

DocBook, la quatrième dimension de la documentation technique

Copyright © 2002 par Camille Bégnis (mailto:camille@mandrakesoft.com)

Ce document peut être librement copié, distribué et/ou modifié selon les termes de la GNU Free Documentation License, Version 1.1 ou ultérieure publiée par la Free Software Foundation (<http://www.fsf.org/licenses/fdl.html>).

Cette présentation est une introduction à la DTD XML DocBook : la philosophie sous-jacente, ses avantages et comment en tirer parti. Nous supposons que le public possède une connaissance de base de la norme XML.

Nous introduirons DocBook par un historique rapide, et aborderons comment il est géré et évolue. Seront ensuite exposés les principes régissant DocBook du point de vue du codage de l'information, agrémentés de quelques exemples. Nous entrerons alors un peu plus en profondeur dans la DTD, comment elle est construite et ce que cela permet en terme de réutilisation. Pour compléter cet exposé, il sera nécessaire de présenter les outils de traitement permettant de transformer les fichiers source DocBook en un autre format afin de, finalement, diffuser l'information. Nous finirons la présentation par une démonstration s'appuyant sur les sources de ce même exposé.

1. D'où vient DocBook ?

DocBook a une histoire de plus de 10 ans :

- 1991 : naissance d'un projet commun entre HaL Computer Systems et O'Reilly, pour faciliter l'échange de documentation Unix. La DTD DocBook 1.0 est publiée le 10 novembre 1992.
- 1994 : le projet intéresse nombre de joueurs, dont Novell, Sun, Digital. Le groupe « Davenport » est institué pour prendre en charge sa maintenance. De plus en plus de projets de développement informatique adoptent ce standard pour leur documentation.
- 1999 : le groupe « Davenport » devient un comité technique du consortium mondial OASIS, à but non lucratif, qui œuvre pour la diffusion et l'adoption de standards professionnels ouverts.
- Le comité se réunit régulièrement pour discuter des propositions d'évolution faites par les milliers d'utilisateurs DocBook de par le monde (33 langues proposées par les feuilles de style), du particulier à la grosse maison d'édition.

Parmi les projets les plus remarquables de DocBook, on notera ceux de The «Linux Documentation Project» (TLDP), et la plupart des gros projets de développement libres, ainsi que PHP, Sun, IBM Linux Technology Center, O'Reilly, etc.

Il convient ici de citer le nom de monsieur Norman Walsh qui, malgré sa modestie légendaire, est l'instigateur de DocBook et son principal développeur depuis les origines du projet.

2. XML et DocBook

À l'origine, DocBook était disponible uniquement en SGML. Avec la propagation de la recommandation XML, une version XML est apparue, et depuis lors les deux versions sont maintenues en parallèle. Les concepts XML que DocBook s'est appropriés sont :

- Source unique (single sourcing) ;
- Principe de séparation entre le contenu et l'apparence ;
- Métadonnée.

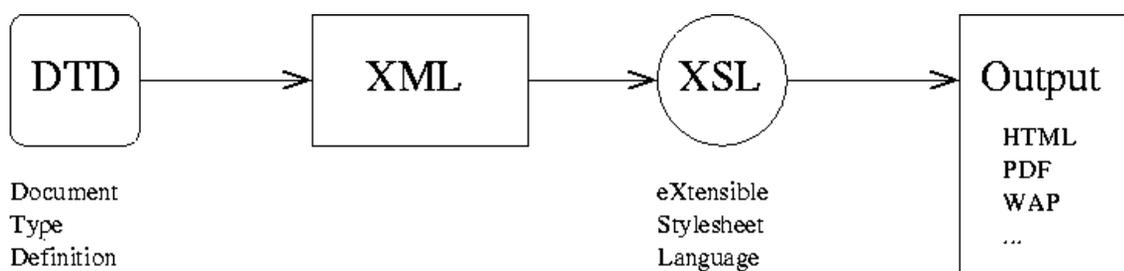
XML et toutes les technologies qui y sont associées sont des recommandations du W3C (World Wide Web Consortium).

3. Avantages

Qu'est-ce que tout cela apporte dans une optique de documentation technique :

- Gain de temps : la rédaction se fait en une seule fois pour une multitude de supports ;
- Pas de confusion entre deux professions très différentes : rédacteur technique et graphiste/éditeur ;
- Les métadonnées donnent une dimension supplémentaire à l'information, d'une valeur inestimable pour l'indexation, les réutilisations futures, l'échange de données, etc.

4. De la grammaire au document final



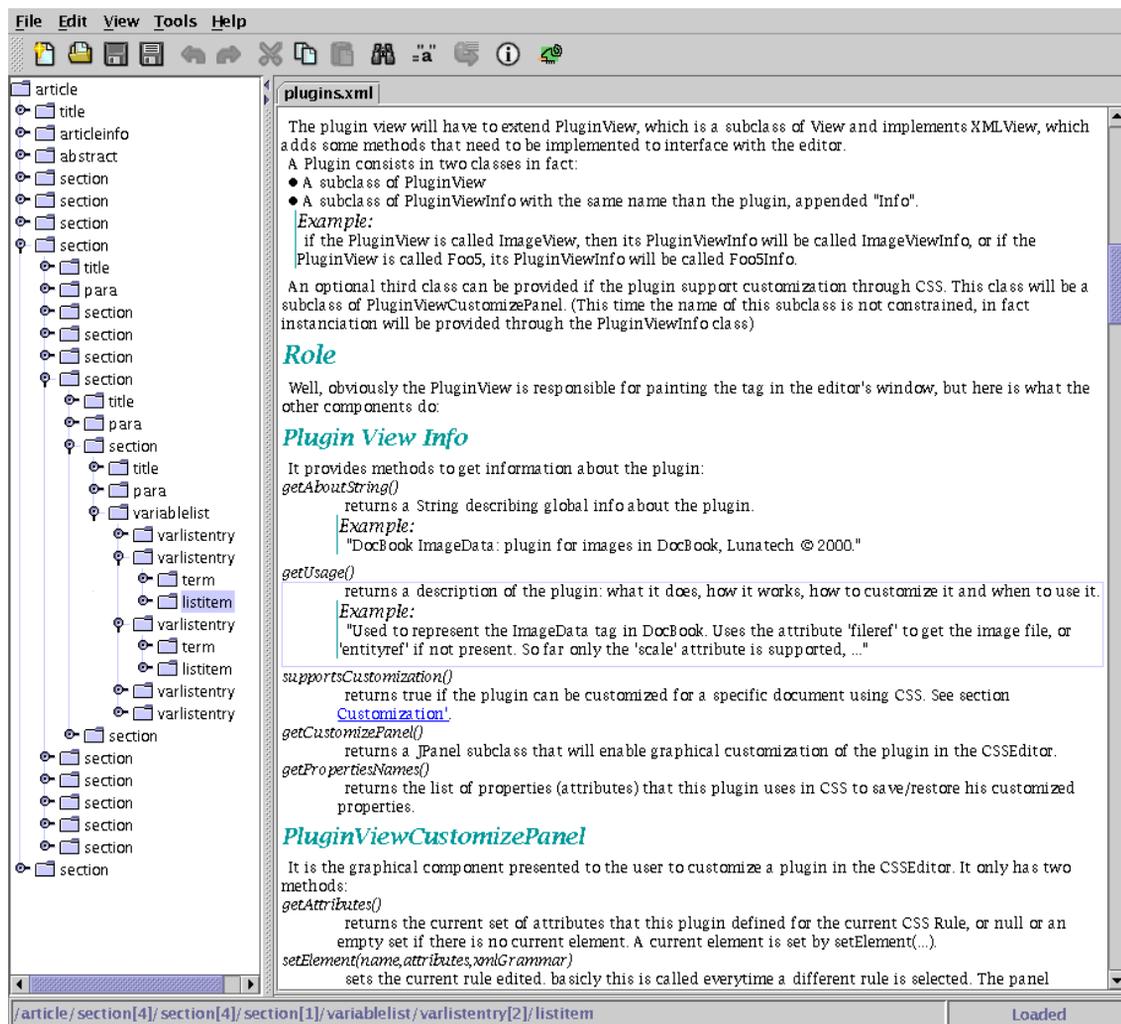
- La DTD définit un langage conforme à la grammaire XML ;

- Le fichier XML contient l'information effectivement codée ;
- La feuille de style XSL définit la façon dont les données seront présentées dans le document final.

5. Écrire XML

Il existe deux façons de base pour rédiger des documents en langage XML.

À la main : en écrivant directement les balises qui marquent le texte, de préférence à l'aide d'une assistance (emacs + PSGML).	<pre><sect1 id="writing"><title>Écrire XML</title> <para>Deux façons d'écrire en utilisant un langage XML&nbsp;</para> <itemizedlist> <listitem> <para>à la main&nbsp;<: en écrivant directement les balises qui marquent le texte&nbsp;</para> </listitem> <listitem> <para>avec un éditeur XML &laquo;&nbsp;<WYSIWYG&nbsp;<raquo;</para> </listitem> </itemizedlist> </sect1></pre>
Avec un éditeur XML visuel	Exemple : Morphon (morphon.png)



6. Les pièges du WYSIWYG

- Le principe du WYSIWYG en général n'est qu'un mirage.

Il est bien rare, en effet, que le rendu à l'écran soit strictement identique à l'impression, que ce soit un problème de fontes, de marges, etc.

- Le WYSIWYG est une aberration dans un contexte de réutilisation de contenu.

Car on souhaite justement de pouvoir présenter les données, au final, sous une multitude de présentations différentes...

- La plupart des spécialistes s'accordent sur le fait qu'utiliser un éditeur visuel (en cachant les balises) est une mauvaise idée car cela balaye la plupart des avantages du XML (sémantique contre présentation, métadonnées, etc.).

De fait, l'expérience prouve qu'après un certain temps d'adaptation à des éditeurs XML textuels, les rédacteurs sont plus rapides et plus efficaces.

7. La DTD DocBook

- Conçue tout spécialement pour écrire de la documentation technique, et plus spécifiquement informatique ;
- Très polyvalente et robuste.

DocBook propose un vocabulaire pour les documents structurés :

- différencie clairement les aspects sémantiques et de présentation ;
- les documents en texte pur ou MS Word ne sont pas structurés ;
- les documents HTML ou MS Word (avec modèle) sont peu structurés ;
- les documents DocBook sont strictement structurés.

Par structuration, nous insistons ici sur le fait que chaque élément texte est très clairement identifié : premièrement, par le nom de la balise parente et ses attributs éventuels, et deuxièmement, par sa position dans l'arborescence XML qui le positionne très distinctement par rapport à ses ancêtres, sa descendance, ses semblables, etc.

8. La structure de la DTD DocBook

Les nombreux (>370) éléments qui composent la DTD DocBook sont séparés en deux catégories :

- Hiérarchie : éléments structurels
- Information : éléments qui contiennent les données elles-mêmes (le texte)

Chacun de ces ensembles peut être réutilisé pour construire une DTD dérivée de DocBook.

Au sens strict, seul mérite l'appellation de DocBook la DTD officielle telle que publiée par OASIS. En pratique, on peut fort bien développer soi-même une DTD compatible DocBook (dont les documents dérivés sont strictement valides du point de vue de la DTD officielle). D'autre part, la dualité exposée ci-dessus permet de développer une DTD en gardant une des deux catégories d'éléments et en créant l'autre ex nihilo.

9. Implications

Pour les rédacteurs :

- l'attention se porte strictement sur le contenu, et non pas sur la présentation ;
- écrire des informations réutilisables nécessite un état d'esprit et une expérience particulière.

Pour les graphistes et/ou programmeurs :

- l'attention se porte uniquement sur la présentation d'un format de sortie particulier.

Pour toute l'équipe :

- le travail coopératif est amélioré de manière significative et facilité grâce à l'utilisation de standards ouverts et d'outils génériques de travail coopératif tels que le CVS.

10. Les feuilles de style DocBook XSL

Nous avons jusqu'ici parlé du contenu, voici venir la présentation de ce contenu.

Développées en parallèle avec la DTD, ces feuilles de style permettent la transformation du document source XML en :

- (X)HTML : pour une publication en ligne en une seule page HTML, ou plusieurs pages dont le découpage logique est au choix de l'utilisateur;
- Aide HTML : format des pages d'aide du navigateur d'aide de Microsoft Windows ;
- FO : ce langage XML définit très précisément l'apparence physique du texte et des images contenues dans un document, ainsi que sa disposition dans une feuille de format donné. Il existe ensuite des outils permettant de transformer ces documents « Formatting Objects » en formats plus courants tels que le PDF.

Ces feuilles de style comportent un grand nombre de paramètres (>200) permettant de contrôler finement et de manière simple le rendu final.

11. Les feuilles de style DocBook DSSSL

Le format DSSSL de feuilles de style était utilisé pour les documents SGML. Comme le XML n'est qu'un SGML simplifié, il est possible d'utiliser les outils SGML sur du XML. Les feuilles de style disponibles sont :

- HTML ;
- jadetex : ce format TeX pourra ensuite être transformé en DVI puis PostScript ou PDF.

Au cas où l'on souhaiterait modifier l'apparence du document final mais qu'aucun paramètre standard ne convient, il reste toujours la possibilité de modifier soi-même la feuille de style comme nous le verrons plus loin. Il s'avère qu'il est beaucoup plus aisé de maîtriser le langage XSL que le DSSSL.

12. Les technologies DocBook sont libres

Les DTD et feuilles de style sont publiées sous une licence libre :

- elles sont utilisables et distribuables gratuitement ;
- le code source est disponible gratuitement et il est autorisé (recommandé) de le modifier.

En conséquence :

- il existe une grande communauté d'experts et une documentation pléthorique susceptible d'aider les nouveaux venus ;
- aucun risque de voir la technologie disparaître avec la société qui le développe ;
- facilité pour personnaliser/réutiliser le produit pour qu'il corresponde à des critères très stricts.

13. Outils de transformation

Outils SGML

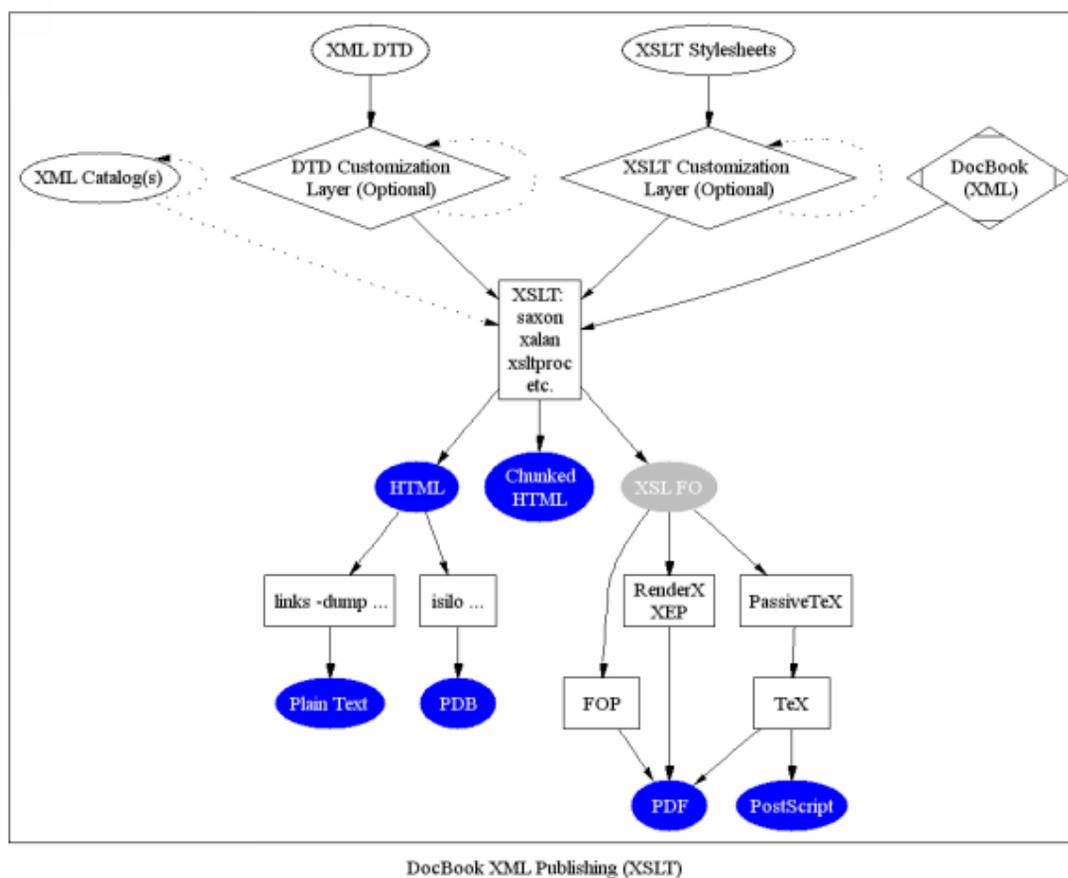
- (open)jade : grâce aux feuilles de style DSSSL, permet la transformation du SGML (mais aussi du XML) en TeX, HTML, RTF, etc. ;
- (pdf)jadetex : permet la transformation du TeX en DVI ou PDF.

Outils XML

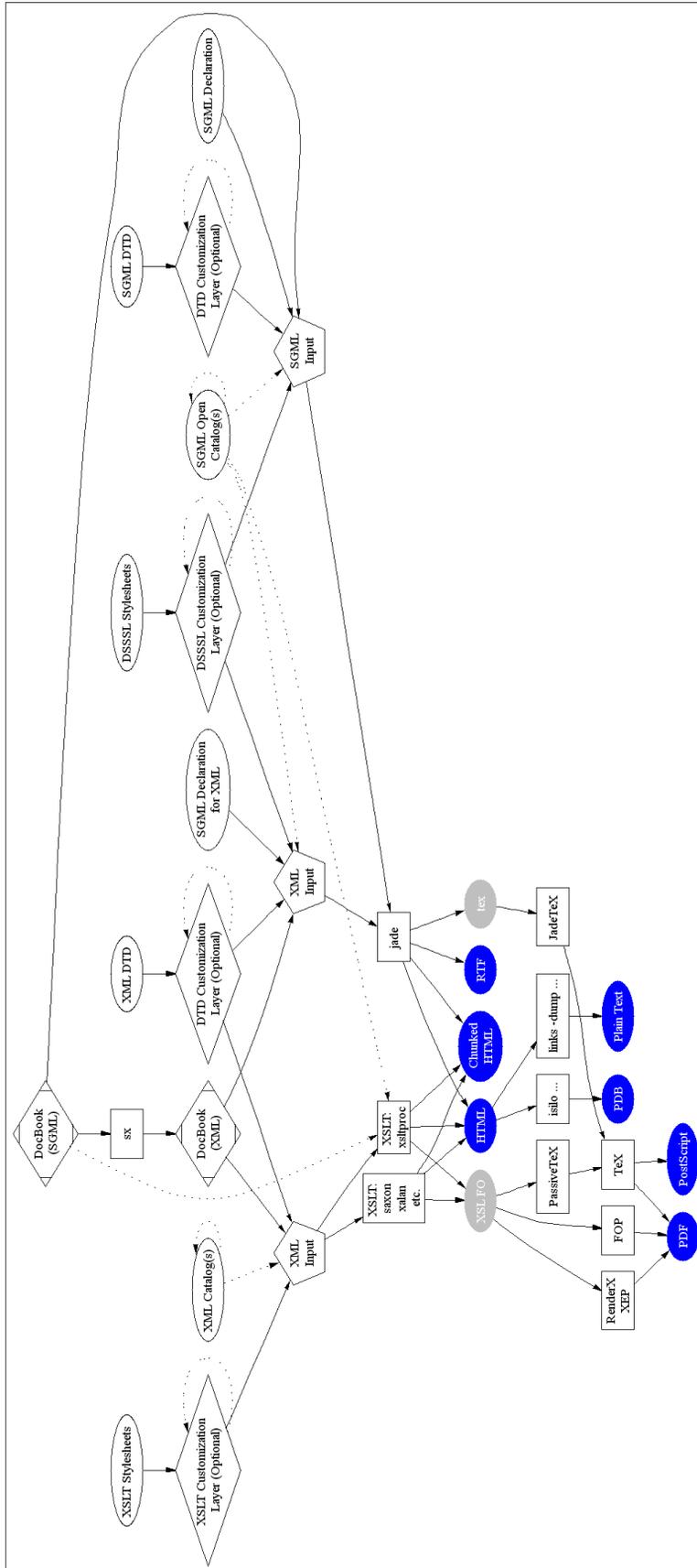
DocBook, la quatrième dimension de la documentation technique

- XSLT : saxon, xalan (classes JAVA), xsltproc, permettant la transformation du XML en FO, HTML, XML ;
- processeurs FO : FOP, passivetex, XEP : transforment les FO en PDF (en cours de développement).

Voici les chaînes de publication du XML (xmlpubxslt.png)



et du SGML + XML (xmlsgmlpub.png) (merci Norman).



Full DocBook SGML/XML Publishing Model

14. Exemples

- Cette présentation a été écrite en DocBook XML (bien entendu !) à partir d'une toute petite DTD (`presentation.dtd`) compatible DocBook écrite pour l'occasion et une feuille de style HTML (`pres2html.xsl`).
- MandrakeSoft utilise DocBook avec beaucoup de succès depuis la fin du siècle dernier :
 - robustesse : permet des modifications majeures (telles que modifier la taille du papier) au dernier moment ;
 - génération d'un livre de 300 pages (avec index et table des matières automatique, etc.) en moins de 15 minutes ;
 - aucun problème dans la chaîne de production avec des documents lourds (100 Mo).

DocBook est utilisé chez MandrakeSoft pour les manuels de l'utilisateur et de référence, pour la documentation en ligne de certains logiciels, pour les manuels de formation, etc. Si les contenus sont difficilement échangeables d'un type de document à l'autre (manuel de référence et de formation), l'information est contenue dans des modules qui peuvent être très facilement déplacés d'un manuel à un autre. Cela permet de construire très rapidement un nouveau manuel sur mesure à partir de briques élémentaires déjà prêtes.

15. DocBook n'est pas la seule solution

- D'autres DTD sont plus adaptées à des contextes différents de la documentation technique.
- La DTD DITA (Darwin Information Typing Architecture) d'IBM est aussi spécialisée pour la documentation technique, qui repose sur le principe de séquences d'information indépendantes appelées sujets (« topics »).

Néanmoins, au vu de sa robustesse et de sa grande souplesse, des auteurs sont amenés à choisir DocBook du fait de sa simplicité relative. Il est possible d'utiliser DocBook pour la rédaction de textes littéraires, de sites Web, de présentations... DocBook fait même le café :-)

Ce n'est qu'un demi canular : il suffit d'utiliser l'élément `<procedure>`, pour décrire toutes les étapes, de la fabrication du café. On peut imaginer d'en imprimer la recette pour un livre de recettes, voire de transmettre ces informations à une cafetière futuriste capable de lire le XML...

16. XML n'est pas la seule solution

- Il s'agit généralement d'un saut technologique important par rapport à un système de documentation « traditionnel » : il est indispensable de bien quantifier l'investissement initial (notamment en temps de formation) et les économies espérées à long terme.
- Lors du démarrage d'un nouveau projet de documentation, il faut savoir reconnaître les avantages immédiats et futurs du XML, envisager les utilisations à long terme. L'aide d'un spécialiste sera précieuse.
- XML est une technologie qui évolue rapidement, ainsi que les outils qui en tirent parti. La chaîne de traitement dans son ensemble n'est pas encore arrivée à maturité.

En effet, si les normes sont abouties, les outils qui les exploiteront arrivent avec un temps de retard car les éditeurs de logiciels sont réticents à développer un outil sur une recommandation susceptible d'évoluer, et qui n'aboutira pas forcément sur un standard.

17. That's all folks!

Ce qu'il faudra retenir de cette présentation :

- Unicité de la source ;
- Séparation contenu - apparence ;
- Il existe des DTDs pour toutes sortes de besoin, mais pour ce qui est de la documentation technique, DocBook est LA référence ;
- DocBook est très utile pour gérer des documents devant être publiés dans des formats différents ;
- DocBook est extrêmement efficace pour de gros projets de documentation, grâce à l'écriture de DTDs et de feuilles de style spécialisées, mais aussi pour de petits projets à travers les feuilles de style fournies en standard
- la grande majorité des outils présentés ici sont libres et gratuits.

18. Références

- OASIS (Organization for the Advancement of Structured Information Standards) (<http://www.oasis-open.org/docbook/>) : la page officielle de DocBook, où l'on trouvera les DTDs DocBook officielles.
- L'espace de développement ouvert de DocBook (<http://sourceforge.net/projects/docbook/>) : chez SourceForge, où l'on trouvera les dernières versions des feuilles de

style DocBook, ainsi que de nombreux outils pour soumettre des erreurs ou suggérer des modifications.

- DocBook: The Definitive Guide (<http://docbook.org/tdg/en/html/docbook.html>): le titre en dit long, en anglais seulement, et par l'auteur de DocBook lui-même. Une traduction française est disponible chez O'Reilly.
- DocBook Wiki (<http://www.docbook.org/wiki/moin.cgi/>): un site Web collaboratif en anglais comportant une somme important d'informations sur DocBook avec notamment l'accès à des liens vers de la documentation, des tutoriels, des présentations, etc.
- DocBook Demystification HOWTO (<http://tldp.org/HOWTO/DocBook-Demystification-HOWTO/>): si cette présentation ne vous a pas encore convaincu, lisez ce document de Eric Raymond.
- Formats de fichiers : attention danger ! (<http://reseaux74.cri74.org/Lettres/N12/index.html>): un article très intéressant sur le choix d'un format de document texte.

A. La DTD présentation (presentation.dtd)

```
<!-- Main DTD defining the structure of a presentation -->

<!-- ..... -->
<!-- ISO character entity sets ..... -->

<!ENTITY euro "&#x20AC;"><!-- euro sign, U+20AC NEW -->
<!ENTITY % dbcent PUBLIC
"-//OASIS//ENTITIES DocBook XML Character Entities V4.1.2//EN"
"/usr/share/sgml/docbook/xml-dtd-4.1.2/dbcentx.mod">
%dbcent;

<!ENTITY % content "mediaobject|itemizedlist|orderedlist|programlisting|informaltable|ulink">
<!ENTITY % section "title,(para|%content;)*">

<!-- The root element -->
<!ELEMENT article (title,subtitle?,articleinfo,sect1+,appendix*)>
  <!ATTLIST article lang CDATA #IMPLIED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT subtitle (#PCDATA)>

  <!ELEMENT articleinfo (copyright,legalnotice?,abstract)>

  <!ELEMENT copyright (year,holder)>
    <!ELEMENT year (#PCDATA)>
    <!ELEMENT holder (#PCDATA|ulink)*>
  <!ELEMENT abstract (para+)>
  <!ELEMENT legalnotice (para+)>

  <!ELEMENT sect1 (%section;)>
    <!ATTLIST sect1 id ID #REQUIRED>
```

```
<!ATTLIST sect1 role CDATA #IMPLIED>

<!ELEMENT para (#PCDATA|%content;)*>
  <!ATTLIST para condition CDATA #IMPLIED>

<!ELEMENT mediaobject (imageobject+)>
  <!ATTLIST mediaobject condition CDATA #IMPLIED>
<!ELEMENT imageobject (imagedata)>
  <!ELEMENT imagedata EMPTY>
  <!ATTLIST imagedata fileref CDATA #REQUIRED>
  <!ATTLIST imagedata width CDATA #IMPLIED>
  <!ATTLIST imagedata format CDATA #IMPLIED>

  <!ELEMENT orderedlist (listitem+)>
<!ELEMENT itemizedlist (listitem+)>

  <!ELEMENT listitem ((para|%content;)*)>

<!ELEMENT programlisting (#PCDATA)>

<!ELEMENT informaltable (tgroup)>
  <!ELEMENT tgroup (tbody)>
  <!ATTLIST tgroup cols CDATA #REQUIRED>
  <!ELEMENT tbody (row+)>
    <!ELEMENT row (entry+)>
      <!ELEMENT entry ((para|%content;)*)>

<!ELEMENT ulink (#PCDATA)>
  <!ATTLIST ulink url CDATA #REQUIRED>

<!ELEMENT appendix (%section;)>
```

B. La feuille de style (extraits de pres2html.xsl)

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:include href="chunk.mod"/>
<xsl:include href="param.mod"/>

<xsl:output method="html" indent="yes"/>

<xsl:template match="/">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="article">
  <xsl:document href="{ $output.path }/index.html">
```

```
<html>
  <head><title><xsl:value-of select="title"/></title>
  <LINK rel="stylesheet" type="text/css" href="{ $css.path }"/>
</head>
<body bgcolor="#ffffff" background="{ $main.background }">
  <table cellpadding="20">
    <tr>
      <td>
        <table cols="2">
          <xsl:apply-templates/>
        </table>
      </td>
    </tr>
  </table>
</body>
</html>
</xsl:document>
</xsl:template>

<xsl:template match="articleinfo">
  <tr>
    <td colspan="2">
      <xsl:apply-templates/>
    </td>
  </tr>
</xsl:template>

<xsl:template match="title">
  <tr>
    <td colspan="2">
      <img SRC="{ $page.icon }" align="left"/>
      <h1 align="center"><xsl:apply-templates/></h1>
    </td>
  </tr>
</xsl:template>

<xsl:template match="copyright">
  <p>Copyright &#xA9; <xsl:apply-templates/></p>
</xsl:template>

<xsl:template match="sect1[position() &lt;= round( last() div 2)]">

  <xsl:variable name="sections">
    <xsl:value-of select="round(count(..//sect1) div 2)"/>
  </xsl:variable>
  <xsl:variable name="next">
    <xsl:value-of select="count(preceding-sibling::*) - 1"/>
  </xsl:variable>

  <tr>
    <td>
      <p><xsl:apply-templates select="." mode="titlepage"/></p>
    </td>
    <td>
      <p><xsl:apply-templates select="//sect1[position() = $next + $sections]" mode="titlepage"/></p>
    </td>
  </tr>
</xsl:template>
```

```
</xsl:template>

<xsl:template match="sect1[position() > ((last() +1) div 2)]">
</xsl:template>

<xsl:template match="sect1" mode="titlepage">
  <a>
    <xsl:attribute name="href"><xsl:value-of select="@id"/>.html</xsl:attribute>
    <xsl:number count="sect1"/>. <xsl:value-of select="title"/>
  </a>
  <xsl:call-template name="write.chunk">
    <xsl:with-param name="filename">
      <xsl:value-of select="@id"/>.html</xsl:with-param>
    <xsl:with-param name="content">

      <html>
        <head><title><xsl:value-of select="title"/></title>
        <LINK rel="stylesheet" type="text/css" href="pres.css"/>
      </head>
      <body bgcolor="#ffffff" background="{ $page.background }">
        <table COLS="1" WIDTH="100%" HEIGHT="100%" cellpadding="20">
          <tr>
            <td>
              <table COLS="1" WIDTH="100%" HEIGHT="100%">
                <tr><td HEIGHT="90%" valign="top">
                  <xsl:apply-templates/></td></tr>
                <tr><td HEIGHT="1%">

                    <xsl:call-template name="chunk.footer">
                      <xsl:with-param name="prev" select="preceding::sect1[1]"/>
                    </xsl:call-template>
                    <xsl:with-param name="next" select="following::sect1[1]"/>
                  </xsl:call-template>
                </td></tr>
              </table>
            </td></tr>
          </table>
        </body>
      </html>

    </xsl:with-param>
    <xsl:with-param name="prev" select="preceding::sect1[1]"/>
    <xsl:with-param name="next" select="following::sect1[1]"/>
  </xsl:call-template>
</xsl:template>

<xsl:template match="sect1/title">
  <table width="100%" cols="2">
    <tr>
      <td width="5%"><img SRC="{ $page.icon }"/></td>
      <td width="95%" valign="top" align="center"><h2><xsl:number count="sect1"/>. <xsl:apply-
templates/></h2></td>
    </tr>
  </table>
</xsl:template>

<xsl:template match="para">
```

DocBook, la quatrième dimension de la documentation technique

```
<p><xsl:apply-templates/></p>
</xsl:template>

<xsl:template match="listitem/para">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="mediaobject">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="imageobject">
  <xsl:apply-templates/>
</xsl:template>

<xsl:template match="imagedata">
  <img>
  <xsl:attribute name="src"><xsl:value-of select="@fileref"/></xsl:attribute>
  </img>
</xsl:template>

<!-- ignore stuff for booklets only -->

<xsl:template match="*[@condition='booklet']">
</xsl:template>

</xsl:stylesheet>
```