Université des Sciences et Technologies de Lille

U. F. R. de Mathématiques Pures et Appliquées

Documents structurés et formats ouverts

Daniel Flipo

Année 2006–2007

Licence de Mathématiques

Sommaire

Introduction

Le but de ce cours de vingt heures est de sensibiliser les futurs enseignants aux problèmes de formats de documents électroniques. L'heure étant à la mutualisation des ressources pédagogiques, il est nécessaire de réfléchir, *avant* de concevoir un document, au choix d'un format assurant

1. sa *pérennité* (pourrais-je ré-utiliser mon document dans 10 ans ?),
2. sa *portabilité* (mon document pourra-t-il être lu sur d'autres systèmes ?).

Nous insisterons sur la nécessité d'un balisage structuré, sur les risques liés aux formats propriétaires et nous présenterons les formats qui nous paraissent actuellement les plus adaptés pour la réalisation de documents de type texte, hypertexte ou multimédia.

Une dizaine d'heures sera consacrée à l'étude des bases des langages XHTML et LaTeX avec mise en œuvre pratique sur ordinateur.

1 Notion de document structuré

Lors de la conception d'un document textuel, beaucoup d'auteurs ont en tête l'aspect qu'ils veulent donner à leur document alors qu'il serait beaucoup plus efficace de se concentrer sur la structure, la mise en page en découlant ensuite.

Exemple d'une lettre : on peut positionner « à la main » les différents champs (adresse de l'expéditeur, adresse du destinataire, date du jour, formule de politesse, etc.) ou bien faire la liste des champs utiles, leur affecter des valeurs, et laisser à une *feuille de style* le soin de placer ces éléments dans la page.

Exemple d'un article comprenant plusieurs sections : l'approche, malheureusement fréquente, qui consiste à saisir les titres de section selon un format visuel (gras, 18 points, précédés d'un numéro…) est calamiteuse ! L'ajout d'une section oblige renuméroter les suivantes ; si une table des matières est souhaitée, elle devra être faite à la main, etc. La bonne approche consiste à utiliser une *feuille de style* adaptée à la *structure* du document.

Avantages d'un balisage structurel :

— La mise en page change selon les supports (papier, poster, page WEB…), la structure est une caractéristique intrinsèque du document.
— À partir de la structure et d'un fichier de configuration externe (feuille de style) — conçue, non par l'auteur, mais par un spécialiste ! —, il est facile à un programme de produire une mise en page adaptée.
— La démarche inverse, reconstituer la structure à partir de la présentation, est impossible à réaliser automatiquement.

— Un balisage structurel rend possible la conversion automatique d'un format à un autre.
— Un balisage structurel rend possible l'extraction de données pour créer automatiquement une table des matières, une liste des figures etc. ou alimenter des bases de données.

Conclusion : un document textuel doit être conçu comme un triplet *structure*, *contenu*, *feuille de style*.

2 Formats de documents : ouverts ou propriétaires ?

2.1 Format d'archivage ou d'échange

Une distinction est à faire entre deux types d'utilisateurs d'un document :
— le ou les auteurs doivent utiliser un format permettant d'éditer (modifier) le document sur une longue période de temps ;
— les destinataires du document ne devraient pas avoir la possibilité de modifier le document, ils devraient avoir un accès *en lecture uniquement*.
Il est affligeant de voir circuler des notes de service en `.doc` !

Il y a lieu de penser à cette distinction auteur/destinataire lors de la mise en ligne de documents.

Il peut être utile, pour les auteurs, d'archiver les versions successives et de pouvoir éventuellement revenir à des versions précédentes du document.

2.2 Les dangers des formats propriétaires

Un format propriétaire (ou fermé) est un format donc les spécifications ne sont pas publiques (exemple `.doc` de MS-WORD).

Archiver un document sous un format propriétaire met son auteur à la merci du propriétaire du format. Le format évoluera au bon vouloir de son propriétaire, qui « l'enrichira » à sa guise pour ajouter de nouvelles fonctionnalités, en ne respectant pas toujours la compatibilité ascendante.

2.3 Formats ouverts et logiciels libres

Un format ouvert est un format dont *toutes les spécifications sont publiques*.

Un logiciel est dit *OpenSource* ou *libre* si son code source est *public, modifiable et redistribuable*.

Il ne faut pas confondre logiciel *libre* et logiciel *gratuit*. Le terme anglais « Free Software » est ambiguë.

Exemple de logiciel gratuit non-libre : Acrobat Reader.

Exemple de logiciels libres et payants : les distributions Linux commerciales de RedHat, SuSe, Mandrake, etc.

2.4 MS-Office & OpenOffice.org

La suite bureautique MS-Office s'appuie sur les formats `.doc` (MS-Word), `.xls` (MS-Excel) et `.ppt` (MS-PowerPoint).

Ces formats cumulent à peu près tous les inconvénients des formats propriétaires :
— absence de portabilité : la suite MS-Office n'est pas disponible sous Linux ni sous Unix en général (sauf MacOSX) ;
— absence de pérennité des documents : essayez d'ouvrir un document produit par Word 2 ou Word 3 avec MS-Office 97 ou 2000...
— problèmes de sécurité : la possibilité d'inclure des *programmes* (Visual Basic) fait des fichiers `.doc` la cible privilégiée des créateurs de virus ;
— problèmes de confidentialité : les fichiers `.doc` contiennent un tas d'informations cachées (chemins d'accès, nom des collaborateurs, anciennes versions du fichier, voir [5] et [6]) ;
— coût financier : au coût initial, il faut ajouter celui des mises à jour indispensables pour pouvoir lire les documents produits par les nouvelles versions ;

Le format `.rtf` est *ouvert*, mais un certain nombre de logiciels qui en produisent (dont MS-Office !) ne respectent pas toujours ses spécifications... ce qui empêche des outils libres comme OpenOffice, Ted, etc. d'ouvrir certains fichiers `.rtf`.

La suite OpenOffice.org constitue une alternative intéressante à MS-Office, elle offre les caractéristiques suivantes :
— elle est disponible sur toutes les plates-formes courantes (Unix, Linux, MacOSX, Windows) ;
— elle permet d'ouvrir et d'éditer la majorité des fichiers `.doc`, `.xls` et `.ppt` ;
— l'exportation aux formats `.doc`, `.xls` et `.ppt` est possible,
— l'exportation au format PDF est intégrée ;
— c'est un logiciel *libre* et gratuit qui n'utilise que des formats *libres* (XML compressé) ; il est facile de vérifier que ces formats ne contiennent ni information cachée ni virus ;

De plus en plus d'organismes publics ou privés abandonnent MS-Office au profit de OpenOffice.org (mairie de Munich, CUDL, Ciments Lafarge, etc.). L'inconvénient que constitue l'absence de compatibilité totale (OpenOffice.org ne permet

pas d'ouvrir à coup sûr tous les fichiers produits par MS-Office) devrait s'estomper au fur et à mesure de la diffusion de OpenOffice.org. La charge de la compatibilité pourrait même changer de camp, si OpenOffice.org parvient à prendre suffisamment de parts de marché.

Conclusion : la migration de MS-Office vers OpenOffice.org s'impose.

2.5 Formats PDF et PostScript

Les deux formats de description de page les plus utilisés sont PostScript et PDF. Tous deux sont sous licence Adobe mais leurs spécifications sont publiques. Ce sont des formats d'échange mais pas des formats d'archivage ; aussi le fait qu'ils ne soient pas libres au sens strict n'est pas vraiment gênant (l'auteur dispose d'un texte « source » qui lui permettrait le cas échéant de ré-éditer son document sous un autre format).

PostScript est un langage de description de page : il permet d'effectuer des opérations (comme tout langage de programmation), ce que ne permet pas PDF. En revanche PDF apporte les fonctionnalités hypertextes qui font défaut à PostScript.

Adobe fournit pour toutes les plates-formes (Unix, Linux, MacOSX, Windows) un lecteur gratuit (non libre) de fichiers PDF : Acrobat Reader. Il existe aussi d'autres lecteurs de fichiers PDF en libre (xpdf, ghostscript etc.).

Le format PDF est un bon choix pour qui veut publier un document (texte, hypertexte ou multimédia) : ce choix assure que le document sera lisible sur toute plate-forme, de plus les documents PDF ne sont pas facilement modifiables (sauf utilisation d'outils adaptés comme Acrobat Distiller).

Le format PostScript est moins recommandé car les postes Windows ne disposent pas de lecteur de fichiers PostScript par défaut (il faut par exemple installer ghostscript et GSView).

Conclusion : privilégier le format PDF pour les échanges de documents qui n'ont pas à être modifiés par le destinataire.

2.6 Formats multimédia

Pour les images, les formats SVG et PNG sont libres ; les autres (GIF, JPEG, TIFF, etc.) ont des licences restrictives.

Les logiciels de manipulation d'images (Gimp sous Linux, CorelDraw sous Windows, etc.) permettent d'importer et d'exporter des images sous les différents formats courants. ImageMagic sous Linux est spécialisé dans la conversion d'images (commande `convert`).

Le seul format libre pour le son est « Ogg Vorbis ». Les logiciels xmms et mplayer lisent le format « Ogg Vorbis ». Le format MP3 est, lui, propriétaire.

3 Formats normalisés de balisage

3.1 SGML

SGML signifie « Standard General Markup Language », c'est une norme ISO qui date de 1986.

Il s'agit d'un « méta-langage », ce qui signifie que l'utilisateur peut définir ses propres balises (en nombre illimité) au lieu de devoir se contenter des combinaisons de balises prédéfinies.

C'est un langage universel (parce que méta-) mais extrêmement complexe et lourd à manipuler. Il n'est utilisé que pour les très gros projets. Exemple : toute la documentation des Airbus est en SGML, ce qui permet de produire toutes sortes de documents, des microfiches pour techniciens aux documentions commerciales. La source étant unique, il ne peut y avoir de divergences entre les différents types de documents produits, ce qui est essentiel dans un domaine critique comme l'aéronautique.

3.2 HTML

HTML signifie « HyperText Markup Language », c'est un langage à balises dérivé de SGML et particulièrement adapté aux documents à afficher sur le WEB.

Ce n'est pas un méta-langage : il ne dispose que d'un nombre limité de balises, l'éventail des balises disponibles a augmenté au fil des versions : HTML 2, HTML 3.2 et HTML 4.01 (version finale) sont les plus utilisées.

L'avantage de HTML est sa simplicité ; son principal inconvénient est l'absence de possibilité d'évolution des balises (sauf à changer de version). Les concepteurs des navigateurs WEB (Netscape, MS-IE, etc.) qui font partie du consortium W3C World Wide Web Consortium), garant de la définition du langage HTML, ont ajouté, chacun de leur côté, des balises dans le but officiel d'offrir de nouvelles fonctionnalités aux utilisateurs... avec pour conséquence (voulue ?) des incompatibilités (« Cette page optimisée pour MS-IE » ne s'affiche pas sous Netscape et inversement !).

Le consortium finissait pas entériner les ajouts successifs dans une nouvelle version du langage...

En mai 1998, le W3C a figé la norme HTML (version 4.01) et a décidé de passer à un nouveau langage : XML.

3.3 XML

XML signifie « eXtensible Markup Language », comme son nom l'indique c'est un méta-langage (sous-ensemble de SGML).

L'extensibilité de la gamme des balises permet de prendre en compte des situations nouvelles. Exemple : rien n'était prévu pour le balisage des maths en HTML (les formules étaient représentées par des images de type GIF en général) ; en XML une gamme de balises a été développée (MathML).

Un des problèmes de HTML est l'absence de validation de la structure du document. De nombreux documents affichés sur le WEB ont des défauts de structure (balises de fin de champ manquantes, etc.), les navigateurs s'en accommodent comme ils peuvent, le résultat peut différer d'un navigateur à l'autre.

Principe de fonctionnement du langage XML :

1. Un document doit être « *bien formé* », ce qui signifie avoir une structure correcte (balises quelconques mais correctement emboîtées (à toute balise ouvrante doit correspondre une balise fermante, pas de chevauchement du genre `<a> ... <b> </a> ... </b>`).
2. S'il est associé à une DTD (Data Type Declaration) qui définit toutes les balises utilisées dans le document et s'il respecte les spécifications de cette DTD, on dit qu'il est « *valide* ».
3. La DTD ne concerne que la structure. Le langage XSL « eXtensible Style Language » permet de définir les feuilles de styles (Cascading Style Sheets).

Une DTD intéressante est XHTML : elle reprend les balises définies en HTML 4.0, ce qui permet aux utilisateurs de HTML de passer en douceur à XML. La section suivante est consacrée à l'étude de XHTML.

4 Un exemple de DTD XML : XHTML

XHTML est facile à apprendre et permet de réaliser rapidement des pages WEB.

Référence française des standards du web : `http://www.openweb.eu.org/`

La norme XHTML : voir `http://www.la-grange.net/w3c/xhtml1/`

Utilisation sophistiquée des feuilles de style :
`http://www.csszengarden.com/tr/francais/`

Validation d'un fichier XHTML 1.0 : `http://validator.w3.org/`

Validation d'une feuille de style CSS :
`http://jigsaw.w3.org/css-validator/`

FAQ en anglais sur HTML et XHTML : voir
`http://www.w3.org/MarkUp/2004/xhtml-faq`

Cryptage des adresses courriel pour limiter le « spam »:
`http://aspirine.org/emailcode.php`

TP (2 séances de 2h) : réaliser un curriculum vitæ en XHTML avec une feuille de style simple. Faire valider le document et la feuille de style.

5 pdfLaTeX pour les documents scientifiques

7 séances de TP de 2 heures.

5.1 Séance n° 4

— Donner les corrigés de l'exo `CV.html`, validation code+css (1/4h).
— Bases de typographie (voir LaTeX/typographie.tex) (3/4h).
— Présenter TeX, LaTeX, premier document `vide.tex` créé sous emacs.
— Formatage de texte : rôle des espaces, des retours-chariot, exemple d'une liste itemize.

5.2 Séance n° 5

— Les noms de commandes TEX : commencent par `\`, ne contiennent que les *lettres* et avalent l'espace qui suit.
— `\\` (`\newline`), `\linebreak[]` ; `\newpage`, `\pagebreak[]` *à éviter !*
— Le caractère % met en commentaire le reste de la ligne.
— Les 10 caractères spéciaux.
— Les guillemets français, les tirets, les points de suspension, l'euro.
— La saisie des diacritiques inaccessibles au clavier (capitales ou étrangers) : `\'E`, `\‘A`, `\c C`, `\O`, `\aa`, `\AA`, etc.
— Babel : commutation des langues, commandes de saisie pour le français. Doc en français : `http://daniel.flipo.free.fr/frenchb`
— Les commandes de sectionnement, `\tableofcontents`.
— Les références croisées.

5.3 Séance n° 6

— Les environnements de liste.
— Les environnements `flushxxx` et `center`.
— Les tableaux (tableaux de variations apès les maths).

5.4 Séance n° 7 (maths)

— Formules en ligne, hors texte, hors texte numérotées.
Ne pas utiliser `\displaystyle` en ligne !
— Ajouter systématiquement l'extension amsmath.
— Indices, exposants, racines carrées et n-ièmes, fractions.
— `\lim \liminf \limsup \sup \inf \max \min`
— quantificateurs et opérateurs
— sommes et intégrales (y compris multiples, espacement 'dx' et 'd' droit).

5.5 Séance n° 8 (maths suite)

— délimiteurs `\big` etc. et `\left \right`.
— array (matrices, déterminants).
— systèmes d'équations : align et aligned, (+ multline, split, cases ?)
— amsthm ? (pas fait)

5.6 Séance n° 9 (figures et pdfLaTeX)

— inclusion de figures
— utilisation de pdfLaTeX

5.7 Séance n° 10 (compléments)

Définitions de nouvelles commandes : pourquoi ? comment ?

Ce qui n'a pas pu être fait :
— fontes
— expliquer le préambule (codages entrée, sortie...)
— format de page avec geometry
— tableaux de variations

Références

[1] Site dédié aux formats ouverts http://formats-ouverts.org/.

[2] Formats ouverts sur le site de l'AFUL
http://www.aful.org/gdt/interop/formats_ouverts/view.

[3] Autre page WEB sur les formats ouverts http://thomas.enix.org/
pub/ll-utbm/formats_ouverts/formats_ouverts.html.

[4] Livret du libre http://www.livretdulibre.org/.

[5] La fuite d'informations dans les documents propriétaires, Revue MISC, n° 7 (mai-juin 2003).

[6] Le site http://www.nota-bene.org/formats_ouverts donne deux pointeurs sur « l'affaire Blair »:
— http://www.computerbytesman.com/privacy/blair.htm
— http://news.bbc.co.uk/2/hi/technology/3154479.htm

[7] Site OpenOffice.org http://www.OpenOffice.org ou
http://fr.OpenOffice.org (site francophone).

[8] Présenation d'OpenOffice.org
http://www.framasoft.net/article472.html

[9] L'introduction à LaTeX de Vincent Lozano
http://cours.enise.fr/info/latex/