

# XML et la documentation

Code: xml-docu

## Originaux

[url: http://tecfa.unige.ch/guides/tie/html/xml-docu/xml-docu.html](http://tecfa.unige.ch/guides/tie/html/xml-docu/xml-docu.html)

[url: http://tecfa.unige.ch/guides/tie/pdf/files/xml-docu.pdf](http://tecfa.unige.ch/guides/tie/pdf/files/xml-docu.pdf)

## Auteurs et version

- Daniel K. Schneider
- Version: 0.3 (modifié le 5/12/05) - brouillon

## Prérequis

Module technique précédent: xml-tech

Module technique précédent: xml-xslt

## Abstract

XML dans le monde de la documentation

Ce document sert de fil rouge pour une série de démonstrations. "Tel quel" son utilité est un peu limitée encore ...

## **Objectifs**

- Connaître quelques éléments de la galaxie "XML + Documentation"
- Connaître quelques solutions

# 1. Table des matières détaillée

1. Table des matières détaillée.....	3
2. Introduction.....	4
<b>2.1 Caractéristiques de base d'un "document" SGML/XML</b>	<b>5</b>
<b>2.2 Edition de textes en XML</b>	<b>6</b>
<b>2.3 Vocabulaires pour "publishing"</b>	<b>7</b>
<b>2.4 Rappel de la philosophie XML</b>	<b>8</b>
3. Processus de traitement.....	9
<b>3.1 Filtres</b>	<b>10</b>
A.Filtres XML -> HTML 10	
B.Filtres Base de donnée -> XML -> XML 10	
C.Filtres XML-> XML -> PDF 10	
<b>3.2 Documents XML pour Internet</b>	<b>11</b>
A.Navigateurs 11	
B.Applets Java et autres outils de visualisation de contenus 11	
C.Server-Side: marche sans problème! 11	
<b>3.3 Note sur XSL-FO</b>	<b>12</b>
4. Langages auxiliaires.....	13
<b>4.1 Xpointer</b>	<b>13</b>
<b>4.2 XInclude</b>	<b>14</b>
<b>4.3 Note: XML de base possède un simple système d'inclusion</b>	<b>16</b>
<b>4.4 MathML</b>	<b>17</b>
<b>4.5 XLink</b>	<b>18</b>
5. Notes sur XHTML.....	19
6. Docbook.....	22
7. DITA.....	24
8. Grammaires sur mesure.....	27

## 2. Introduction

### Historique

- Le formalisme XML a été inspiré par le langage SGML
- Le SGML - Standard Generalized Markup Language (langage normalisé de balisage généralisé) - est un langage de description à balises, de norme ISO (ISO 8879:1986)
  - SGML a été adopté par les industries qui doivent produire des très gros documents et/ou qui doivent les garder pendant longtemps et/ou qui veulent que le contenu soit "machine readable"
  - Exemples: IBM, Boeing, Airbus, etc.
  - Il existe pleins d'applications propriétaires (chez IBM par exemple), mais le langage le plus populaire pour écrire des documents est DocBook
  - Une autre application SGML connue est HTML

### Formalismes XML majeurs

- DocBook (pour écrire des larges documents)
- XHTML (fait spécialement pour afficher des pages Web)
- DITA (pour écrire des documents très modulaires)
- MathML (pour les formules mathématiques, surtout en combinaison avec XHTML)
- Open Office (Open Document Format for Office Applications) et le nouveau langage propriétaire de Microsoft

## 2.1 Caractéristiques de base d'un "document" SGML/XML

- 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

### Voici un tableau comparatif HTML/XML/SGML

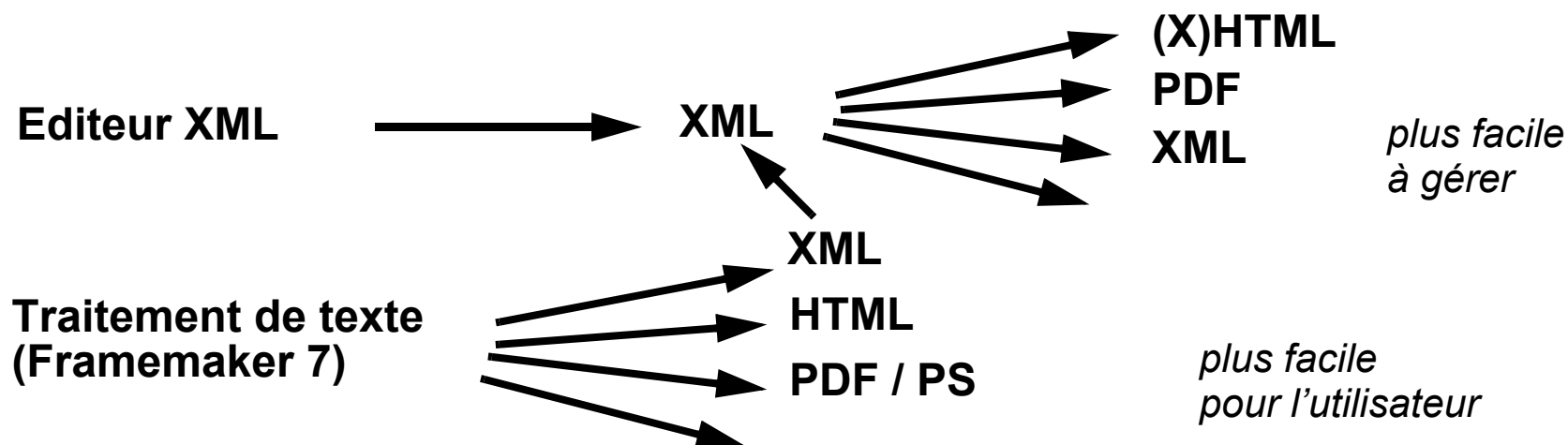
Fonction	monde HTML	monde XML	monde SGML
<b>Liens entre documents</b>	Balise <A>	Xlink (+ Xpointer & Xpath)	HyTime & TEI
<b>Assemblage de documents</b>	"calculs server-side"	XInclude (+ Xpointer & Xpath) ou entités ou "calculs server-side"	Entités SGML
<b>Style</b>	CSS2	XSL (CSS)	DSSL
	CSS1		
<b>Markup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

[url: Source \(http://wdvl.internet.com/Authoring/Languages/XML/Overview\)](http://wdvl.internet.com/Authoring/Languages/XML/Overview) (1998)

## 2.2 Edition de textes en XML

- Outils permettant d'éditer un "arbre" (quelques programmes Java gratuits)
- Outils d'édition de texte structuré (éditeurs XML ou de programmation comme Emacs)
- Outils semi-professionnels qui font les deux plus un pseudo WYSIWYG avec CSS: prix variables
- Traitements de texte professionnels XML comme FrameMaker 7.2: chers
- Plug-ins pour traitement de texte à la Word (médiocres en règle générale)
- Filtres vers XML (depuis Word, RTF, Open Office, Latex, etc.): médiocres par nature car non-structuré vers structuré.

### 2 grandes options:



## 2.3 Vocabulaires pour "publishing"

- Voici une petite typologie

### (1) Text markup général pour des gros documents

- 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

url: <http://www-106.ibm.com/developerworks/xml/library/x-dita3/index.html>

### (3) Par domaine

- Vocabulaires pour un domaine précis (souvent juste pour échanger des données)
  - Exemple: SwissDroitML ?

### (4) Pour l'échange de données:

- par exemple des nouvelles avec RSS, des services entre machines avec SOAP

### (5) Par médias / formalismes

- SVG, Web3D, MathML, ....

## 2.4 Rappel de la philosophie XML

### Concernant la structure des documents

1. Séparation du contenu de la forme
2. Balisage qui décrit le contenu
3. Standardisation qui facilite la réutilisation

### Concernant le style:

- Appliquer le style en fonction du médium (impression, écran, téléphone portable etc.)

### Concernant la gestion:

- Partage du travail entre "system architect" (base de données/de documentation), "information architect" (créateur du schéma) et auteur.
- Possibilité de faire de recherches plus puissantes (pas juste indiquer des mots clefs, mais pouvoir indiquer qqch. comme "je veux retrouver toutes les personnes ayant un email à l'université de Genève et qui ont une préférence pour les DVD de type polar").

### NOTE:

- Des solutions comme Open Document Format for Office Applications (Open Office) ou son équivalent Microsoft ne font pas partie de cette philosophie...



## **3. Processus de traitement**

Dans ce chapitre on introduira quelques concepts utiles pour la suite:

- La notion de "filtre"
- XML et les navigateurs Web
- Production de documents papier

## 3.1 Filtres

Un filtre est un programme qui traduit des données d'un format vers un autre.

### A. Filtres XML -> HTML

- Processeurs XSLT
  - On peut les utiliser en temps "réel" dans un client web (on associe une feuille de style à une page XML)
  - Mais dans le monde de la documentation on les utilise plus du côté serveur (pages dynamiques) et surtout en mode "batch" (on traduit les contenus avant de les poser en tant que HTML sur le serveur)
  - XSLT n'est pas abordé dans ce document !

[url: http://tecfa.unige.ch/guides/tie/html/xml-xslt/xml-xslt.html](http://tecfa.unige.ch/guides/tie/html/xml-xslt/xml-xslt.html) (Introduction à XSLT)

- Il existe aussi pleins de scripts fait avec des langages qui ont des parseurs XML (PHP, JSP, ASP, Perl, Java, etc.)

### B. Filtres Base de donnée -> XML -> XML

- Dans l'industrie, on utilise souvent des architectures à tiers
  - Exemple: les données sont dans un base de données SQL; quand on les sort on les met d'abord en XML proche de la base de données; dans un pas final traduction en XHTML (format XML)
- Il est facile de faire des visualisation de contenus XML en SVG ou X3D (format XML)

### C. Filtres XML-> XML -> PDF

- La traduction de XML vers le format XSL-FO est très populaire, car ce format permet de fabriquer du PDF, Postscript, RTF, etc.

## 3.2 Documents XML pour Internet

### A. Navigateurs

- Les navigateurs modernes ont un support CSS pour XML et XSLT
  - IE Explorer 6+: support XML + CSS1, CSS2 médiocre, XSLT bon
  - Mozilla 1.3x: XML + CSS1, CSS2 bon, XSLT (qq limites), MathML
  - Firefox 7.5: XML + CSS1, CSS2 bon, XSLT (qq limites), MathML, SVG limité
- Pour la plupart des "grands formats" (Docbook, Dita, etc.) il existent des feuilles de style
  - soit XML + CSS2, soit XML + XSLT

### B. Applets Java et autres outils de visualisation de contenus

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

### C. Server-Side: marche sans problème!

- Principe: On pose XML sur un serveur (fichiers ou base de données), mais on traduit le contenu en HTML avant de le "servir". C'est la solution préférée et marche le mieux.
- Traducteurs
  - Avec XSLT: XML->HTML ou XML->XSL/FO->PDF
  - solutions diverses avec des scripts: XML->HTML ou PDF
- Bases de données
  - BD SQL -> XML -> HTML
  - BD XML -> XML -> HTML

### 3.3 Note sur XSL-FO

- Pour une introduction à XSL-FO, voir:  
[url: http://tecfa.unige.ch/guides/tie/html/xml-xslfo/xml-xslfo.html](http://tecfa.unige.ch/guides/tie/html/xml-xslfo/xml-xslfo.html) (Transparents)
- XSL/FO est un standard populaire pour produire des documents imprimés de haute qualité
- Aucun navigateur majeur ne supporte XSL-FO, donc on l'utilise du côté serveur ou en batch.
- En règle générale, XSL-FO est utilisé avec XSLT dans la chaîne de production suivante:
  - On traduit d'abord XML avec XSLT vers XSL-FO
  - Ensuite XSL-FO vers PDF, PostScript ou un autre format d'impression

#### Outils:

- Processeur FOP de Apache (2002), marche assez bien, mais ne sait pas adapter la largeur des cellules dans une table...  
Une nouvelle version améliorée de Apache FOP est en test alpha (dec 2005)

[url: http://xmlgraphics.apache.org/fop/](http://xmlgraphics.apache.org/fop/)

- Outils commerciaux chers

#### Exemple d'un fragment FOP:

```
<fo:block font-size="12pt" line-height="12pt" text-align="center"
          space-before.optimum="12pt">Organisation:
TECFA, Université de Genève
</fo:block>
<fo:block font-size="12pt" line-height="12pt" text-align="center"
          space-before.optimum="12pt">
  Organisation: Service de formation continue, Université de Genève
</fo:block>
```

## 4. Langages auxiliaires

- On introduira (très brièvement) quelques langages utiles pour traiter/combiner/manipuler des documents XML

### 4.1 Xpointer

- De la spécification
  - Beaucoup de types d'application à traitement XML ont besoin d'adresser **les structures internes** des ressources XML en recourant à des adresses URI, par exemple, le langage de liaison XML [XLink], le langage d'inclusion XML [XInclude], le cadre de description des ressources [RDF] et le protocole SOAP 1.2 [SOAP12].
  - Le processeur XPointer reçoit en entrée une ressource XML et une chaîne à utiliser comme pointeur (par exemple, un identificateur de fragment, ..., puis essaye d'évaluer le pointeur par rapport à la ressource et produit en sortie une identification des sous-ressources
- A retenir en gros:
  - il s'agit d'une problématique assez compliquée
  - avec XInclude on utilise dans les cas simples des expressions XPath
  - Autrement dit: Xpoint = XPath + quelques fonctionnalités additionnelles.

```
xpointer(//EPISODE)
```

- Exemple HTML

```
<a href="test.xml#xpointer//name[1]" >
```

- Exemple d'utilisation avec XInclude (à lire sur une seule ligne):

```
<xi:include href="/db/shakespeare/plays/macbeth.xml#xpointer  
  (//SPEECH[SPEAKER&='witch' and near(LINE, 'fenny snake')])" />
```

## 4.2 XInclude

- XInclude est un standard pour assembler un document XML à partir de plusieurs autres.  
[url: http://www.w3.org/TR/xinclude/](http://www.w3.org/TR/xinclude/) (spécification)
- XInclude utilise XPointer
- Une expression XPointer simple = XPath, qui n'est abordé dans ce document, cf:  
[url: http://tecfa.unige.ch/guides/tie/html/xml-xpath/xml-xpath.html](http://tecfa.unige.ch/guides/tie/html/xml-xpath/xml-xpath.html)
- Les navigateurs ne supportent pas encore XInclude
- Server-side (page dynamiques) ou batch: plusieurs outils existent
- A Tecfa:
  - L'utilitaire xmllint (fait partie de la librairie libxml2 et marche pour la plupart des plateformes  
`xmllint --xinclude includes.xml > output.xml`  
(le fichier "includes.xml" contient le XInclude)  
[url: http://xmlsoft.org/](http://xmlsoft.org/) (downloads)
  - XInclude dans PHP 5 (interface expérimental)

### Exemple 4-1: Exemple assemblage d'un "livre"

- Création d'un gros XML à partir de fichiers XML entiers, puis XSLT+XSL-FO->PDF
- Un exemple en production pour produire un "livre" à partir de documents d'étudiants  
[url: http://tecfa.unige.ch/tecfa/teaching/staf18/files-j/](http://tecfa.unige.ch/tecfa/teaching/staf18/files-j/) (explications, styles xsl-fo et scripts)  
[url: http://tecfaseed.unige.ch/staf18/index.php?module=ePBLjolan&func=book](http://tecfaseed.unige.ch/staf18/index.php?module=ePBLjolan&func=book) (sources xml)  
[url: http://tecfa.unige.ch/tecfa/teaching/staf18/files-j/papiers-jolan.pdf](http://tecfa.unige.ch/tecfa/teaching/staf18/files-j/papiers-jolan.pdf) (résultat PDF)  
(cf. page suivante)

- Contenu du fichier includes.xml

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<book xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="intro-kiah.xml"/>
  <xi:include href="paper6.xml"/>
  <xi:include href="http://tecfaseed.unige.ch/staf18/modules/ePBL/uploads/
proj3/paper81.xml"/>
  ..... </book>
```

### Exemple 4-2: XInclude de fragments non XML

```
<?xml version='1.0'?>
<document xmlns:xi="http://www.w3.org/2001/XInclude">
  <p>This document has been accessed
  <xi:include href="count.txt" parse="text"/> times.</p>
</document>
```

### Exemple 4-3: XInclude de fragments XML

```
<price-quote xmlns:xi="http://www.w3.org/2001/XInclude">
  <prepared-for>Joe Smith</prepared-for>
  <good-through>20040930</good-through>
  <xi:include href="price-list.xml" xpointer="w002-description"/>
  <volume>40</volume>
  <xi:include href="price-list.xml" xpointer="element(w002-prices/2)"/>
</price-quote>
```

## 4.3 Note: XML de base possède un simple système d'inclusion

- Pas très puissant, mais marche partout (sauf les navigateurs)
- On associe simplement un fichier à une entité, ensuite on l'utilise

### Exemple

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet href="ePBLbook10.xsl" version="1.0" type="text/xsl"?>
<!DOCTYPE book SYSTEM "ePBLpaper11.dtd"
[
  <!ENTITY intro SYSTEM "intro-iris.xml">
  <!ENTITY file7 SYSTEM "paper7.xml">
  <!ENTITY file8 SYSTEM "paper8.xml">
  <!ENTITY file9 SYSTEM "paper9.xml">
  <!ENTITY file10 SYSTEM "paper10.xml">
  <!ENTITY file12 SYSTEM "paper12.xml">
  <!ENTITY file13 SYSTEM "paper13.xml">
  <!ENTITY file14 SYSTEM "paper14.xml">
]>
<book>
  &intro;
  &file7;
  &file8;
  &file9;
  &file10;
  &file12;
  &file13;
  &file14;
</book>
```



## 4.4 MathML

[url: http://www.w3.org/Math/XSL/](http://www.w3.org/Math/XSL/)

[url: http://www.w3.org/TR/MathML/](http://www.w3.org/TR/MathML/)

- permet de représenter des formules mathématiques
- implémenté dans Mozilla (depuis 0.9) et Amaya
- ne marche pas dans IE (mais il existe des plugins: MathPlayer et Techexplorer)

### Exemple:

[url: http://tecfa.unige.ch/guides/xml/examples/mathml/](http://tecfa.unige.ch/guides/xml/examples/mathml/)

[url: http://tecfa.unige.ch/guides/xml/examples/mathml/xhtmll-with-mathml-test.xhtml](http://tecfa.unige.ch/guides/xml/examples/mathml/xhtmll-with-mathml-test.xhtml)

- Code

```
<b>Corollary 2</b> [Contractive Sequence Theorem] <em>If
<math xmlns='http://www.w3.org/1998/Math/MathML'><mo>( </mo><msub><mi>x</mi> <mi>n</mi></msub><mo></mo></math> is a sequence, for which there is a number <math
xmlns='http://www.w3.org/1998/Math/MathML'><mi>C</mi><mi>&lt;</mi><mn>1</mn></math>
such
that <math xmlns='http://www.w3.org/1998/Math/MathML'><mo>| </mo><msub><mi>x</mi>
<mrow><mi>n</mi><mo>+</mo><mn>2</mn></mrow></msub><mo>-</mo><msub><mi>x</mi>
<mrow><mi>n</mi><mo>+</mo><mn>1</mn></mrow></msub><mo>| </mo><mo>&le;</mo><mi>C</mi>
<mo>&sdot;</mo><mo>| </mo><msub><mi>x</mi> <mrow><mi>n</mi><mo>+</mo><mn>1</mn></mrow></msub><mo>-</mo><msub><mi>x</mi> <mi>n</mi></msub><mo>| </mo></math>, then <math
xmlns='http://www.w3.org/1998/Math/MathML'><mo>( </mo><msub><mi>x</mi> <mi>n</mi></msub><mo></math>
```

- Pour le résultat, c.f. le fichier :)

## 4.5 XLink

- XLink est un standard qui permettrait de créer des hypertextes puissants
- En réalité seulement le type *simple* est implémenté (dans SVG, X3D, Mozilla/Firefox, etc.)
- XHTML 2.0 utilise qqch. de plus simple, mais qui s'inspire des simple links

### Exemple d'un simple xlink:

```
<RECIT xmlns:xlink="http://www.w3.org/1999/xlink">
.....
<paragraphe xlink:href="http://jigsaw.w3.org/css-validator/check/referer"
  xlink:type="simple">
  CSS Validator</paragraphe>
```

Voici la promesse (*R = required, O = optional*):

Type d'attribut	Type de lien (exemple: <monlien xlink:type="locator" ... />					
	simple	extended	locator	arc	ressource	title
<b>type</b>	R	R	R	R	R	R
<b>href</b>	R		R			
<b>role</b>	O	O	O		O	
<b>arcrole</b>	O			O		
<b>title</b>	O	O	O	O	O	
<b>show (comment?)</b>	O			O		
<b>actuate (quand?)</b>	O			O		
<b>label</b>			O		O	
<b>from</b>				O		
<b>to</b>				O		

## 5. Notes sur XHTML

### XHTML ressemble d'abord beaucoup à HTML

- La différence superficielle: le langage adopte les normes XML
  - "case sensitivity", "valide" et "bien formé"
- Notamment:
  - toutes les balises en minuscules,
  - déclaration d'une DocTypeDef (la déclaration XML reste une option)
  - déclaration du name space dans la balise html
  - pas de croisements de balises
  - Les éléments vides doivent avoir un "terminateur": <hr> devient <hr /> etc.
  - Toujours utiliser `&amp;` pour `&` dans des attributs, y compris les URLs
  - Chaque attribut doit avoir une valeur: `<hr noshade="noshade">`.

### Voici un exemple

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title>document test </title> </head>
  <body>
    <h1>document test</h1>
    Voici un test!
  </body>
</html>
```

## Pourquoi XHTML ?

- C'est l'avenir (donc ces pages vivront plus longtemps)
- Intégration d'autres langages comme MathML et SVG !
- Intégration au "semantic web" (langages RDF)
- Utilisation de XSLT (par exemple pour fabriquer des tables des matières)
- Extensibilité et modularité
- Meilleure interopérabilité entre clients (portables, ordinateurs, etc.)

## Versions de XHTML

### ***XHTML-1.0-Strict (Jan 2000)***

- Seule version de XHTML vraiment dans l'esprit "XML"
- Pas de "texte mixte", ce qui le nettement plus facile à digérer pour un style-sheet.
- Séparation de contenu et style (CSS)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

### ***XHTML-1.0-Transitional***

- Texte mixte
- Balises "font"

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

### ***XHTML-1.0-Frameset***

- Frames

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

## **XHTML-1.1 (Juin 2001)**

- Version nettoyé modularisée de XHTML-1.0 strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

## **XHTML-2.0 (working draft 2005)**

- Plus de structuration, par exemple il y un élément "section" qui entoure une section
- Plus de souplesse
- Meilleure intégration avec le monde XML
- Meilleur "event handling"
- Plus de flexibilité pour l'inclusion d'images:

```
<p src="w3c-logo" type="image/png, image/jpeg;q=0.2">W3C logo</p>
```

- etc. (la liste est longue)

## **Utilisation non orthodoxe de XHTML**

- Vous pouvez écrire un document large avec XHTML et ensuite appliquer un filtre qui produit un hypertexte (ensemble de pages reliées)
- Exemple: une simple feuille de style pour XHTML 1 strict (!)
- Voir aussi le script HTMLDoc pour HTML: <http://www.htmldoc.org/>  
*url: <http://tecfa.unige.ch/guides/xml/examples/xsl-xhtml/> (divide-doc.xsl)*
- Vous pouvez appliquer une feuille XSLT -> XSL-FO pour produire un document qui s'imprime avec numérotation de pages et un bon "look".

## 6. Docbook

### DocBook est une DTD pour écrire des documents techniques

- mais utile pour toutes sortes de documents (puisque un manuel technique est juste plus compliqué qu'un livre "normal")
- il existe des centaines de balises
- souvent, les organisations configurent un sous-ensemble
- il existe de nombreux outils (mais qui nécessitent souvent un support technique, donc une séparation des rôles d'auteur et de "producteur de documents finaux")

### Edition

La plupart des éditeurs XML sont préconfigurés pour éditer DocBook. Il existe trois types d'éditeurs:

- Des éditeurs XML comme Oxygen (pas facile pour les jeunes)
- Des éditeurs XML comme XMLSpy (plus ou moins wysiwyg), font l'affichage "authoring" avec CSS
  - alternative gratuite, mais moins bonne: Editeur Morphon
- Avec un traitement de texte comme Framemaker ou ArborText (difficiles à configurer)

### Exemple 6-1: Texte écrit avec Framemaker et SDocBook (un sous-ensemble)

[url: http://tecfa.unige.ch/guides/xml/examples/framemaker-sdocbook/](http://tecfa.unige.ch/guides/xml/examples/framemaker-sdocbook/)

(répertoire avec fichier FM et résultat PDF)

## Dans l'éducation: peu

- A Tecfa, 2 étudiants ont écrit leur mémoire avec SDocBook/Framemaker
  - Ex. O. Clavel (2002) Un environnement de développement pour TECFA

[url: http://tecfa.unige.ch/staf/staf-e/clavel/memoire/memoire.pdf](http://tecfa.unige.ch/staf/staf-e/clavel/memoire/memoire.pdf)

- Exemple d'un projet qui vise à développer des livres pédagogiques "open content": Edukalibre

[url: http://edukalibre.org/](http://edukalibre.org/)

Citation: "Edukalibre est un projet financé par la Commission Européenne sous le programme Socrates/Minerva. Ce projet est destiné à la promotion des technologies de communication et d'information dans le domaine de l'éducation. L'objectif principal de ce projet est d'explorer des nouvelles manières de produire des matériaux éducatifs, basé sur les pratiques et procédures observées dans la communauté du Logiciel Libre."

## Ressources:

- Site O'Reilly (qui publie des livres sur DocBook et produit des livres avec DocBook):

[url: http://www.docbook.org/](http://www.docbook.org/)

- Travail de séminaire (2005): Silvia Cadillo, La Standardisation de Documents

[url: http://tecfaseed.unige.ch/staf18/modules/ePBL/uploads/proj4/paper3.xml](http://tecfaseed.unige.ch/staf18/modules/ePBL/uploads/proj4/paper3.xml)

## 7. DITA

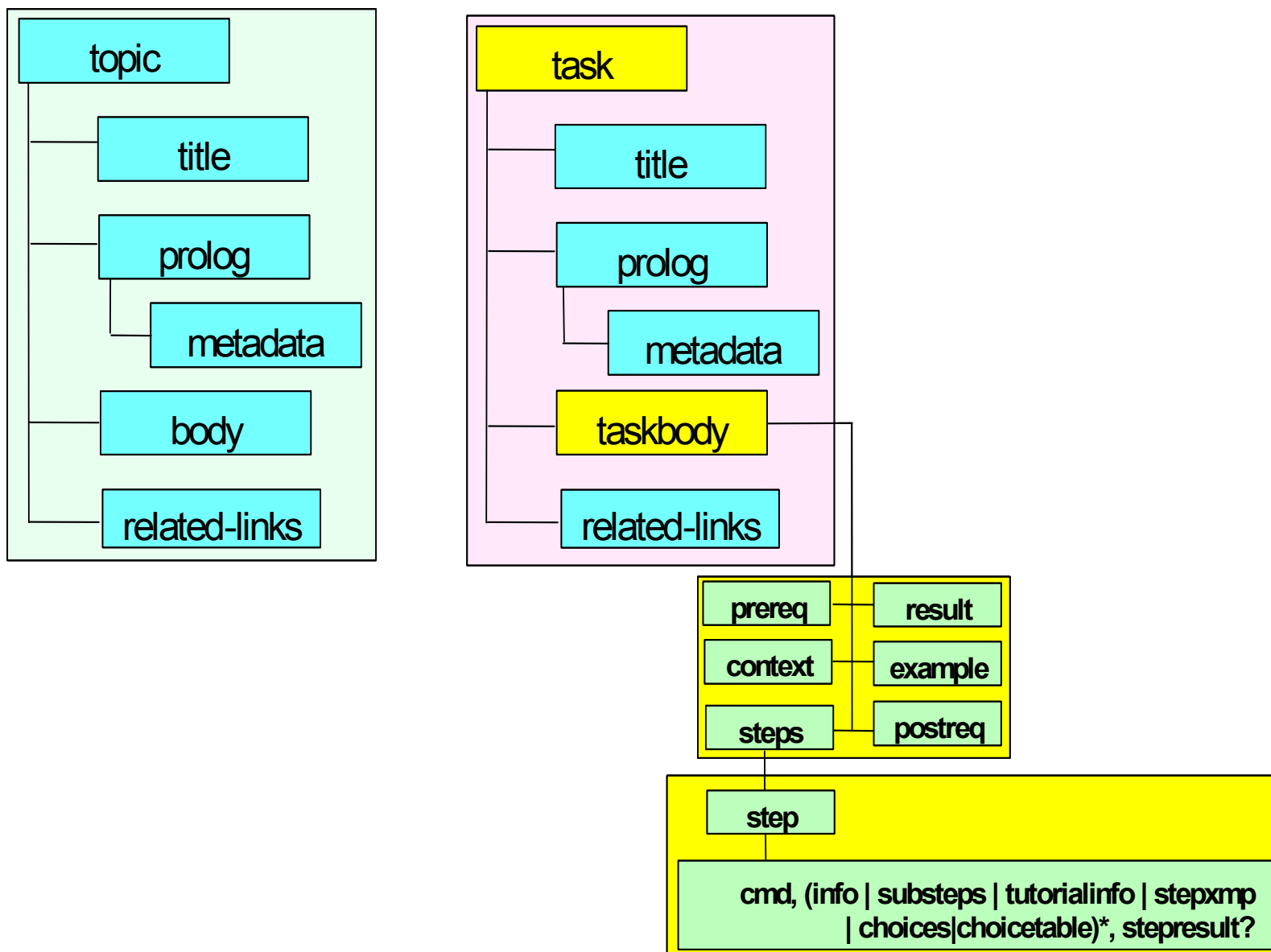
- DITA était proposé comme "nouvelle façon" de penser la documentation en 2000 par une équipe de IBM
- DITA est une architecture ouverte permettant de décrire des sujets (topics) extensibles
- DITA fournit 3 types de topics: topic générique, concepts, tâche, et référence
  - mais on peut en ajouter, soit des nouveaux types, soit des types qui héritent des propriétés
- Depuis 2005, DITA est un standard OASIS et "on en parle"
  - Exemple: Adobe a fait la documentation pour Creative Suite 2 (InDesign, InCopy, GoLive, Photoshop, Illustrator), avec DITA et Framemaker 7.1
  - Note: Certains (dont moi) ont accusé Adobe de laisser pourrir ce produit. Depuis que Adobe utilise Framemaker en interne, ils ont compris qu'il fallait améliorer des choses: résultat = Framemaker 7.2
- Définition du mot "DITA":
  - Darwin: DITA utilizes principles of inheritance for specialization
  - Information Typing: DITA was originally designed for technical information based on an information architecture of Concept, Task and Reference
  - Architecture: DITA is a model for extension both of design and of processes

### Architecture générale d'un topic:

- titre
- prologue (auteur, métadonnées, petite description, etc.)
- corps (sections structurés selon le type de topic)
- topics imbriqués et/ou liens



## Exemple de "topic générique" et de task



source: Don Day, <http://xml.gov/documents/completed/ibm/dita.ppt>

## Modularité et extensibilité

- Un nouveau type topic peut être autonome ou rattaché à un topic parent dont il peut "hériter" les propriétés et inaugurer une hiérarchie
- L'objectif de DITA est de constituer des documents assemblables par configuration de publication.

### Exemple 7-1: DITA et le Catalogue TECFA SEED

- Ce catalogue utilise plusieurs topics faits maison
  - card (définition d'une fiche comme base pour les autres)
  - c3msbrick (module pour portail communautaire)
  - c3mssoft (software pour ces modules)
  - learnactivity (activité d'apprentissage)
  - learnact (élément d'une activité d'apprentissage)
- Edition avec Framemaker (styles faits par DKS)
- Résultat

[url: http://tecfa.unige.ch/proj/seed/catalog/](http://tecfa.unige.ch/proj/seed/catalog/) (documentation variée)

[url: http://tecfa.unige.ch/proj/seed/catalog/net/catalog-eng.pdf](http://tecfa.unige.ch/proj/seed/catalog/net/catalog-eng.pdf)

[url: http://tecfa.unige.ch/proj/seed/catalog/net/xml/catalog-eng.xml](http://tecfa.unige.ch/proj/seed/catalog/net/xml/catalog-eng.xml)

- la version XML est un gros fichier et nécessite IE 6 ou Firefox 1.5
- aucun effort a été fait pour traiter ce fichier (le découper par exemple ou de remplacer la table des matières)
- On utilise le XSLT fourni pour le topic par défaut, donc nos topics héritent ...)
- Plans pour le futur:
  - Mettre ces topics dans une base de données XML et les servir dans des combinaisons "on demand".

## 8. Grammaires sur mesure

### Quand ?

- Lorsque il est important que le contenu soit complet et validé
- Pour aider à structurer une rédaction
- Lorsque vous voulez traiter l'information par un script (par exemple combiner des éléments de plusieurs textes, faire des statistiques, etc.)

### Exemple 8-1: DTDs "ePBL" pour l'enseignement par projet

- une DTD pour définir un projet, une DTD pour écrire le papier
- le tout est intégré dans un système (gestion des versions, extraction de données etc.)
- Différentes versions des DTD selon la nature du cours:
- Exemple "DTD peu structuré" (portail d'enseignement STAF-18):  
[url: http://tecfaseed.unige.ch/staf18/](http://tecfaseed.unige.ch/staf18/)
  - cliquer sur "projX" ou "papierX" dans les fenêtres "mini cockpit"
- Exemple "DTD plus structuré" pour un cours un psychologie expérimentale  
[url: http://tecfaseed.unige.ch/staf18/modules/ePBLjolan/xmlfiles11/](http://tecfaseed.unige.ch/staf18/modules/ePBLjolan/xmlfiles11/) (DTDs et feuilles de style)

### Utilisation de la DTD "Itsy Bitsy Teeny Weeny Simple Hypertext" IBTWSH 6.0

- [url: http://home.ccil.org/~cowan/XML/](http://home.ccil.org/~cowan/XML/)
- Définit un mini-XHTML que l'on peut imbriquer dans sa propre DTD.
- Utilisé à l'intérieur des balises "sémantiques"
- On utilise comme "modèle importé" le struct.model, c.à.d une sorte de mini XHTML strict.

