionnels SGML/XML comme FrameMaker+SGML: chers.)
- Plug-ins pour traitement de texte (médiocres encore)
- Filtres vers XML (HTML, RTF, Latex, etc.): médiocres par nature

2 grandes options:



3.2 Vocabulaires pour "publishing"

1. Text markup général: Latex en mieux
   Vocabulaires "neutres" mais très détaillés pour rédiger des textes larges. Ces vocabulaires ont souvent leur origine dans le monde "SGML".
   - Exemple Docbook **http://www.docbook.org/xml/**
     - répandu dans le monde technique
     - support au niveau des outils XML (et SGML) haut de gamme
     - très détaillé (> 350 éléments)
2. Text markup décentralisé
   Schémas servant à générer et assembler du texte à partir de multiples sources selon besoin
   - Exemple DITA **http://www-106.ibm.com/developerworks/xml/library/x-dita3/index.html**
3. Par domaine: SwissDroitML ?
   Vocabulaires pour un domaine précis (souvent juste pour échanger des données)
4. Pour échange de données:
   - par exemple des nouvelles avec RSS
5. Multimédia: SVG, Web3D, MathML, ...

Vers plus d'efficacité, vers un nouveau Babylon, vers une bureaucratisation ???

3.3 Filtres

A. Filtres XML -> HTML

- Les mêmes qui sont utilisés “server-side”, souvent des processurs XSLT
  - souvent écrits en Java,
- traductions "manuelles" avec un langages de scripting ayant accès à un parseur XML (PhP, JSP, ASP, Perl, etc.)

B. Filtres XXX -> XML

- RTF, HTML, Lotus Notes, ...

3.4 XML sur Internet

A. Browsers WWW: à déconseiller pour sites grand public !

- IE Explorer 5.5: support XML et CSS, XSL et XSLT (limité)
- Mozilla (Netscape 6): XML + CSS (XSLT pour bientôt).

B. Applets Java

- Pleins de librairies pour développeurs existent
- .... et certaines applications aussi

C. Server-Side: marche !

- Traducteurs XML->HTML
  (y compris "XML publication/application frameworks")
  - XML + CSS, XML + XSL, XML + traduction “ad-hoc”, etc.
- Bases de données
  - DB -> XML -> traduction -> HTML
- Transactions

D. XML comme protocole de communication: marche !

- Langages pour interfacer des applications, clients-serveurs etc.

4. 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.

4.1 Le principe du DOM avec un exemple

Exemple 4.1 Un simple tableau HTML nommé “DOMtree”

- voir: REC-DOM-Level-1.19981001

Le document XML

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

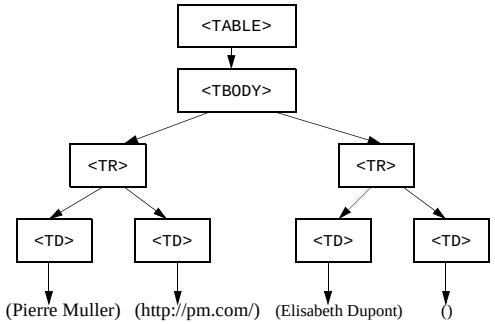
Rendering typique dans un browser

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

Représentation alternative dans un browser ou un autre

- line 1:
  - Pierre Muller
  - http://pm.com/
- line 2:
  - Elisabeth Dupont

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 "costum" browser (applets par exemple)
- Shift de paradigme pour les pages Internet:
  Display quasi-statique => Application

5. Conclusion (provisoire)

5.1 Questions techniques

A. XML et DOM ont un potentiel 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

B. 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
- Problème: balkanisation, problèmes d'acceptation (terrorisme de schémas)

C. Il faut investir dès maintenant:

- se renseigner, se former
- faire des essais pour savoir comment organiser son XML "server-side"

(la technologie marche, voir le module XML-server-side !)

D. Clients: lacunes sauf pour CSS1

- Aucun support pour Netscape 4.x
- IE Explorer 5.5 contient des bugs et ne supporte pas XSL en mode direct
- Netscape 6 n'implémente pas encore XSLT ni XSL/FO
- CSS (level 1) correct avec IE 5.5 et NS 6.0

E. Schémas

- Il n'existe pas assez de DTD et style-sheets largement acceptés
- .....(à suivre) .....

F. Server-side XML->HTML: marche bien

- Cocoon Framework de Apache: http://xml.apache.org/cocoon/
- engins XSLT (pleins de solutions: IBM/Lotus, Saxxon, XT, ...)
- manuellement (avec PHP par exemple)

G. Outils de dévelopement

- marchent bien (libraires XML en Java par exemple)

5.2 XML dans le monde de l'éducation

A. Vocabulaires typiques

- "Tutorial Markup"
- "Quiz Markup"
- Informations sur les étudiants ("exchange formats")
- Textes structurés dans des bibliothèques on-line, par ex:
  - The Oxford Text Archive
  - The Humanities Text Initiative (Univ of Michigan)
- Catalogues:
  - "Meta-data servers"
  - Catalogues bibliothécaires comme Marc
- "Content packaging"
- "Instructional text"

Il existe peu d'applications pour ces standards emergeants

B. Organisations

- Il existe plusieurs organisations qui développent des standards:
  - IMS (Meta-data, persons, some content, student work)

  url: **http://www.imsproject.org/**
  - DoD ADL / SCORM (meta-data, contents)

  url: **http://www.adlnet.org/**
  - IEEE LTSA (LOM meta-data)

  url: **http://edutool.com/ltsa/**
  - Ariadne (meta-data)

  url: **http://ariadne.unil.ch/**
- se coordonnent quelque peu,
- se sont concentrés principalement sur les "meta-data" et le "content packaging" (catégoriser et emballer des ressources),
- travaillent sur le concept du "reusable learning object" (des éléments que l'on peut réutiliser dans différents contextes et dans différents systèmes).