

Introduction à XML

Par Yves Bekkers

Mise à jour : 4 décembre 2006

XML - Yves Bekkers - IFSIC

1

Plan

- Balisage textuel
- Un peu d'histoire
- XML - concepts de base
- Outils de la galaxie XML
- Exemples d'applications
- Conclusions

XML - Yves Bekkers - IFSIC

2

Balisage textuel

XML - Yves Bekkers - IFSIC

3

Le balisage une longue histoire

- Balisages des manuscrits pour l'imprimerie
- Balisages des documents numériques
 - Photocomposition
 - Unix : troff, Latex
 - GML - SGML

XML - Yves Bekkers - IFSIC

4

Des manuscrits pour l'imprimerie

- **Balisage** : processus par lequel un rédacteur introduisait des marques (balises) dans un manuscrit pour indiquer à l'imprimeur comment composer (formater) le manuscrit.
- **Balise** : généralement, commentaire écrit à la main :
« composer le titre avec la police Hevetica-médium-Italic corps 12 et avec un interlignage de 14 points, justifié sur une largeur de 22 picas avec renforcement de un demi cadratin à gauche et de zéro à droite »

XML - Yves Bekkers - IFSIC

5

Le balisage de texte numériques avant les réseaux

XML - Yves Bekkers - IFSIC

6

Photo-composition

le langage « Nortext-100 » de Norsk Data

```
<CC 15,5,12>On demande la police numéro 5,  
en corps 12 et avec une justification de  
15 unités.  
<SS><QL> <CC 20,8>Maintenant. on prend la  
police 8 et une justification de 20  
unités.  
<QL> <RS>Et maintenant on repart avec les  
valeurs typographiques initiales.<EP>
```

Troff

Une commande Unix pour le formatage de texte

```
.\ " @(#)troff.1 1.66 95/10/09 SMI;  
.TH troff 1 "6 Oct 1992"  
.SH NAME  
troff \- typeset or format documents  
.SH SYNOPSIS  
.B troff
```

Tex & LaTeX

- *Quoi* : langage pour décrire des documents imprimés de qualité
- *Par qui* : Donald Knuth
- *Quand* : 1970-1980

```
\noindent  
Ce texte comporte 3 pages\hfill  
\vspace*{1cm}  
\section{Analyse syntaxique et les DCG}  
La syntaxe BNF offre des opérateurs tels que  
``|'' qui signifie un choix parmi
```

Balilage de documents numériques pour le WEB

Exemples

Visualiser une carte de visite

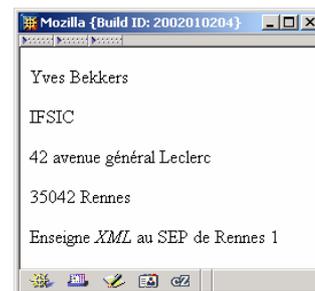
Exemple

```
Yves Bekkers  
IFSIC  
42 avenue général Leclerc  
35042 Rennes  
Enseigne XML au SEP de Rennes 1
```

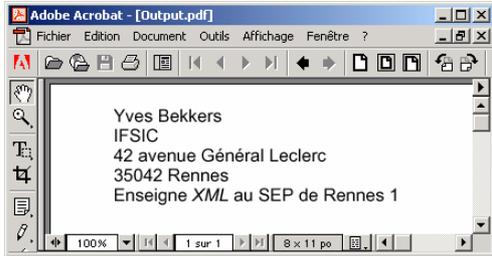
- Trois manières de visualiser ce contenu sur le WEB
 - Sur un écran : HTML
 - Par impression : PDF
 - Graphique : SVG

HTML

(visualisé par Mozilla)



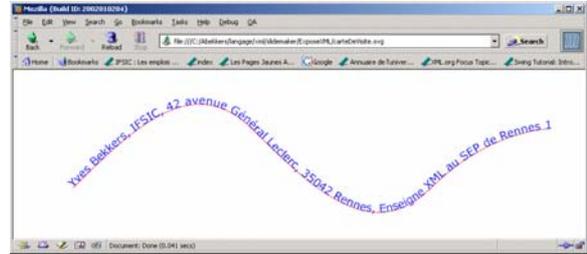
PDF (visualisé par Acrobat)



XML - Yves Bekkers - IFSIC

13

Graphe SVG (visualisé par Mozilla + Adobe SVG-Viewer)



XML - Yves Bekkers - IFSIC

14

Question-réponse

- Question
Qu'est ce qui relie ces trois exemples ?
- Réponse
Documents structurés
Les langages à balises

XML - Yves Bekkers - IFSIC

15

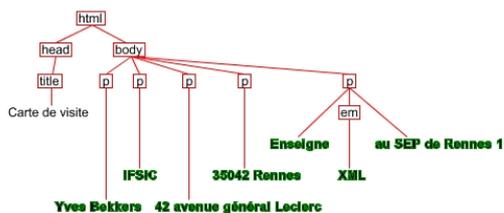
Carte de visite en html

```
<html>
  <head><title>Carte de visite</title></head>
  <body>
    <p>Yves Bekkers</p>
    <p>IFSIC</p>
    <p>42 avenue général Leclerc</p>
    <p>35042 Rennes</p>
    <p>Enseigne <em>XML</em>
      au SEP de Rennes 1</p>
  </body>
</html>
```

XML - Yves Bekkers - IFSIC

16

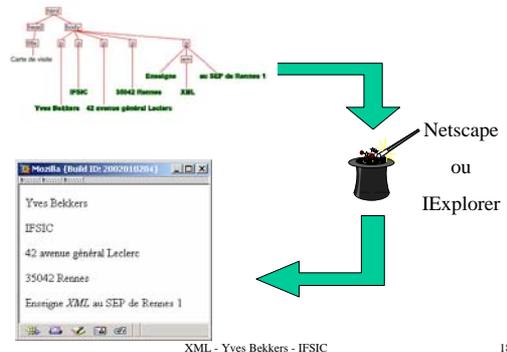
Arbre - HTML



XML - Yves Bekkers - IFSIC

17

Utilisation du format HTML



XML - Yves Bekkers - IFSIC

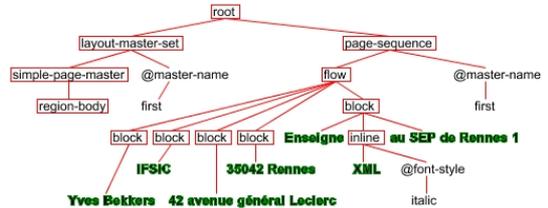
18

Format imprimable Xsl-FO

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="first" margin-left="2cm"
      margin-right="2cm" margin-bottom="0.5cm" margin-top="0.75cm">
      <fo:region-body />
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-name="first">
    <fo:flow flow-name="xsl-region-body" font-size="16pt">
      <fo:block>Yves Bekkers</fo:block>
      <fo:block>IFSIC</fo:block>
      <fo:block>42 avenue Général Leclerc</fo:block>
      <fo:block>35042 Rennes</fo:block>
      <fo:block>Enseigne <fo:inline font-style="italic">
        XML</fo:inline> au SEP de Rennes 1</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

19

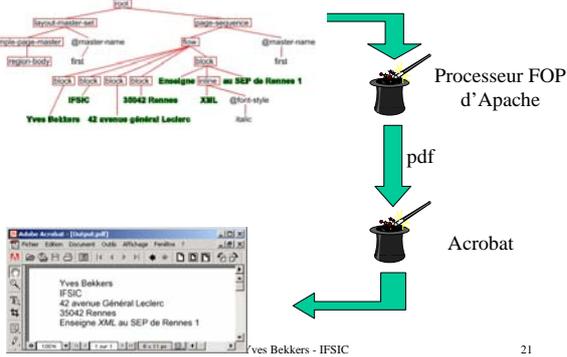
Arbre - format FO



XML - Yves Bekkers - IFSIC

20

Utilisation du format FO



21

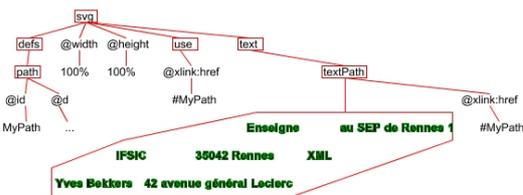
Carte de Visite graphique en SVG

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%"
  height="100%">
  <defs><path id="MyPath" d="M 100 200
    C 200 100 300 0 400 100
    C 500 200 600 300 700 200
    C 800 100 900 100 900 100"/></defs>
  <use xlink:href="#MyPath" fill="none" stroke="red"/>
  <text font-family="Verdana" font-size="19" fill="blue">
    <textPath xlink:href="#MyPath">
      Yves Bekkers, IFSIC, 42 avenue Général Leclerc,
      35042 Rennes, Enseigne XML au SEP de Rennes 1
    </textPath>
  </text>
</svg>
```

XML - Yves Bekkers - IFSIC

22

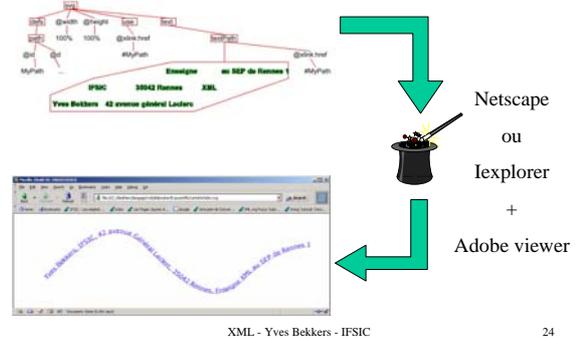
Arbre SVG



XML - Yves Bekkers - IFSIC

23

Utilisation du format SVG



XML - Yves Bekkers - IFSIC

24

Une même information trois structures

XML - Yves Bekkers - IFSIC

25

A propos de ces trois exemples

- Quatre outils
 - Netscape
 - IExplorer
 - Processeur FOP d'Apache
 - Adobe SVG viewer
- Le problème :
 - être en mesure de différencier la *forme* du *fond* au sein d'un document
- La solution : **les langages à balise**

XML - Yves Bekkers - IFSIC

26

Trois langages à balises trois dialectes xml

xhtml



Xsl-FO



svg



27

Un peu d'histoire

SGML, HTML les deux précurseurs d'XML

XML - Yves Bekkers - IFSIC

28

GML première normalisation du balisage

- *Quoi* : unifier trois systèmes disparates
 - une application d'édition de texte,
 - une application de composition
 - une base pour l'interrogation documentaire.
- *Par qui* : Charles Goldfarb
- *Où* : IBM, New York
- *Pour qui* : un consortium d'avocats
- *Quand* : 1969

XML - Yves Bekkers - IFSIC

29

GML : comment

- Des documents contenant un mélange harmonieux :
 - d'information et
 - de méta-information de structure
- Documents compréhensibles par les machines
- Documents textuels faciles à mettre en œuvre
 - éditables à l'aide de n'importe quel éditeur de texte

XML - Yves Bekkers - IFSIC

30

GML : comment (bis)

- Comme les bases de données
 - On peut rechercher l'information par son contenu
 - On peut étiqueter l'information par « identificateur unique »

Un exemple de dialect GML (1978)

DCF (Document Composition Facility), un ancêtre de html

Le premier dialect GML à qui Goldfarb va donner une DTD.

```
:h1.Chapter 1: Introduction
:p.GML supported hierarchical containers, such as
:ol.
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
as well as simple structures.
:p.Markup minimization (later generalized and
  formalized in SGML),
  allowed the end-tags to be omitted for the "h1" and
  "p" elements.
```

GML un énorme succès

- La mise en oeuvre est générique, elle s'applique à d'autres types de documents
 - Exemple : documenter un porte avion pour le DOD ...
- 1986 norme ISO (ISO 8879) pour Standard GML (SGML)

Naissance d'HTML

- *Contexte* : 1991, généralisation des réseaux
- *Problème* : afficher en n'importe quel point d'un réseau de l'information conservée de manière répartie sur divers machines du réseaux
- *Qui* : Tim Berners-Lee, CERN

Solution au problème de « publication » sur réseau (1)

- *Contexte* : déjà 15 ans d'expérience SGML
- On conçoit un dialecte de SGML pour décrire dans un même document l'information et son mode de présentation
- La simplicité du langage garantit qu'on peut réaliser rapidement sur toute machine des processeurs de présentation des documents :
naissance des « navigateurs »

Solution au problème de « publication » sur réseau (2)

- *Contexte* : SGML ne prend pas en compte la dimension réseau
- *Solution* : On étend le dialecte SGML par un système d'adressage universel des documents à travers les réseaux
 - naissance des « liens hypertexte » ou « URLs »

Solution au problème de « publication » sur réseau (3)

- *Contexte* : De nouveaux types de documents tels que des images et du son, ont fait leur apparition.
- *Solution* : On étend les possibilités d'affichage des navigateurs
 - naissance du « multimédia »

XML - Yves Bekkers - IFSIC

37

HTML - les avantages

- Un langage à balise issu de SGML
- permettant de transporter en même temps l'information et la méta-information qui décrit sa présentation
- Accès réparti sur le réseau grâce aux liens hypertexte
- Extension de la notion d'affiche à d'autres documents que les textes, introduction du multimédia

XML - Yves Bekkers - IFSIC

38

HTML un énorme succès mais ...

- Succès repris par *Le World Wide Web Consortium (W3C)*
- structuration limitée
 - liée à l'édition sur écran
- cache la structure logique
- liens hypertexte trop pauvres
- méta-données pas assez riches
- simple ==> compliqué
 - plus de 100 balises normalisées, foisonnement des balises propriétaires

XML - Yves Bekkers - IFSIC

39

Réflexions issues de l'expérience HTML

- *Le World Wide Web Consortium (W3C)*
Créer en 1996 d'un groupe de réflexion sur SGML et HTML
- *Idée* : recherche d'un schéma général de langage pour la structuration de l'information et sa communication sur le WEB
- *Intérêt* : mise en commun des efforts
 - par des outils
 - par des normes complémentaires

XML - Yves Bekkers - IFSIC

40

Conclusions du groupe de Travail

- Résultats donnés en 1998
- Prendre à SGML
 - sa simplicité
 - l'accessibilité à l'information
 - la généricité (pas de balise prédéfinie)
- Prendre à HTML
 - liens hypertexte en les généralisant
 - notion de document multimédia
 - La notion de méta-données en la généralisant

XML - Yves Bekkers - IFSIC

41

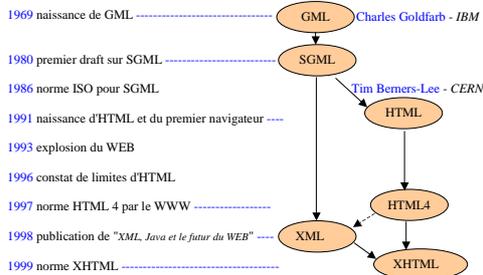
HTML versus XML

- HTML publication hypertexte sur écran
- XML généricité + échange de données
 - pas de balises normalisées, l'auteur crée ses propres balises
 - balisages plus strict pour simplifier le traitement des documents
 - la casse des caractères dans les noms d'éléments est significative
 - Structurer l'information selon sa propre logique

XML - Yves Bekkers - IFSIC

42

Les dates



43

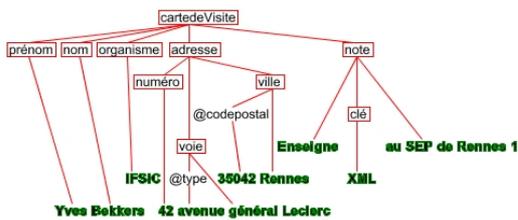
Carte de visite logique

```
<carteDeVisite>
  <prénom>Yves</prénom>
  <nom>Bekkers</nom>
  <organisme>IFSIC</organisme>
  <adresse>
    <numéro>42</numéro>
    <voie type="avenue">Général Leclerc</voie>
    <ville codepostal="35042">Rennes</ville>
  </adresse>
  <note>
    Enseigne <clé>XML</clé> au SEP de Rennes 1
  </note>
</carteDeVisite>
```

XML - Yves Bekkers - IFSIC

44

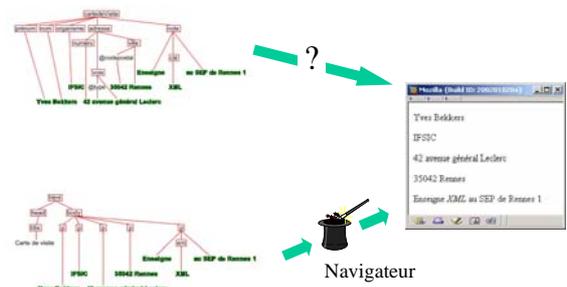
Arbre logique



XML - Yves Bekkers - IFSIC

45

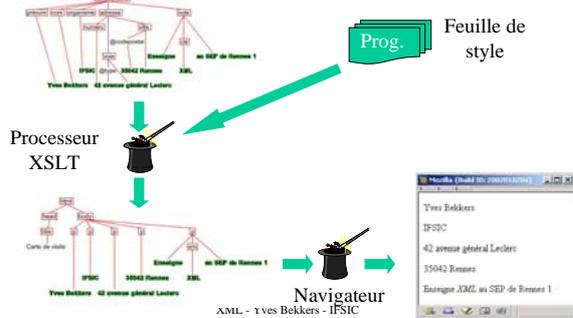
Le prix d'un arbre logique (1)



XML - Yves Bekkers - IFSIC

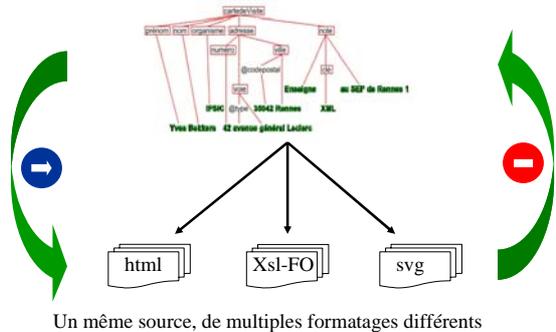
46

Le prix d'un arbre logique (2)



XML - Yves Bekkers - IFSIC

Intérêt d'un arbre logique



Un même source, de multiples formatages différents

XML - Yves Bekkers - IFSIC

48

XHTML

- XHTML = HTML + XML
 - plus qu'un simple changement de syntaxe
 - production automatique de table des matières
 - production automatique de multiples versions
 - combinaison avec d'autres documents (MathML, SVG, ...)
 - modularité (à venir)
- éditer du xhtml : Amaya
- produire du xhtml avec du html : Tidy

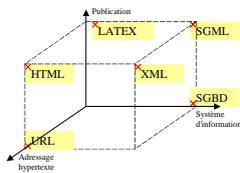
XHTML versus XML

- XHTML un dialecte XML dédié à la présentation


```
<body>
<p><b>Titre</b> : XSLT : Programmer's Reference</p>
<p><b>Auteur</b> : Michael Kay</p>
<p><b>Pages</b> : 392</p>
<p><b>Editeur</b> : Wrox</p>
<p><b>Série</b> : Programmer to Programmer</p>
</body>
```
- XML des dialectes qui permettent la recherche par le contenu

```
<livre>
<titre>XSLT : Programmer's Reference</titre>
<auteur>Michael Kay</auteur>
<pages>392</pages>
<editeur>Wrox</editeur>
<serie>Programmer to Programmer</serie>
</livre>
```

Les trois dimensions de XML



XML

eXtensible Markup Language

Concepts de base

Élément

Un « nom » + des « attributs » éventuels + un « contenu »

... <nom attr1="valeur" ...> contenu </nom> ...

Balise de début avec un nom + balise de fin avec le même nom

2 Exemples :

```
<p>
  Yves Bekkers
</p>
```

```
<personne>
  <prénom>Yves</prénom>
  <nom>Bekkers</nom>
</personne>
```

La forme + le fond

Balise d'ouverture	<p>
Contenu	Yves Bekkers
Balise de fermeture	</p>

A l'image des SGBD, un document XML contient le texte et son référent

Contenu d'élément

- Quatre types de contenus

Texte	<code><p></code> Yves Bekkers <code></p></code>
Eléments	<code><body></code> <code><p>...</p></code> ... <code><p>...</p></code> <code></body></code>
Contenu mixte	<code><p></code> enseigne <code>XML</code> au SEP de Rennes1 <code></p></code>
Vide	<code>
</br></code>

XML - Yves Bekkers - IFSIC

55

Éléments à contenu vide

`
</br>`

Raccourci autorisé

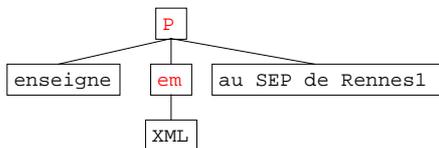
`
`

XML - Yves Bekkers - IFSIC

56

Construire des arbres

```
<p>
enseigne
  <em>XML</em>
au SEP de Rennes1
</p>
```



XML - Yves Bekkers - IFSIC

57

Attribut

- *Quoi* : un nom + une valeur
- *Où* : au sein de balise ouvrante d'élément
- *Signification* : qualifie l'élément
- *Exemple* :

```
<fo:inline font-style="italic">
XML
</fo:inline>
```

XML - Yves Bekkers - IFSIC

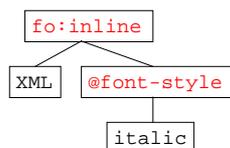
58

Attribut (bis)

- *Exemple d'élément* :

```
<fo:inline font-style="italic">
XML
</fo:inline>
```

et son arbre :



XML - Yves Bekkers - IFSIC

59

Documents bien formés

XML - Yves Bekkers - IFSIC

60

Structure d'un document XML

- Un Prologue (éventuellement)
- Un élément racine (et un seul)
 - arbre d'éléments et leurs attributs
- Des Commentaires
- Des instructions de traitement

XML - Yves Bekkers - IFSIC

61

Structure d'un document - exemple

```

<?xml version="1.0" encoding="ISO-8859-1"?> } Prologue
<!-- ceci est une carte de visite --> } Commentaire
<carteDeVisite> } Élément racine
  <prénom>Yves</prénom>
  <nom>Bekkers</nom>
  <adresse>
    <numéro>42</numéro>
    <voie type="avenue">général Leclerc</voie>
    <ville codepostal="35042">Rennes</ville>
  </adresse>
  <note>
    Enseigne <clé>XML</clé> au SEP de Rennes 1
  </note>
</carteDeVisite>
    
```

XML - Yves Bekkers - IFSIC

62

Document bien formé

Règle : *Un document XML comporte un seul élément racine*

```

<?xml version="1.0"?>
<livre>XLSST Programmer's reference</livre>
<livre>XML et Java</livre>
    
```

Est interdit

Règle : *Toute balise ouvrante doit avoir sa balise fermante*

 sans </br> est interdit en XML

XML - Yves Bekkers - IFSIC

63

Document bien formé (bis)

Règle : *Le parenthésage doit être correct*

```
<a><b></b></a>
```

est bien formé, équivalent à

```
[ ( ) ]
```

```
<a><b></a></b>
```

est mal formé, équivalent à

```
[ ( ] )
```

XML - Yves Bekkers - IFSIC

64

Syntaxe des noms d'élément (exemples)

- nom
- prénom
- à-la-ligne
- xsl:value-of
- n123
- décompte.client
- first_name

XML - Yves Bekkers - IFSIC

65

Syntaxe des noms d'éléments (bis)

Name ::= (Letter | '_' | ':') (NameChar)*

NameChar ::= Letter | Digit | '.' | '-' | '_' | ':'

Digit ::= 0 | 1 | ... | 9

Letter *Toutes les lettres de tous les alphabets (par exemple les lettres accentuées de l'alphabet latin)*

XML - Yves Bekkers - IFSIC

66

Syntaxe des noms d'éléments (ter)

Règle : Les majuscules et les minuscules sont différenciées
(plus stricte qu'HTML)

Notation des valeurs d'attributs

- Règle : Les valeurs d'attribut doivent être encadrées par des doubles cotes " ou des simples cotes '
- mais
``
- est interdit (plus stricte qu'HTML)

Attribut

- Règle : Un attribut pour un élément donné ne peut avoir qu'une seule valeur
- Exemple
``
...
``
- est interdit ...

Composants complémentaires

- Entités (macro)
 - déclaration `<!ENTITY lp "lambdaProlog">`
 - référence `&lp;`
- Commentaires
`<!-- ceci est un commentaire -->`
- Instructions de traitement
`<?nom arg1 arg2 ... argn ?>`
- Sections littérales (échappements)
`<![CDATA[if i < 2 then i++]]>`

Instructions de traitement Exemples

- Schéma d'une instruction de traitement
`<?nom arg1 arg2 ... argn ?>`
- Appel d'un processeur XSLT
`<?xml-stylesheet type="text/xsl" href="myXsl.xslt" ?>`
- Référence à une feuille de style CSS
`<?xml-stylesheet type="text/css" href="myCss.css" ?>`

Résumé

- Un document XML est composé de
 - Éléments
 - Texte
 - Autres éléments
 - Mixte (texte + éléments)
 - Vide
 - Commentaires
 - Instructions de traitement
 - Sections littérales

Exemple complet

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- ceci est un document XML -->
<?xml-stylesheet type="text/css" href="prog.css"?>
<programme id="pl">
  <!-- Instruction conditionnelle -->
  <instruction>
    <test><![CDATA[x < 2]]></test>
    <instruction>x = x +1</instruction>
  </instruction>
</programme>
```

- **Attention**

La première ligne ne fait pas partie du document, elle est *consommée* au moment de la lecture du document par les applications

Codage

Codage

- Deux niveaux de codage
 - codage externe : dépendant des outils de saisie
 - codage interne : Unicode



- Détection semi automatique du type de codage externe
 - s'aide du premier caractère de la première ligne : "<"

Unicode

- Codage sur 16 bits (au début) 21 bit actuellement
- Ambitieux : toutes les langues de la terre
- Unification : éliminer les redondances
- Sépare la définition du jeu de caractères (la liste des caractères, leur nom et leur index) de celle de l'encodage
 - on ne peut donc pas parler de la taille d'un caractère Unicode (dépend de l'encodage choisi)
- Prés de 100000 caractères
 - caractères composites
- Convertibilité
 - Ascii dans le bas du tableau
- >1000 pages de définitions ...

Composition de caractères

- Unicode ne code pas les descriptions des caractères, les *glyphes* c'est-à-dire la représentation graphique du caractère.
- ainsi, le caractère français *é* peut-il être décrit de deux manières :
 - en utilisant directement le numéro correspondant au *é*,
 - en faisant suivre le numéro du 'e' par celui de l'accent aigu sans chasse.
 - Quelle que soit l'option choisie le même glyphe sera affiché.
 - On dira du premier caractère qu'il est précomposé, du second que c'est une composition (deux caractères forment un seul glyphe composé des deux).
- De nombreux glyphes peuvent être codés de ces deux manières. Le plus souvent, le glyphe précomposé est préférable

Partionnement

```
0000..007F; Latin de base
0080..00FF; Supplément Latin-1
...
0370..03FF; Grec et copte
0400..04FF; Cyrillique
0500..052F; Supplément cyrillique
0530..058F; Arménien
0590..05FF; Hébreu
0600..06FF; Arabe
...
1780..17FF; Khmer
1800..18AF; Mongol
...
10100..1013F; Nombres égéens
...
1D000..1D0FF; Symboles musicaux byzantins
1D100..1D1FF; Symboles musicaux occidentaux
...
```

Codage externe

- Dépend de l'outil utilisé pour la saisie du document et du système qui le supporte :
 - Word sous windows
 - Emacs sous linux
 - ...

Codages externes - exemples

- Ascii ou Iso 646 (adopté en 1963)
 - Codage à 7 bits ($2^7 = 128$ caractères)
- Latin 1 (ISO-8859-1)
 - Codage à 8 bits ($2^8 = 256$ caractères), les 128 premiers caractères sont ceux d'Ascii (manque le œ)
- Latin 9 (ISO-8859-15)
 - Codage à 8 bits ($2^8 = 256$ caractères), légère modification de Latin 1
 - apparition de : œ, Œ, et €

UTF Unicode Transformation Format

- Unicode accepte plusieurs formes de présentation pour représenter un caractère (au sens de *symbole*).
- UTF8 (le plus courant)
 - codage sur une séquence d'octets (1 à 4)
 - nombreuses propriétés intéressantes pour les logiciels de traitement
 - ASCII sur un seul caractère
 - Chaque octet à une nature qui facilite la reconnaissance du multi-octet
- UTF16 (java, Windows)
 - codage sur une séquence d'octets (2 ou 4)
- UTF32
- Le chiffre après UTF représente le nombre minimal de bits avec lequel un caractère est codé.

Exemple - le symbole Euro

Glyphe (œil)	€
Code Unicode	20AC
Latin 9	A4
UTF8 (3 octets)	11100010 10000010 10101100
UTF16	20AC

Désigner un caractère par son code

- Notation qui permet de désigner un caractère par son code unicode

	Référence	Caractère
En décimal	î	î
	∀	∅
En hexadécimal	Œ	Œ
	©	©

- Exemple d'utilisation : "le gîte et le couvert"
`<titre>le gîte et le couvert</titre>`

Caractères spéciaux

- Cinq entités prédéfinies

Référence	Glyphe	Nom
&	&	ampersand
<	<	plus petit
>	>	plus grand
'	'	apostrophe
"	"	double quote

- Exemple d'utilisation : "A >5"
`<if>A > 5</if>`

Notion de dialects XML

Généricité d'XML

- XML comme SGML n'est pas un langage
 - c'est un cadre pour définir des langages
- Chaque application définit son (ses) dialect(s)
 - Un dialect = un type de document
- Il existe des grammaires pour définir les dialects
 - DTD, Schéma XML, ...

Contexte	Méta données	Information
SGBDR	Définition de schéma	Instance de table
Programmation objet	Définition de classe	Instance de classe
XML	DTD, XMLSchema	Document XML

Exemple de DTD

```
<!ELEMENT carteDeVisite (prénom, nom, organisme, adresse)>
<!ELEMENT prénom (#PCDATA) >
<!ELEMENT nom (#PCDATA) >
<!ELEMENT organisme (#PCDATA) >
<!ELEMENT adresse (numéro, voie, ville)>
<!ELEMENT numéro (#PCDATA) >
<!ELEMENT voie (#PCDATA) >
<!ATTLIST voie type (rue | avenue | impasse | cours |
    square | boulevard | chemin |
    quai | route | passage | place |
    rondPoint ) 'rue' >
<!ELEMENT ville (#PCDATA) >
<!ATTLIST ville codepostal CDATA #IMPLIED >
```

Déclaration de DTD

- Dans le prologue du document
- Déclaration interne


```
<!DOCTYPE carteDeVisite [
... déclarations ...
]>
```
- Déclaration externe, la chaîne est une URL

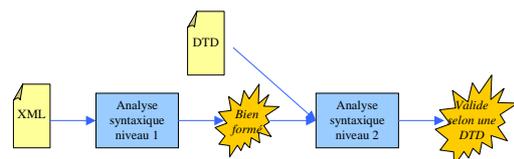

```
<!DOCTYPE carteDeVisite SYSTEM "maDTD.dtd">
```
- Déclaration mixte


```
<!DOCTYPE carteDeVisite SYSTEM "maDTD.dtd" [
... déclarations ...
]>
```

Déclaration externe public

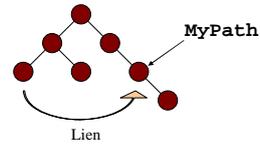
- Exemple
 - déclaration pour un document SVG 1.0
- ```
<!DOCTYPE svg PUBLIC
"-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```
- La première chaîne est une identification
    - Le système cherche une correspondance dans des tables internes
  - La seconde chaîne est une URL
    - à utiliser en cas de défaut de la première

## Validation de document



## Structurer en graphe

## Liens internes



```
<path id="MyPath" d="M 100 200
C 200 100 300 0 400 100
C 500 200 600 300 700 200
C 800 100 900 100 900 100"/>
...
<use xlink:href="#MyPath"
fill="none" stroke="red"/>
```

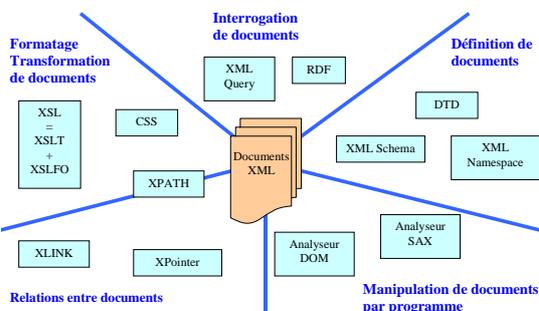
## Modularité des documents Liens externes

```
<livre>
 <chapitre href="chap1.xml"/>
 <chapitre href="chap2.xml"/>
 <chapitre href="chap3.xml"/>
</livre>
```

- Document composé de plusieurs documents
- Héritage d'HTML

## La galaxie XML

## Galaxie des normes par catégories



## Quelques détails sur principaux composants de la galaxie XML

## Visualisation de documents

- Exemple : visualiser un carnet d'adresses
  - Le document [carnetDAdresse.xml](#)
- Deux possibilités
  - Appliquer une feuille de style CSS au document XML
    - [Appliquer](#)
  - Transformer le document xml en html à l'aide d'XSLT
    - [Transformer](#)

XML - Yves Bekkers - IFSIC

97

## Feuille de style CSS

- CSS *Cascading Style Sheets*
  - Conçu pour les langages à balise (HTML, XML)
  - Permet d'associer des *propriétés de présentation* à des balises
  - Partage : peuvent être écrites dans une page séparée.  
`<?xml-stylesheet type="text/css" href="fic.css" charset="ISO-8859-1" ?>`
- Exemple de définition de propriété

```
prénom,organisme,note {
 text-align: left; margin-top: 12.000000pt;
 margin-bottom: 6.000000pt;
 font-size: 12.000000pt; font-weight: medium;
 font-style: Regular; color: #000000;
 font-family: "Times New Roman";
}
```

XML - Yves Bekkers - IFSIC

98

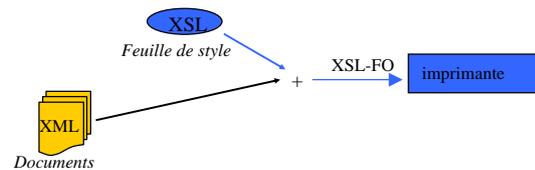
## Transformation XSL

- XSL (*eXtensible Stylesheet Language*)
- Le langage « *Perl* » d'XML
- Trois parties
  - Xpath : langage pour adresser les nœuds d'un arbre
  - XSLT : langage de transformation
  - XSL-FO : langage de formatage

XML - Yves Bekkers - IFSIC

99

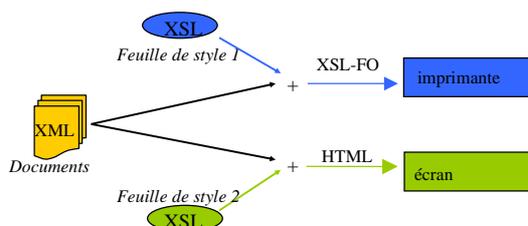
## Transformation de document (1)



XML - Yves Bekkers - IFSIC

100

## Transformation de document (2)



XML - Yves Bekkers - IFSIC

101

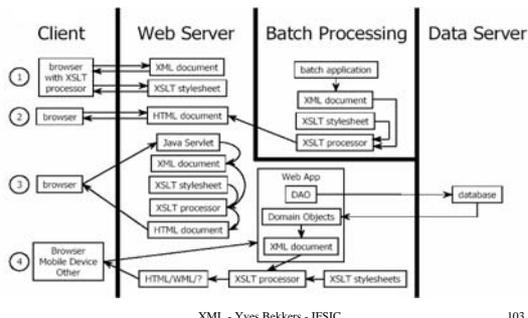
## Limites de CSS par rapport à XSLT

- CSS ne permet pas de visualiser les attributs
- CSS ne permet pas de réarranger l'information
- CSS n'offre pas de vrai moyen de faire des calculs
- La cible ne peut pas être un autre dialecte XML (CSS est un outil de présentation)

XML - Yves Bekkers - IFSIC

102

## Quatres moyens d'utiliser une feuille de style XSLT



XML - Yves Bekkers - IFSIC

103

## XML NameSpace « Espaces de noms »

- **Objectif** : faire cohabiter au sein d'un même document des éléments appartenant à des dialectes différents
- **Comment** : Mise en place d'un mécanisme de préfixes de nom
  - Déclaration de préfixe au sein d'une balise ouvrante
    - `<p xmlns:M='http://www.w3.org/REC-MathML'>`
  - Utilisation de préfixe
    - `<M:mfrac>`

XML - Yves Bekkers - IFSIC

104

## Méta-données

- Des données à propos des données, pour décrire l'information
- **RDF** : *Resource Description Framework*
- Lisible à l'œil, compréhensible par les machines
- **Norme générique** : propriétés (ou relations) à définir pour chaque domaine d'application
- **Applications** : cartes de site, étiquetage, recherche d'information, bibliothèques électroniques, édition coopérative, propriété intellectuelle, commerce électronique, etc.

XML - Yves Bekkers - IFSIC

105

## Exemples d'application

- **SVG** Graphiques structurés, vectorisés
- **SMIL** Synchronisation de documents multimédia
- **MathML** Manipulation de formules mathématiques
- **XUL** Le langage d'interface de Mozilla  
[exemples.html](http://exemples.html)

XML - Yves Bekkers - IFSIC

106

## Editeurs

- Conception d'applications
  - Emacs avec mode sgml
  - XML spy (très cher, Windows uniquement)
  - Eclipse avec le plugin XMLBuddy, Bocaloco Software
  - Eclipse WTP
  - Amaya (XML, mathML, SVG, XHTML)
- Exploitation :
  - éditeurs graphiques qui masquent le code
  - Exemples : White all, Xmetal, XML spy

XML - Yves Bekkers - IFSIC

107

## Conclusions

XML - Yves Bekkers - IFSIC

108

## Qu'est ce qu'XML

- XML est une méthode pour conserver des données structurées dans des fichiers texte
- XML est une technologie (famille de norme, d'outils, espace ouvert pour définir des dialectes propriétaires)
- XML ressemble à HTML mais ne remplace pas HTML
- XML ressemble aux SGBD mais ne remplace pas un SGBD
- XML est un texte (pas toujours fait pour un humain)
- XML est bavard (mais cela n'est pas un problème)
- XML n'est pas si nouveau que ça

XML - Yves Bekkers - IFSIC

109

## A propos de vocabulaire

- Les mots *document* et *feuille de style* sont hérités de l'édition
  - GML était un projet d'édition
  - HTML est un langage d'édition
- Les mots sont trompeurs
  - XML est un système d'information plus qu'un système d'édition
  - On peut interroger un document par son contenu, le transformer

XML - Yves Bekkers - IFSIC

110

## Similitude avec les SGBDR

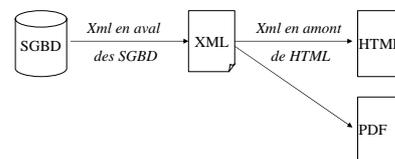
	SGBDR	XML
<b>Structure</b>	Relation (table)	Arbre
<b>Méta information</b>	Langage de définition	DTD - XML Schéma
<b>Information</b>	Contenu de table	Instance de document
<b>Contrainte d'unicité</b>	Clé primaire	Attribut ID
<b>Intégrité de référence</b>	Clé étrangère	Attribut IDREF
<b>Langage de manipulation</b>	SQL	XSLT

XML - Yves Bekkers - IFSIC

111

## SGBD-XML-HTML

- SGBD, XML, HTML ne sont pas des technologies concurrentes
  - Elles sont complémentaires



XML - Yves Bekkers - IFSIC

112

## Multiples applications d'XML

- diffusion de contenu sur le web
  - complément à HTML, à CSS, aux SGBD ...
- échange entre applications distribuées
  - EDI, Ebisness, ...
- archivage :
  - fichiers de configuration
  - information au faible dynamisme (annuaires)
- gestion de la connaissance

XML - Yves Bekkers - IFSIC

113

## Apport d'XML

- Gains qualitatifs
  - **robustesse** :
    - force à l'abstraction
    - un unique niveau d'abstraction
  - **partage des efforts** : mutualisation des outils
  - **technologie ouverte** : qui va traverser le temps
  - **simplicité** : texte éditable
- Gains quantitatifs
  - mise en œuvre rapide car elle est directe

XML - Yves Bekkers - IFSIC

114

## Quelques références

- Le site XML du W3C  
<http://www.w3.org/XML/>
- Un site francophone XML  
<http://www.chez.com/xml/>
- Un glossaire XML  
<http://www.xmlstarterkit.com/xmlzone/glossary.htm>
- Une histoire des langages à balise  
<http://www.gutenberg.eu.org/pub/GUTENBERG/publications/cahiers.html#Cahier12>

## Livre en français (*concis*)

- *XML précis et concis*  
– O'Reilly, par Robert Eckstein et Michel Casabianca