BERNARD PROST

XML POUR L'ÉDITION

STRUCTURER • SAISIR • PUBLIER

ISBN : 978-2-212-12657-0

EYROLLES

a'sfored

SOMMAIRE

Avant-propos

Wait a minute, wait a minute, oh hold it ! Hold it ! Hold it !

Hey (hey) ho (ho) hey (hey) ho (ho) hey (hey) ho (ho) hey
Ray Charles *(What I'd Say)*

If everything is under control, you are going too slow
Mario Andrett

L'édition vit une mutation dont la portée est considérable. Le support papier voit arriver la concurrence de supports dématérialisés chaque jour plus nombreux : le Web bien sûr, l'ordinateur ultra compact, appelé *netbook* – sans qu'il s'agisse d'un hasard –, mais surtout le téléphone et les outils nomades spécialisés comme le *reader* (liseuse) d'e-books ou la tablette qui mettent à la portée de chacun, *urbi et orbi*, des bibliothèques complètes d'œuvres littéraires ou professionnelles. Pour l'éditeur, il s'agit désormais de publier pour ces supports en tenant compte de leurs spécificités, tout en minimisant les coûts et les délais de production. Aux expériences nécessitant des reprises multiples du même fonds documentaire a succédé une approche plus industrielle – mais aussi plus normative ou contraignante – basée sur XML.

DÉFINITION Liseuse

Liseuse est le terme français proposé pour *reader* qui, chez les Anglo-Saxons, désigne le lecteur nomade de livres électroniques (e-book). La liseuse désigne un matériel dédié à la lecture utilisant la technologie d'affichage dite *e-paper*, terme marketing désignant un écran non rétro-éclairé, donc peu consommateur en énergie et réputé moins fatigant à la lecture. Associé à l'*e-paper*, le marketing a introduit la notion d'*e-ink* pour qualifier un pixel…

La souplesse et l'universalité de XML ont séduit aussi bien les éditeurs – en premier lieu les éditeurs juridiques, habitués il est vrai à SGML – que les informaticiens qui ont pu ainsi échanger des données entre ordinateurs aux fonctionnements hétérogènes.

Normalisé en 1999, XML a atteint la maturité : il existe désormais un écosystème XML où l'on rencontre des logiciels spécialisés (les éditeurs XML), des prestataires *on-shore*, *near-shore* et *off-shore* rompus à ce

langage, des développeurs d'applications sachant utiliser le DOM *(Document Object Model)* pour créer des produits électroniques toujours plus innovants, et des modèles documentaires spécialisés verticalement par secteur ou horizontalement par type de publication.

DÉFINITION DOM

DOM *(Document Object Model)* est une modélisation informatique sous forme d'arbre de documents XML ou HTML. DOM est indépendant de toute taxonomie (voir définition plus loin). Cette modélisation permet une manipulation par programme des constituants (éléments) du document.

Pour autant, la pratique n'est pas encore stabilisée et les habitudes diffèrent considérablement d'un éditeur à l'autre. Le propos de cet ouvrage est d'apporter une vision pratique de l'utilisation d'XML dans les maisons d'édition, fondée sur des usages concrets ayant fait leurs preuves mais par nature limités à des cas particuliers. *XML pour l'édition* n'est ni une bible, ni un essai dogmatique sur le sujet, et chacun pourra adapter sa pratique aux exemples fournis.

Structure de l'ouvrage

L'ouvrage est organisé en trois parties – structurer, saisir, publier – qui recouvrent l'ensemble du cycle XML pour une publication multisupport de type « livre ». *XML pour l'édition* s'adresse aux éditeurs, aux lecteurs/correcteurs, aux fabricants, en premier lieu, mais également aux managers qui souhaitent mieux comprendre les techniques sous-jacentes et l'influence du support sur la conception et la forme de l'objet de publication numérique. Enfin, les auteurs curieux des possibilités qu'apporte cette technique y trouveront matière à concevoir différemment leur œuvre.

L'ouvrage s'appuie sur un exemple d'article encyclopédique, de type Wikipedia, qui sera repris au fil des chapitres. Cet exemple fait appel à une structure spécifiquement développée pour le présent ouvrage (`article_v1.2.dtd`). Les balises ont été choisies en langue anglaise afin de se conformer aux usages internationaux. L'exemple répond à un cahier des charges éditorial simple :

- pouvoir publier l'article sur le papier, le Web, le téléphone ;
- en dotant les objets de publication électronique d'interactivité au niveau des auteurs, des bibliographies, des filmographies et des discographies, cette interactivité doit être indépendante des bases de données cibles.

Pour des raisons de simplicité, il n'est pas prévu de tableaux ni de formules mathématiques (en dehors de l'utilisation éventuelle de celles-ci sous forme d'images).

Structurer le XML

DÉFINITION Taxonomie

La taxonomie désigne le jeu de balises utilisé pour l'encodage XML d'un document. La taxonomie est généralement décrite à l'aide d'un langage spécialisé (DTD, XML Schema, Relax NG).

Le premier chapitre s'intéresse plus particulièrement à la modélisation des documents et à la démarche du balisage XML. Le second chapitre est, lui, consacré à la description des principales structures que l'on peut rencontrer dans un ouvrage, ou plus généralement dans un document. Le troisième chapitre donne enfin les clés pour écrire une DTD, c'est-à-dire une des représentations (la plus simple en fait) d'une taxonomie.

Saisir le XML

Le quatrième chapitre concerne la saisie proprement dite. Dans la plupart des cas, ce travail est confié à un prestataire externe, mais l'éditeur doit de plus en plus être capable de modifier un document à l'aide d'un outil de saisie XML utilisé en interne pour rattraper de petites erreurs ou effectuer des corrections de dernière minute. Ce chapitre aborde en particulier la configuration d'un éditeur XML du marché et son utilisation dans le cadre d'une DTD particulière.

Le cinquième chapitre s'intéresse à la relation avec les sous-traitants : comment préparer la copie pour minimiser les risques d'erreurs d'interprétation au niveau de la structure par le prestataire et bâtir des consignes de saisie efficaces ?

Le sixième chapitre est ensuite consacré à une étape rarement décrite dans les processus de production : l'épreuvage XML. Comment s'assurer que le XML fourni par le prestataire est conforme aux attentes de l'éditeur ? Ce chapitre aborde également les différents modèles de production XML selon que la saisie XML s'effectue en amont de la mise en pages papier, pendant celle-ci ou a posteriori.

Publier

Le septième chapitre expose, dans leurs grandes lignes, les techniques de transformation d'un document XML en un format cible qui peut être aussi bien du XML (par exemple en entrée d'InDesign), du XHTML ou tout autre format textuel. Bien que très technique, le langage de transformation XSLT n'a rien de mystérieux et il importe que les acteurs de l'édition en comprennent le fonctionnement pour apprécier l'impact d'une décision éditoriale.

Le huitième chapitre décrit brièvement la publication pour les supports électroniques en se limitant au Web, aux tablettes, aux liseuses et à l'iPhone choisi comme représentant le plus abouti (actuellement) de la lecture sur téléphone.

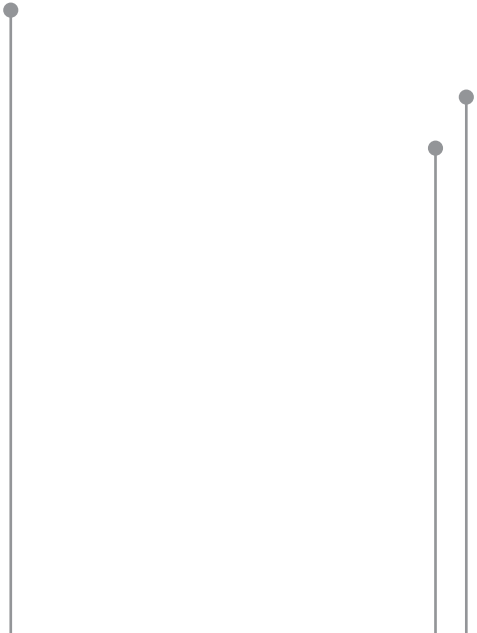
Le neuvième chapitre aborde enfin la publication sur papier à partir d'un document XML, selon deux approches :

- la transformation directe du XML en PDF en utilisant XSL-FO, langage de mise en pages lui-même exprimé en… XML ;
- l'importation du XML, éventuellement remanié, par un outil de PAO (ex. : InDesign).

Cet ouvrage vise à donner les clés d'utilisation de XML dans la chaîne éditoriale et se limite à l'essentiel de cette approche moderne de l'édition ; pour chacune des techniques abordées, il existe, aussi bien en français qu'en anglais, des ouvrages très complets auxquels le lecteur se référera avec profit.

À NOTER

Le vocabulaire du domaine XML est assez opaque. De nombreux termes sont chargés de références à SGML, aux feuilles de styles, etc., qui ont perdu leur sens d'origine et ne sont plus du tout représentatifs de leur fonction. Il faudra donc faire l'effort de les mémoriser indépendamment de leur sens usuel en français (ou en anglais) courant.



Préparer et diriger la saisie XML

La saisie XML par l'auteur, rêve de tout éditeur, n'est pas pour tout de suite. La réalité s'appelle « Word » du côté de l'auteur, et « opérateurs de saisie XML multiclient et multi-DTD », du côté du prestataire.

La préparation et l'organisation de la saisie XML passent par des choix simples qui commencent dès la rédaction du document Word par l'auteur et se poursuivent par des consignes de saisie permettant au prestataire de minimiser les risques d'erreur.

Styler le document Word

Auteur et éditeur travaillent généralement sous Word. L'atelier de saisie utilise des outils orientés XML et rencontre des ambiguïtés structurelles dues à une interprétation du texte et au niveau disparate des opérateurs de saisie. Utiliser les fonctions de styles de Word pour décrire la structure XML peut lier ces deux mondes.

Qu'est-ce qu'un style dans Word ?

Tout traitement de texte moderne comme Word ou Open Office dispose de fonctionnalités de styles permettant sous un seul label de regrouper tous les paramètres typographiques qui pourront être appliqués à des paragraphes ou à des portions de texte. L'avantage recherché est de pouvoir changer globalement l'apparence typographique de l'objet stylé : si tous les titres de niveau 1 ont reçu un style (par exemple Titre1), il sera possible en une seule opération de changer leurs attributs typographiques (retrait, taille de police, couleur…).

OPEN OFFICE

Open Office est le pendant open source de la suite Office de Microsoft. Bien que très puissante, agréable à utiliser et robuste, la suite Open Office n'a pas réussi à s'imposer auprès des auteurs et des entreprises. www.openoffice.org

Dans ses différentes versions jusqu'à 2007, Word distingue deux principaux types de styles :

- les styles de paragraphe qui correspondent à la notion de bloc en XML, c'est-à-dire un ensemble généralement composé dans un rectangle ;
- les styles de caractère qui correspondent à la notion de texte inline en XML.

Dans la suite de l'ouvrage, nous parlerons de style bloc (au lieu de style de paragraphe) et de style inline (au lieu de style de caractère) afin d'unifier le

discours entre les styles des traitements de texte, les styles CSS et les balises HTML lorsque la distinction est nécessaire.

REMARQUE

Word propose en réalité 4 types de styles : paragraphe, caractère, tableau, liste. Open Office en propose davantage.

Il est d'usage de parler de « feuille de styles » pour désigner le regroupement de tous les styles utilisés. Ce terme vient de ce que les documents Word, comme dans tout traitement de texte, peuvent s'appuyer sur un modèle externe au document – appelé *template* dans le jargon du domaine – qui regroupe ces styles. Celui-ci est créé à partir d'un document Word de départ comportant tous les styles et sauvegardé sous forme de modèle. Il porte l'extension .dot qui signifie : *document template*.

Le rattachement du modèle au document courant se fait par une commande du type `Outils>Modèles et compléments` qui diffère d'ailleurs d'une version de Word à l'autre…

Objectifs du stylage

Ce qui est recherché lors d'un stylage, c'est avant tout de la productivité dans la phase de mise en pages des publications.

En première intention, il s'agit de simplifier le coulage des textes dans les outils de PAO : grâce aux styles, les attributs typographiques sont plus facilement posés, le prestataire gagne en temps de mise en pages et l'éditeur en coûts.

En seconde intention, avec l'introduction de XML, le stylage est devenu le moyen de préparer la saisie XML. Le repérage des différents composants sémantiques d'un texte, titres de niveau, paragraphes, listes, attributs typographiques inline par des styles permet en effet par un mécanisme de conversion d'obtenir l'essentiel du XML attendu. En même temps, cette démarche laisse à l'auteur son outil de prédilection et ne perturbe pas, du moins en théorie, ses méthodes habituelles de travail.

REMARQUE

Dans la pratique, le compuscrit (texte rédigé sur un ordinateur) fourni par l'auteur est rarement conforme à la feuille de styles, ce qui exige un travail de nettoyage a posteriori, faisant perdre une partie du bénéfice attendu du stylage.

Les limites du stylage à v ocation XML

Le stylage à vocation XML présente une première limitation majeure : l'imbrication des éléments à la manière des poupées russes, essentielle en XML, est impossible au niveau des styles bloc (c'est-à-dire des styles de paragraphe). Exprimer qu'un paragraphe appartient à un contexte de division en termes de style se pose de la manière suivante : si l'on appelle `div` le style de division et `p` le style du paragraphe standard, il faudrait pouvoir exprimer `div p`. La seule manière de le faire est de créer un nouveau style bloc appelé par exemple `div-p`. En admettant que la `div` puisse recevoir des paragraphes, des listes, d'autres `div`, la combinatoire devient explosive. Le nombre de styles est excessif et leur utilisation devient illusoire.

Le stylage présente une deuxième limitation : l'utilisateur du traitement de texte peut à tout moment « surcharger » un style. Cela signifie qu'un style peut être modifié localement pour répondre à un besoin éditorial. Un titre stylé `Titre1` peut tout à fait être surchargé avec des paramètres typographiques du style `Titre2` pour exprimer l'apparence d'un titre de second niveau : le traitement de texte l'acceptera techniquement et le lecteur du document interprétera cela comme un titre de niveau 2.

La troisième limitation des styles est leur extrême perméabilité à l'importation d'autres styles. Il suffit de faire l'expérience à partir d'un copier/coller depuis un document HTML pris n'importe où sur Internet. Après collage, la liste des styles Word s'est enrichie (!) de nouveaux styles aux noms imprévisibles issus des attributs insérés dans les balises HTML du texte importé.

La quatrième limitation est l'absence de règles de structure : il est tout à fait possible de mettre un titre de niveau 2 (style `Titre2`) dès le début d'un document Word, tandis que la structure XML interdirait ce cas de figure.

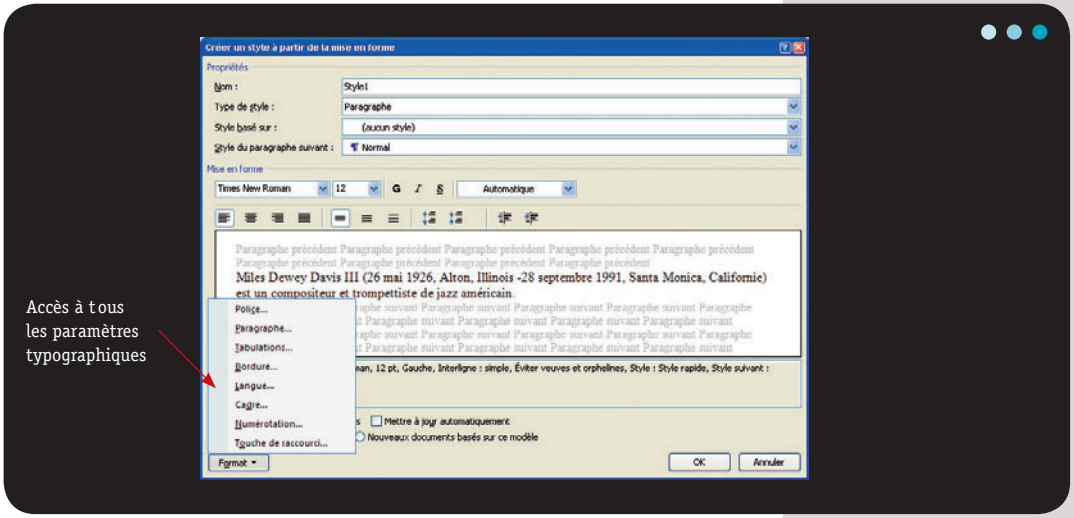
La cinquième limitation est le rendu différent de la même feuille de styles dans les différentes versions de Word. Ce n'est pas très gênant, sauf pour l'auteur qui se trouve confronté à une réalité différente visuellement de celle proposée par les consignes d'écriture fournies par son éditeur.

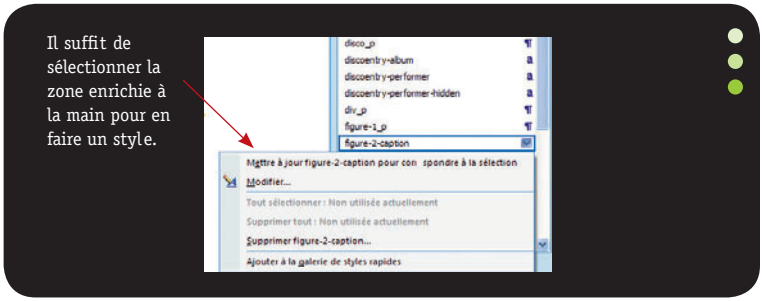
Déclarer un style dans W ord

Word, comme Open Office, propose deux manières de déclarer un style :

- directement, via des boîtes de dialogue permettant de choisir de manière précise les attributs typographiques : police, taille, espace avant, espace après… Cela impose une connaissance fine des paramètres typographiques et une maîtrise des boîtes de dialogue correspondantes dans Word ;

- indirectement, via une mise en forme préalable et manuelle de l'objet à styler (paragraphe ou caractères), en le sélectionnant et en lui attribuant a posteriori un nom de style : c'est la méthode la plus intuitive, car guidée par une approche visuelle mais moins précise.





● Définition directe d'un style (Word 2007)

● Définition indirecte d'un style en s'appuyant sur une mise en forme préalable.

Nommer un style

Le nommage des styles est laissé à l'initiative de la personne qui manipule le traitement de texte. Dans une double perspective de préparation du texte au passage XML et de productivité, il convient cependant de respecter quelques règles :

- les noms de styles doivent être assez proches du nom de l'élément XML qui leur correspondra. On choisira ainsi le nom `bib_p` pour exprimer un paragraphe bibliographique dont l'élément XML est `pbib`;

- les noms de styles doivent incorporer, quand cela est perçu comme nécessaire ou utile, le pseudo chemin représentatif de la position dans la structure. On nommera ainsi `bibentry-booktitle` le titre d'un ouvrage dans une bibliographie, dont le correspondant XML est `booktitle`;
- les noms de styles sont accessoirement munis d'un suffixe pour exprimer le fait qu'il s'agit d'un style de paragraphe ou d'un style de caractère. On choisira par exemple `_p` pour les styles de paragraphe et `_c` (ou pas d'extension du tout) pour les styles de caractère.

L'intérêt de suffixer par `_p` les styles de paragraphe est de rappeler à l'utilisateur qu'il s'agit d'un style bloc lorsqu'il fait son choix dans la liste déroulante des styles. Certes, Word propose un petit indicateur de nature de style (à droite du nom de style), mais peu d'utilisateurs le connaissent ou le voient.

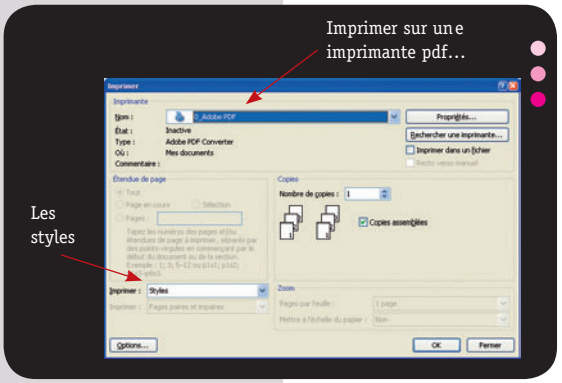
DES STYLES PARTOUT

Encore une polysémie redoutable : il est difficile de s'y retrouver lorsque l'on parle de styles. Il existe le style du traitement de texte, celui de la CSS, celui des feuilles XSL, le style de l'auteur… Un effort intellectuel à consentir !

Par déduction, la plupart des autres styles ne sont pas suffixés. Il s'agit soit de styles de caractère, soit de styles réservés de Word, en l'occurrence les styles de niveau de titres. En effet, ceux-ci sont bien connus des utilisateurs et sont automatiquement interprétés par les Word localisés, c'est-à-dire en d'autres langues que le français (l'ouverture de la feuille de styles dans une traduction américaine de Word remplacera `Titre1` par `Head1`, par exemple).

Lister les styles de W ord

● Récupérer la liste des styles du fichier ouvert



Il est fort pratique de disposer d'une liste des styles lorsque l'on veut gérer une feuille de styles de référence. Word ne propose pas un export direct : dans Word 2003 (la procédure est proche pour les autres versions), il faut passer par la commande `Fichier>Imprimer` et choisir `Imprimer:Styles` dans la liste déroulante. Le plus simple est d'imprimer en PDF afin de récupérer la liste des styles par copier/coller, soit dans votre traitement de texte, soit dans Excel.

Une autre méthode, plus technique, consiste à enregistrer le document Word en format « Page Web, filtrée » et à ouvrir le HTML correspondant avec un éditeur de texte standard (ex. : Notepad ++). Les styles sont déclarés sous forme d'une CSS

Publier pour l'électronique

Publier pour l'électronique est un art éphémère. La principale difficulté est liée aux multiples cibles (site web, téléphone, tablette, liseuse) et donc à leurs caractéristiques propres de taille d'écran et de logiciels de lecture fortement dépendants de l'environnement employé. Cela impose pour le court terme de générer de multiples formats, et ainsi de produire des fichiers différents à gérer en aval au niveau des plates-formes de distribution.

Publier au format epub

epub est le format de publication préféré des éditeur s. Ouvert et élaboré par l'industrie de l'édition américaine via l'IDPF , il sert de base à la plupart des publications électr oniques de livres. Universel et simple, il offre l'interopérabilité (lecture en synchronie d'un fichier sur écran, tablette et téléphone…).

Le format epub

ACRONYMES

IDPF : International Digital Publishers Forum (`www.idpf.org`)
OPF : Open Packaging Format

Le format epub, quasiment devenu la norme pour la publication d'e-books, est un format très simple destiné à la publication de livres électroniques. Sa mise au point, inspirée de trois normes préexistantes (OPS, OPF, OCF) accessibles en ligne sur le site de l'IDPF, s'est appuyée sur des considérations de base et de bon sens :

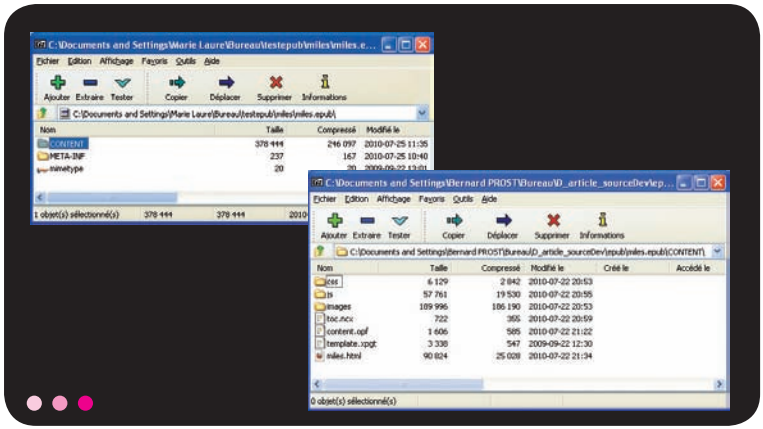
- le fichier de publication doit être unique pour être facilement transportable d'un système à l'autre ;
- le format doit être interopérable, c'est-à-dire fonctionner sur tout type d'appareil et de logiciel se conformant à la norme epub ;
- le texte doit être repaginable (i.e. *flowable*), c'est-à-dire s'appuyer sur un mécanisme permettant de l'inscrire dans l'écran de présentation quel que soit le matériel de restitution – tablette, ordinateur, téléphone – sans avoir à le faire défiler verticalement ou horizontalement ;
- l'affichage s'appuie sur une représentation XHTML (prochainement HTML 5), avec ou sans CSS ; il s'agit d'un HTML complet avec toutes les possibilités d'hyperliens. JavaScript est autorisé, mais son interprétation dépend des logiciels de lecture du fichier epub, ce qui, actuellement, freine son interopérabilité ;

- le découpage des fichiers est libre : on respectera par exemple le découpage en parties, chapitres et sections que l'on fera apparaître dans la table des matières. Rien n'empêche – à l'exception notable de certaines liseuses qui n'acceptent pas de gros fichiers (300 Ko maximum pour la liseuse de Sony), de n'utiliser qu'un seul fichier ;
- la navigation s'appuie sur un fichier XML, généralement suffixé `.ncx`, représentant l'ordre et la hiérarchie des entrées de la table des matières. Celles-ci pointent sur les fichiers XHTML grâce au mécanisme classique `id/idref` de XML ;
- l'ensemble des fichiers nécessaires à la publication, fichiers XHTML, images et plus généralement ressources (vidéo, son) est décrit dans un seul fichier XML, nécessairement suffixé `.opf`, ce qui permet au niveau des outils de contrôle de s'assurer que toutes les ressources invoquées dans l'epub sont bien présentes ;
- les fichiers sont organisés en trois groupes – un fichier unique et deux répertoires – respectant des règles d'organisation ;
- avec quelques contraintes de priorité, la totalité des fichiers est zippée en un seul fichier dont l'extension est `.epub` et non pas `.zip`.

Créer un epub

Ouvrir un epub existant

L'epub étant un fichier zippé, il est simple de l'explorer en l'ouvrant à l'aide de n'importe quel outil sachant reconnaître le format zip ; on choisira soit un outil open source comme TUGZip (`www.tugzip.com`) ou 7-Zip (`www.7-`



Ouverture d'un epub avec 7-Zip : les fichiers peuvent être modifiés soit en les extrayant par copier/glisser sur le bureau et en les réintégrant après correction, toujours par copier/glisser, soit en les éditant directement (touche F4 à condition d'avoir préalablement associé un éditeur à 7-Zip via la commande Outils>Option).

zip.org), soit un outil commercial comme winzip (www.winzip.com) ou winrar (www.win-rar.com), tous fonctionnant en mode graphique sous Windows. Ce type d'outil permet de retoucher facilement l'epub (la CSS en particulier) en extrayant localement le fichier à modifier et en le réintégrant dans l'epub par copier/coller après correction.

À noter : en raison de leur gestion spécifique de l'ordre des fichiers, 7-Zip comme TUGZip en mode graphique ne permettent pas d'obtenir des epub irréprochables lorsque l'on crée une archive au format epub. En ouverture et retouche de fichiers en revanche, ils ne posent pas de problème.

Créer manuellement un epub

La création manuelle d'un epub est une opération simple, une fois le HTML disponible. Conformément à la norme, un epub comporte trois fichiers obligatoires, dont deux (mimetype et container.xml) ont un nommage imposé, et deux dossiers obligatoires dont un seul (META-INF) a un nommage imposé :

1. Le fichier mimetype (en minuscules et sans extension), dont le nom est imposé, ne comporte qu'une seule ligne de texte :

```
application/epub+zip
```

Ce fichier doit être placé en premier dans l'archive .epub et ne doit pas être compressé. Stocker un si petit fichier peut sembler bizarre, mais il permet à l'application de lecture – Stanza, iBooks, etc. – d'identifier très rapidement un fichier de type epub avec plus de certitude que la seule extension .epub.

2. Le dossier META-INF, dont le nom (en capitales) est imposé, contient au moins un fichier dont le nom (container.xml) est également imposé. Ce dernier est un fichier XML dont le vocabulaire, contrôlé par une DTD, est fixé par la norme. Son élément racine est <container>. Il contient a minima, et dans la plupart des cas, un accès à un fichier de configuration de l'epub (xxx.opf) :

```
<?xml version="1.0" encoding="UTF-8"?>
<container xmlns="urn:oasis:names:tc:opendocument:xm
lns:container" version="1.0">
 <rootfiles>
 <rootfile
          full-path="CONTENT/package.opf" media-
          type="application/oebps-package+xml"/>
 </rootfiles>
</container>
```

Les attributs `xmlns` et `media-type` doivent être repris tels quels ; leur complexité n'est qu'apparente. Contrairement à `mimetype` le dossier `META-INF` peut être compressé, mais ce n'est pas obligatoire, surtout en phase de mise au point de l'epub.

ATTENTION À LA COMPRESSION

Lors de la mise au point de l'epub, il est préférable de ne pas compresser les dossiers : on gagne en rapidité et en facilité, les paramètres étant fixés sur « aucune compression ». Ce n'est qu'au moment de la publication que l'on modifiera ceux-ci.

3. Un dossier dont le nom est libre, par exemple `\CONTENT` (on trouve presque toujours `OEBPS`), où sont regroupés tous les fichiers de contenu, HTML, CSS, JavaScript, etc. ainsi que deux fichiers obligatoires :

ACRONYME OEBPS

Open EBook Publication Structure

ACRONYME NCX

Signifie soit *Navigation Control for XML*, soit *Navigation Center eXtended*, suivant les écoles. Il s'agit de la table des matières.

– `xxx.opf`, généralement appelé `package.opf` (on trouve aussi `content.opf` et `meta.opf`). Ce fichier a nécessairement l'extension `.opf` et il doit être unique à porter cette extension au sein de `\CONTENT` ;

– `xxx.ncx`, presque toujours appelé `toc.ncx` bien que le nom ainsi que l'extension soient libres.

`Package.opf` et `toc.ncx` nécessitant d'être bien compris dans leur structure si l'on veut maîtriser le format epub, nous les décrivons ici de manière plus détaillée.

package.opf

Le fichier `package.opf` est un fichier XML unique dont le vocabulaire est contrôlé (élément racine `<package>`). Il comprend trois sections :

- les métadonnées (`<metadata>`), conformes à la taxonomie du Dublin Core (les éléments sont donc préfixés `dc`). Selon la norme, seules trois métadonnées sont obligatoires : `<dc:title>` pour le titre du document, `<dc:identifier>` pour stocker un identifiant, `<dc:language>` pour la langue principale du document :

```
<metadata>
        <dc:title>Miles davis</dc:title>
        <dc:identifier id="prostbookid-876523-
        456">prostbook</dc:identifier>
        <dc:language>FR</dc:language>
</metadata>
```

- iBooks utilise deux métadonnées de fait obligatoires si l'on veut figurer correctement dans la bibliothèque d'e-books de l'iPad :
  - `<dc:subject>` pour faire apparaître la catégorie de rattachement du livre ;
  - `<dc:creator>` pour faire figurer l'auteur.
- la déclaration de toutes les ressources (`<manifest>`), c'est-à-dire tous les fichiers utiles pour l'epub : la table des matières (`toc.ncx`), éventuellement le fichier de configuration d'Adobe pour Adobe Digital Edition (`.xpgt`), la CSS principale et ses CSS dépendantes s'il y a lieu, tous les fichiers XHTML, toutes les images, etc. Chaque ressource est décrite dans un élément `<item>` muni de 3 attributs : `id` pour identifier la ressource, `href` pour pointer sur le fichier ressource et `media-type` pour qualifier techniquement la ressource :

<MANIFEST>

To manifest signifie « montrer » en anglais. Un manifest est un fichier montrant toutes les ressources utilisées dans un projet. Ce mot, très général, se retrouve dans de nombreux environnements informatiques.

```
<manifest>
<item id="miles_germany"
href="images/miles_germany.jpg"
media-type="image/jpeg"/>
</manifest>
```

Ces informations sont, entre autres usages, utilisées par les outils de contrôle pour vérifier la conformité du format du fichier déclaré dans `href`.

- un fichier squelette (`<spine toc="ncx">`) donnant les identifiants des fichiers XHTML utilisés pour l'affichage du texte du livre dans les pages. Dans l'exemple d'`article_v1.2`, il n'y a qu'un seul fichier xhtml, donc une seule information dans `<spine>` :

SPINE

Il est important de comprendre que le `<spine>` ne sert qu'à donner l'ordre d'affichage des fichiers dans les pages du livre lors du feuilletage. Si l'on met en premier le chapitre 2, le feuilletage du livre commencera par le chapitre 2, même si la table des matières est donnée dans la séquence correcte. Les deux notions sont donc indépendantes.

```
<spine toc="ncx">
        ......
    <itemref idref="main"/>
        ......
</spine>
```

L'attribut `toc` est obligatoire. Dans cet exemple, l'identifiant auquel il est fait référence (`idref="main"`) renvoie à un identifiant utilisé dans la zone `<manifest>` sous la forme `<item id="main" href="miles.html">`.

toc.ncx

Le fichier `toc.ncx` est un fichier XML unique à vocabulaire imposé (racine `<ncx>`) ; il regroupe :

- un en-tête `<head>` listant quatre valeurs de métadonnées sous forme de paires name/content :

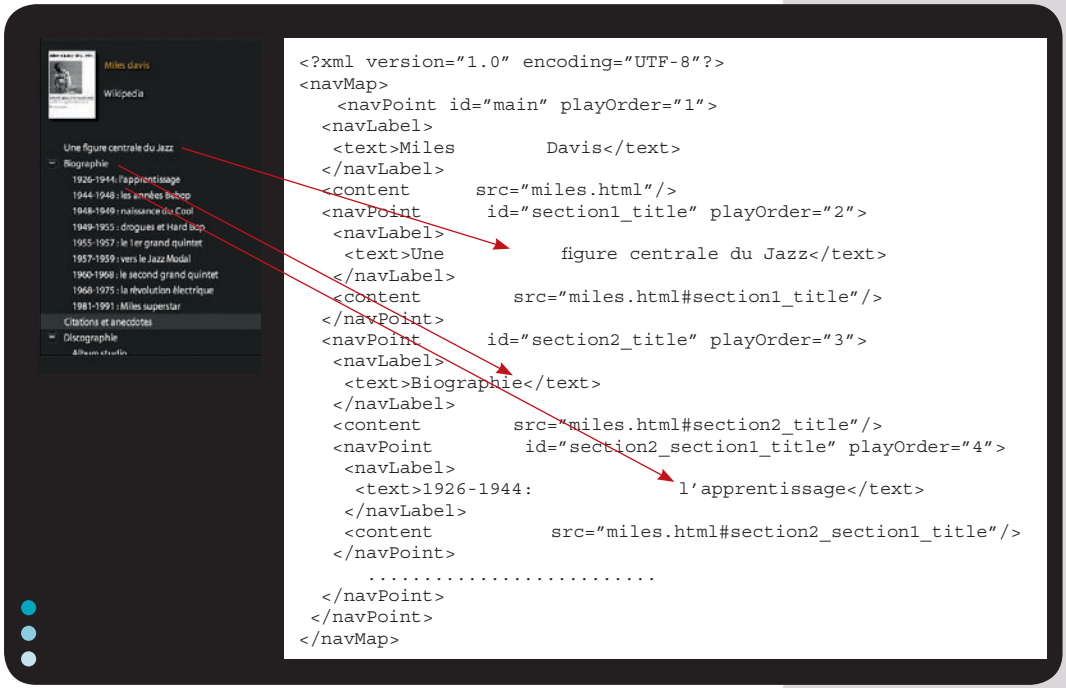
– `<meta name="dtb:uid" content="xxx">` : xxx est un identifiant unique (uid signifie user id) pour l'ouvrage. Ce peut être un numéro ISBN ou tout identifiant généré par un programme permettant de produire des uuid (*universal user id*) ;

– `<meta name="dtb:depth" content="N">`: N définit la profondeur de la table des matières sous forme d'un entier. `depth = "1"` correspond aux niveaux supérieurs d'un livre (front, body ou back) ;

– `<meta name="dtb:totalPageCount" content="N">` : métadonnée qui n'a pas d'effet pratique en matière d'e-books ; en pratique, on met cette valeur à 0 ;

– `<meta name="dtb:maxPageNumber" content="N">` : métadonnée qui n'a pas d'effet pratique en matière d'e-books ; en pratique, on met également cette valeur à 0.

La correspondance entre les imbrications de `navPoint` et le rendu hiérarchique de la table des matières dans ADE



- un titre de document `<docTitle>`. On notera la majuscule sur le t de Title, la DTD de définition du vocabulaire `ncx` est un peu ancienne et a conservé les habitudes de la typo dite « pauvre-riche » censée rendre le code plus lisible. En pratique, aucun ereader ne tient compte de cette balise ;
- la navigation elle-même sous forme d'une structure d'emballage `<navMap>` de différentes entrées hiérarchiques (`<navPoint>`). L'élément `navPoint` étant récursif, il permet de définir n'importe quelle profondeur de hiérarchie : un `navPoint` dans un `navPoint` introduit ainsi une section dans la table des matières. `<navPoint>` exige deux attributs `id` et `playOrder`, ce dernier devant respecter une séquence stricte (1, 2, 3, etc.).

Produire l'epub

REMARQUE

Les outils de compression zip permettent, dans une seule archive, de mixer des fichiers avec des taux de compression différents. Le fichier `mimetype` doit nécessairement être stocké avec une compression nulle, tandis que les autres fichiers peuvent être compressés au maximum ou pas du tout compressé si on le souhaite.

Dans le cas d'`article_v1.2` le processus a été simplifié au maximum : le fichier XHTML produit précédemment pour la publication web est repris comme fichier unique XHTML de l'epub. Les fichiers obligatoires `package.opf` et `toc.ncx` sont produits par transformations XSLT depuis le XML source en respectant la cohérence d'ensemble (ordre des fichiers dans le `<spine>` en particulier). Les fichiers sont finalement intégrés dans une archive unique suffixée `.epub` à l'aide de l'utilitaire de zippage préservant l'ordre des fichiers (ex. : Winzip), en deux étapes :

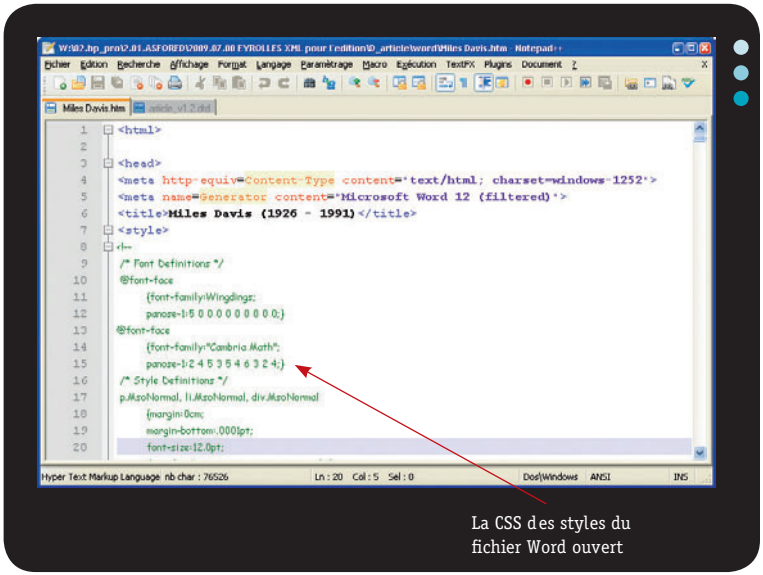
- création de l'archive que l'on baptisera d'entrée de jeu `xxx.epub` (inutile de passer par un nommage en `.zip` et un renommage ultérieur) et intégration du premier fichier `mimetype` sans compression ;
- ajout à l'archive des dossiers `\META-INF` et `\CONTENT` par glisser/déposer avec ou sans compression.

Une fois l'archive créée, il suffit de fermer l'epub et de le contrôler d'abord rapidement en l'ouvrant localement dans Adobe Digital Edition, de manière plus rigoureuse ensuite en le passant dans un *checker* d'epub (programme spécialisé de validation).

Valider un epub

Le format epub comportant de nombreux fichiers dès que l'ouvrage est un peu complexe ou doté d'images, il est facile de lui faire perdre sa conformité à la norme, et cela même s'il s'ouvre sans difficulté dans Adobe Digital Edition : il convient donc de le soumettre, à l'instar de la validation des XHTML, à un

intégrée en début de document Word. Il peuvent même être édités : on éliminera ceux qui ne servent pas, à l'exception des titres de Word qui sont des styles réservés, afin de disposer d'un modèle épuré.



La CSS du fichier Word en mode Page Web filtrée

Enfin, dans Open Office, il suffit de récupérer le XML des styles (fichier `style.xml`) stocké dans le fichier `.odt` et d'extraire les styles intéressants, tout en nettoyant au passage ceux qui sont inutiles.

Gérer les feuilles de styles

Peut-on imaginer une seule feuille de styles pour toute la maison d'édition ? Oui, si la production est homogène : une maison qui ne fait que du roman pourra facilement suivre cette voie. Une maison aux collections diverses devra concevoir plusieurs feuilles, en gardant en commun tout ce qui peut être partagé : les titres, etc.

La gestion des feuilles de styles se doit par ailleurs d'être centralisée : une seule référence doit être disponible sur le réseau de l'entreprise, assortie de sa documentation. Une seule personne doit en être responsable. Dans la pratique, cette contrainte est rarement respectée, la feuille de styles circulant dans l'entreprise et évoluant au gré des besoins, ce qui rend l'approche nettement moins industrielle.