Langages à balises : une introduction

LIONEL MÉDINI
UFR INFORMATIQUE
UNIVERSITÉ CLAUDE BERNARD LYON 1

D'après le cours de Yannick Prié
2010-2011 – Master SIB
M1 – UE 3 / Bloc 2 – Cours 1

Objectif généraux du module

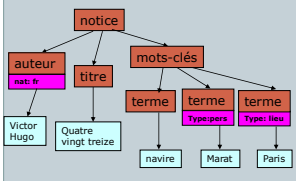
- Comprendre les grands principes de la représentation de données et de documents numériques à l'aide d'un langage à balises
- Découvrir XML, son histoire et son fonctionnement
- Définir des langages basés sur XML à l'aide de DTD
- Apprendre les bases de XHTML pour la génération de pages web
- S'initier à la transformation de documents en utilisant XSL et un moteur XSLT

Objectifs de ce cours introductif

- Introduction aux langages à balises et à leurs principes
  - arbres
    - définition
    - grammaires
  - balises et balisage
  - langages à balises
    - historique de ces langages
    - présentation des principaux langages de la « galaxie XML »
    - présentation de la suite du cours

Idée générale

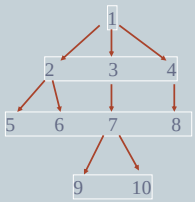
- Représenter de l'information dans des structures arborescentes
- Coder ces structures dans des fichiers, qui pourront être échangés



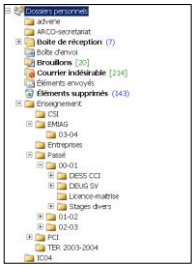

```
<?xml version="1.0"
encoding="ISO-8859-1"?>
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

Parler des arbres

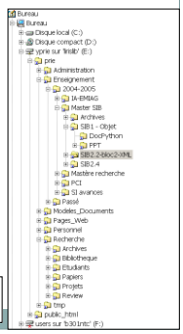
- Arbre
- Noeud
  - nœuds fils et pères
- Racine
- Feuille
- Chemin
  - suite de nœud
- Branche
  - chemin se terminant sur une feuille
- Ancêtres et descendants
- Taille d'un arbre
  - nombre de nœuds
- Profondeur d'un nœud



Les arbres sont partout !



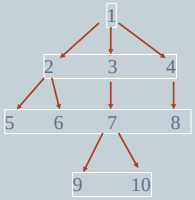
chemin
E:\prie\Enseignement\2004-2005\
Master SIB\SIB3.2-bloc2-XML



Parcours d'arbre

7

- Largeur d'abord
  - 1
  - → 2 → 3 → 4
  - → 5 → 6 → 7 → 8
  - → 9 → 10
- Profondeur d'abord
  - 1 → 2 → 5 → 6
  - → 3 → 7 → 9 → 10
  - → 4 → 8

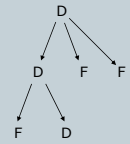


Notion de grammaire

8

- Définition
  - Une grammaire est un système formel qui définit complètement un arbre
- Contenu
  - vocabulaire
  - règles de production
    - Avec une syntaxe spécifique
- Exemple du système de fichiers
  - Vocabulaire
    - D (Dossier)
    - F (fichier)
  - Règles
    - Départ : D
    - D → (D|F)*
  - Syntaxe des règles

Règle	Signification
*	Zéro, un, plusieurs
|	OU

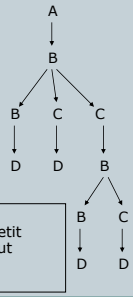


Grammaire : autre exemple

9

- Vocabulaire
  - A, B, C, D
- Règles
  - Départ : A
  - A → B+
  - B → (B , C*) | D
  - C → (D | B)
- Syntaxe des règles

Règle	Signification
*	Zéro, un, plusieurs
+	Un ou plusieurs
|	OU
,	ET



→Question
quel est le plus petit arbre que l'on peut écrire avec cette grammaire ?

Notions de balise et de balisage

- Définitions
  - Donnée : valeur associée à un type de données
    - Décrite par des caractéristiques de forme
    - Indépendante de son interprétation : donnée « brute »
  - Élément d'information : ensemble de données faisant « sens »
    - On parle aussi de « grains d'information »
  - Document : regroupement cohérent d'éléments d'information
    - Autour d'une thématique commune
    - Selon une structure donnée

Notions de balise et de balisage

- Définitions
  - Information structurée : bases de données
    - Éléments d'information stockés séparément
    - Accès par requêtes
    - Traitements facilités
  - Information non structurée : corpus documentaires
    - Éléments d'information stockés sous forme de textes
    - Accès par recherche d'information
    - Traitements complexes
  - Information semi-structurée : langages à balises
    - Éléments d'information stockés dans des documents
    - Permet les deux types d'accès et de traitements

Notions de balise et de balisage

- Définitions
  - Balise : signe marquant une position particulière
    - Signe = élément d'information
    - Marquer = permettre la distinction
      - Ex : fusée de détresse, signal radio, « tag » HTML
    - Position = par rapport à l'espace considéré
      - Ex : espace 3D (avion), 2D (bateau), 1D (document)
    - Particulière = en général, la position d'un élément de l'espace qu'on cherche à repérer
      - Ex : aéroport, bateau, élément d'information
  - ⇒ Type d'information facilement repérable permettant d'identifier d'autres éléments informationnels (i.e. « méta-information »)

Notions de balise et de balisage

- Définitions
  - Balisage documentaire : utilisation de balises
    - Pour marquer des points précis d'un document
    - Pour marquer des zones (segments) de document
      - Balisage de début et de fin de zone
    - Pour structurer le document
  - ⇒ Utilisation de plusieurs types de balises
    - ⇒ En fonction du type d'élément à marquer
    - ⇒ En fonction du type de marquage (point, début, fin)

Notion de langage à balises

14

- Tous les langages ayant pour objectif de représenter de l'information en utilisant des balises
- Définis par
  - vocabulaire
    - noms des éléments
  - grammaire
    - mode d'organisation des éléments
      - des éléments en contiennent d'autres
  - + attributs des éléments
    - un peu plus de structure (voir cours XML)
- Une description
  - ensemble d'éléments organisés dans un fichier
  - contenus terminaux (texte)

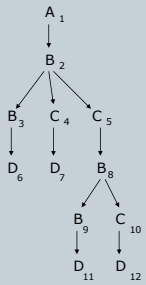
Notion d'élément

15

- Problème
  - Un fichier est une suite d'octets
  - ➔Comment représenter une structure d'arbre dans un fichier ?
- Solution
  - décrire l'arbre comme un ensemble d'éléments qui se contiennent les uns les autres.
  - représenter les éléments entre deux balises
    - balises ouvrantes
      - on les notera par exemple `<nom>`
    - balises fermantes
      - on les notera par exemple `</nom>`

Exemple

16



Eléments

$A_1 \subset B_2$
$B_2 \subset B_3\ C_4\ C_5$
$B_3 \subset D_6$
$C_4 \subset D_7$
$C_5 \subset B_8$
$B_8 \subset B_9\ C_{10}$
$B_9 \subset D_{11}$
$C_{10} \subset D_{12}$

Eléments et balises

```
<A>
  <B>
    <B>
      <D></D>
    </B>
    <C>
      <D></D>
    </C>
    <C>
      <B>
        <B><D></D></B>
        <C><D></D></C>
      </B>
    </C>
  </B>
</A>
```

Types de langages à balises

- Langages de *balisage procédural*
  - permettent de décrire la mise en forme (formatage) d'un document
  - ex. : PS, RTF, TeX, HTML

Exemple 1/3

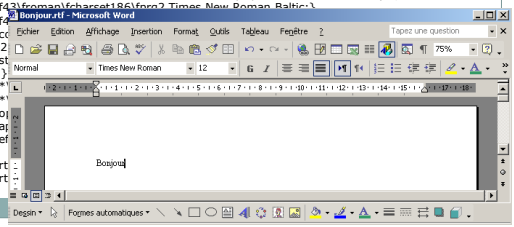
- Le langage Postscript

```
%!PS-Adobe-3.0
%%Title: Microsoft Word - Document1
%%Creator: PSCRIPT.DRV Version 4.0
%%CreationDate: 03/02/02 10:47:00
...
%%EndComments
%%BeginProlog
%%BeginProcSet: Pscript_Res_Emul 1.0 0
/defineresource where{pop}{userdict begin/defineresource{userdict/Resources 2
...
}ifelse}bind readonly def end}ifelse
%%EndProcSet
%%BeginResource: file
...
%%EndResource
...
%%EndProlog
%%BeginSetup
...
%%BeginFeature: *PageSize A4
...
%%EndFeature
...
%%EndSetup
%%Page: 1 1
%%BeginPageSetup
...
%%EndPageSetup
...
%%IncludeFont: Courier
...
 (Courier) cvn /Type1
...
(Essai impression)S
...
(%%[ Page: 1 ]%%)  =
%%PageTrailer
%%Trailer
%%DocumentNeededFonts:
%%DocumentSuppliedFonts:
/Pscript_Win_Driver /ProcSet findresource dup /terminate get exec
Pscript_Win_Compat dup /terminate get exec
%%Pages: 1
(%%[ LastPage ]%%)  =
%%EOF
```

Exemple 2/3

- Le langage RTF

{\rtf1\adeflang1025\ansi\ansicpg1252\uc1\adeff0\deff0\stshfdbch0\stshfloch0\stshfhich0
\stshfbi0\deflang1036\deflangfe1036{\fonttbl{\f0\froman\fcharset0\fprq2{*\panose
020206030504050203 04}Times New Roman;}{\f36\froman\fcharset238\fprq2 Times
New Roman CE;}{\f37\froman\fcharset204\fprq2 Times New Roman Cyr;}...

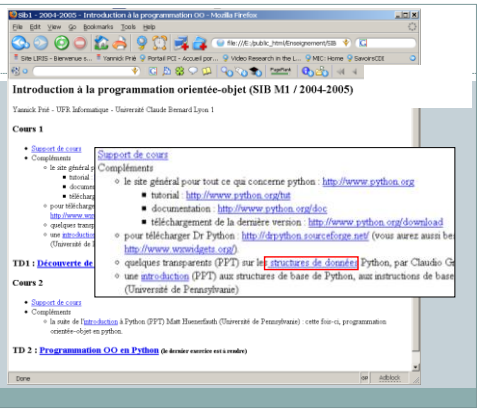


Exemple 3/3

- HyperText Markup Language (HTML)

```
<ul>
  <li>tutorial : <a href="http://www.python.org/tut">http://www.python.org/tut</a></li>
  <li>documentation : <a href="http://www.python.org/doc">http://www.python.org/doc</a></li>
  <li>t&eacute;l&eacute;chargement de la derni&egrave;re version :
      <a href="http://www.python.org/download">http://www.python.org/download</a><br>
  </li>
</ul>
</li>
<li>pour t&eacute;l&eacute;charger Dr Python :
  <a href="http://drpython.sourceforge.net/">http://drpython.sourceforge.net/</a>
  (vous aurez aussi besoin de la librairie graphique WxWidget :
      <a href="http://www.wxwidgets.org/">http://www.wxwidgets.org/</a>).</li>
<li>quelques transparents (PPT) sur les <a href="http://cmsdoc.cern.ch/i7Egcandic/pythonintro.ppt">
structures de donn&eacute;es</a> en Python, par Claudio Grandi (Universit&eacute;
de Bologne)</li>
<li>une <a href="http://www.cis.upenn.edu/~matuszek/cse391_2004/PythonIntro1.ppt">introduction</a>
(PPT) aux structures de base de Python, aux instructions de base, et &agrave;
la syntaxe, par Matt Huenerfauth (Universit&eacute; de Pennsylvanie)</li>
</ul>
```

lien



Types de langages à balises

- Langages de *balisage descriptif*
  - se contentent de décrire les données, sans but de traitement
  - ex. : RDF, OWL, MathML, n'importe quelle structure documentaire

Exemple

- Décrire une notice bibliographique
  - notice, titre, auteur, mots-clés, terme, résumé, ...
- Décrire un poème :
  - poeme, quatrain, tercet, vers, ...

```
<poeme type="sonnet">
<quatrain>
<vers>Je vis, je meurs ; je me brûle
et me noie.</vers>
<vers>J'ai chaud extrême en
endurant froidure ; </vers>
<vers> ... </vers>
<vers> ... </vers>
</quatrain>
...
</poeme>
```

```
<notice>
<auteur nat="fr">Victor Hugo
</auteur>
<titre>Quatre vingt treize</titre>
<mots-clés>
<terme>navire</terme>
<terme Type="pers">Marat</terme>
<terme Type="lieu">Paris</terme>
</mots-clés>
</notice>
```

- → vocabulaires différents
- → grammaires différentes
- → mais *même manière d'exprimer les descriptions*

Types de langages à balises

- *Méta-langages* de balisage
  - Langage avec lequel on peut définir d'autres langages
  - Pour les langages à balises
    - langage exprimant la manière dont on peut décrire une famille de langages à balise
      - comment exprimer les éléments ?
      - comment organiser les éléments ?
  - Exemples de métalangages
    - SGML
      - permet de définir : TEI, HTML...
    - XML
      - permet de définir : SVG, TEI, XHTML...

Les langages dédiés au Web

- Historique 1/3
  - 1960-1986 : SGML (norme ISO)
  - 1989 : ODA (norme ISO, concurrent de SGML)
  - Fin des années 80 : apparition/essor du web
  - 1992-1997 : HTML (versions 1.0 -> 4.01)
  - Octobre 1994 : création du World Wide Web Consortium (W3C) : http://www.w3.org
  - 1996-1999 : CSS Level 1 (fonctionnalités de base)
  - Février 1998 : XML version 1.0
  - 1998 : CSS Level 2 (fonctionnalités supplémentaires)
  - Octobre 1998 DOM Level 1 (supporte XML et HTML)

Les langages dédiés au Web

- Historique 2/3
  - Décembre 1999 : XHTML 1.0
  - 1999-2004 : RDF et RDF-Schema 1.0
  - Novembre 2000 : DOM Level 2 (supporte CSS et espaces de noms XML)
  - Mai 2001 : schémas XML 1.0
  - Juin 2001 : XLink 1.0
  - Juillet 2001 : SVG 1.0
  - Novembre 2001 : XSL 1.0
  - Janvier 2003 : SVG 1.1

Les langages dédiés au Web

- Historique 3/3
  - Février 2004 et août 2006 : XML 1.1
  - Février 2004 : OWL 1.0
  - Avril 2004 : DOM Level 3 Core
  - Octobre 2004 : XML Schema (2e edition)
  - Octobre 2004 : XQuery 1.0
  - Octobre 2004 : XPath 2.0 (Working Draft)
  - Novembre 2004 : XSLT 2.0 (Working Draft)
  - Décembre 2004 : WSDL 2.0
  - ...
  - Restent en développement : CSS L3, DOM L3...

Remarque sur la normalisation

28

- Norme industrielle
  - Référentiel publié par un organisme officiel (ISO, AFNOR...).
  - En anglais : *standard*
- Standard
  - Référentiel publié par une entité privée
  - Si diffusion large : standard *de fait*
- Consortium
  - Ensemble d'entreprises, de centres de recherche, de particuliers qui s'allient pour définir des normes et standards sur tout et n'importe quoi
  - Gain : fournir les outils au moment où le référentiel est publié
    - JPEG (Joint Picture Expert Group) → norme ISO
    - MPEG (Moving Picture Expert Group) → norme ISO
    - W3C (World Wide Web Consortium) → standards
    - ...

SGML

29

- Objectif : représenter l'information contenue dans un document indépendamment
  - des systèmes utilisés pour la saisie et le traitement
  - de la forme physique qu'il sera amené à prendre (papier, CD-ROM, web...)
  - des langues et des alphabets, latins ou non
  - des applications
- Naissance chez IBM (années soixante)
  - GML
  - gestion de la documentation technique
- Normalisation 1986 ISO-8879
  - une dizaine d'années de travail
- Utilisation
  - Description des documents dans les grosses organisations
    - complexité des langages
    - lourdeur et cherté des outils (chaîne de traitement)
    - Journal Officiel, grosses entreprises/documenations, éditeurs...
  - Echange des documents

SGML : principes

30

- Métalangage
  - permet de décrire des modèles (grammaires)
- Notion de DTD
  - Document Type Definition
  - Permet de décrire un modèle
    - un type de document
- Un document SGML
  - Est une instance du type de document
  - Doit être conforme à la DTD associée

SGML : exemple

31

Instance

```
<!DOCTYPE memo SYSTEM "memo.dtd">
<memo statut="conf">
<auteur>Serge Fleury</auteur>
<dest>
<nom>André Salem</nom>
<nom>Pollet Samvelian</nom>
</dest>
<sujet>Cours SLFE6</sujet>
<corps>
<par>Veuillez noter que le cours SLFE6 sur
les documents électronique aura bel et
bien lieu au mois de mai 2002</par>
<par>S'il y avait des changements de votre
côté, veuillez m'en aviser dans les plus
brefs délais.</par>
</corps>
</memo>
```

DTD (memo.dtd)

```
<!-- DTD utilisable pour baliser les memos en
SGML -->
<!ELEMENT memo -- ((auteur & (date?) &
sujet & dest & (cc?)), corps)>
<!ATTLIST memo statut (conf | pub) pub>
<!ELEMENT (dest | cc) -- (nom+)>
<!ELEMENT corps -- (par*)>
<!ELEMENT (auteur | date | sujet | nom |
par) - - (#PCDATA)>
```

Un élément corps contient un nombre quelconque de paragraphes

Un élément dest ou cc contient au moins un nom

(d'après http://www.cavi.univ-paris3.fr/ilpga/ilpga/tal/)

Extended Markup Language (XML)

32

- Position du problème
  - représenter et échanger des données et des documents sur le web
- SGML
  - un peu vieux
  - trop complexe
- HTML
  - trop basique
    - document = en-tête + corps
  - mélange logique / présentation
    - balise b = bold (mise en gras) :
      Attention !
    - bonne approche
      - <important>Attention !</important>
      - présenter la chaine de caractères importante avec une mise en forme particulière (italique, rouge, gras, *etc.*)

XML : prérequis

33

- XML doit être facilement utilisable sur le Web
- XML doit supporter une grande variété d'applications
- XML doit être compatible avec SGML
- Il doit être facile d'écrire des programmes qui traitent des documents XML
- Le nombre d'options doit être réduit au minimum, idéalement à zéro
- Les documents XML doivent être lisibles et raisonnablement clairs
- La conception de XML doit être menée rapidement
- La description de XML doit être formelle et concise
- Les documents XML doivent être faciles à créer
- La concision du balisage XML est d'une importance minime

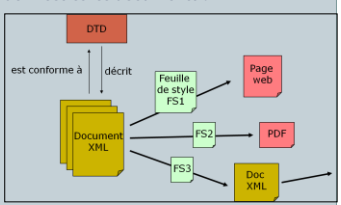
XML : principes de base

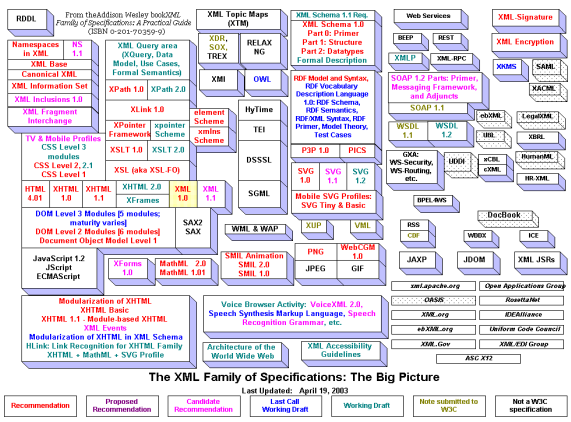
- DTD de SGML (beaucoup plus concis)
  - Restrictions de syntaxe et non de contenu
- Versions
  - V 1.0 : fév. 1998 -> 04 fév. 2004 (3ème édition)
  - V 1.1 : 04 fév. 2004 ; 16 août 2006 (2ème éd.)
- Au même titre que SGML, c'est un méta-langage de description des données
  - ⇒ Ça ne sert à rien
  - ⇒ Ça permet de définir des « applications » (types de documents) pour faire ce qu'on veut avec
    - Visualiser des pages web
    - Décrire des images...
  - Chaque document correspondant à une application est appelé « instance »

Utilisation de XML : principe général

35

- DTDs, Schéma
  - comment décrire les données et les documents ?
- Documents XML
  - les données et les documents, eux-mêmes dans des fichiers
- Feuilles de style
  - manière de présenter les données et les documents
- Remarque
  - on ne sait plus bien où sont les données et où sont les documents





XML : intégration dans les SI

37

- Stockage de données
  - simples fichiers (ex. configuration)
  - bases de données semi-structurées (requêtes, etc.)
  - bases de données documentaires
    - documents XML
    - documents XHTML (web)
- Echange de données
  - d'une base de données vers une autre (format d'échange)
  - serveur vers un navigateur : données + feuille de style
- Remarque :
  - circulation de flux XML sur un réseau :
    - utilisation de l'arbre entier (le document)
    - utilisation à la volée pour les très gros documents (exemple : BiM)

Retour sur la représentation de documents

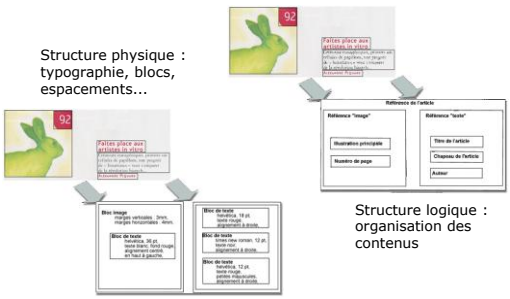
38

- Document numérique
  - manipulations et gestion par des ordinateurs
- Document structuré
  - séparation structure physique / structure logique
  - séparation forme / contenu
- D'où possibilité
  - de manipuler la structure logique des documents
  - d'accéder au texte des différentes parties logiques des documents
  - de générer plusieurs structures physiques à partir d'une structure logique

Structures logique / physique

39

Structure physique : typographie, blocs, espacements...



Structure logique : organisation des contenus

(d'après http://sophia.univ-lyon2.fr/didacticiel/unite1/module2.html)

Balisage de texte

40

- Idée
  - marquer des zones des textes pour les qualifier
    - les balises ouvrantes et fermantes délimitent les éléments de description
    - la structure logique est un arbre « ajouté » au texte

```
<p>Il est de tradition de présenter un langage de programmation à l'aide d'un premier exemple comme : <eg> CHAR*20 GRTG GRTG = 'BONJOUR TOUT LE MONDE' PRINT *, GRTG END </eg></p>
<p>Dans cet exemple, on commence par déclarer la variable <ident>GRTG</ident>, dans la ligne <kw>CHAR*20 GRTG</kw>, qui identifie <ident>GRTG</ident> comme formée de 20 octets de type <kw>CHAR</kw>. On affecte alors à cette variable la valeur <mentioned>BONJOUR TOUT LE MONDE</mentioned>. Suivent alors l'ordre d'impression <kw>PRINT</kw> et l'instruction finale <kw>END</kw>.</p>
```

p : servira à la mise en page

eg, kw,mentioned : seront mis en évidence dans la structure physique

kw,mentioned : utilisés pour construire un index

etc.

Conclusion

41

- Notions d'arbre et méta-langage XML à la base de ce cours
- Utilité
  - pour le Web, l'échange de données, le tagging de documents...
- Suite du module
  - XML : 3 CM / 2 TP
  - (X)HTML / CSS : 2 CM / 2 TP
  - XPATH, XSL : 2 CM / 2 TP
  - Projet
  - Tutorat