Edition de documents XML bien-formés et valides

Objectif

Utilisation des outils de base pour la création et l'édition de documents XML. Vérification des documents bien formés et validation d'un document par rapport à une DTD. Exemple de transformation avec XSLT vers HTML, XSL puis vers pdf.

Les documents exemples sont fournis à partir de l'adresse `http://sis.univ-tln.fr/~bruno`. Les outils sont installés sur la machine `sis.univ-tln.fr` (**Vous commencerez par mettre en place une connexion avec votre clé publique vers cette machine.**).

1 Edition et validation de documents

Outils utilisables pour l'édition :

- L'éditeur Emacs `http://www.gnu.org/software/emacs/emacs.html`
  - Le mode psgml `http://sourceforge.net/projects/psgml/`
  - Le mode tdtd `http://www.menteith.com/tdtd/`
- Le logiciel commercial xmlspy (`http://www.xmlspy.com`) disponible sous windows ou sous linux avec wine.
- Le logiciel commercial oxygen (`http://www.oxygenxml.com`), `/usr/local/bin/oxygen6.2`
- Un plugin pour eclipse xmlbuddy (`http://xmlbuddy.com/`)

Pour la validation

- Parser xml : Xerces (`http://xml.apache.org/xerces2-j/index.html`)
  Processeur XSLT : Xalan (`http://xml.apache.org/xalan-j/index.html`)
- Opensp (depuis emacs), parser SGML adapté à XML (`http://openjade.sourceforge.net/`).

Un exemple de fichier de configuration de emacs est proposé dans `emacs.dotfile` (à renommer en .emacs).

2 Edition d'un document associé à une DTD (valide)

Le fichier `books.dtd` décrit un modèle de document qui permet de représenter une collection de livres. Le fichier `books-errors.xml` est un document XML exemple qui est une instance de la dtd précédente, il contient des erreurs. Deux types d'erreurs peuvent apparaître. Des erreurs de non-conformité avec le langage XML (balises non fermées, ...), on dira alors que le document n'est pas bien formé; et des erreurs de non conformité par rapport à une DTD, on dira alors que le document n'est pas valide par rapport à cette DTD.

Pour vérifier la conformité d'un document à la norme XML, on utilise un parser :

- La vérification peut aussi être faite depuis Emacs avec le parser `onsgmls (ctrl-c ctrl-v)`
- Ou bien depuis Oxygen

Corriger les erreurs syntaxiques du document `books-errors.xml` pour le rendre bien formé.

Pour valider un documents XML par rapport à une DTD, on ajoute un DOCTYPE dans le document pour le lier à cette DTD, le parser l'utilisera pour vérifier la conformité du document. Rendre le document `books-errors.xml` valide par rapport à la dtd `books.dtd`. Vous pouvez soit réutiliser onsgmls, soit utiliser Xerces, soit oxygen :

- `/usr/local/bin/xerces-validate <doc.xml>` (regarder le script !)

3 Edition d'un document XML

En utilisant l'éditeur **Emacs et le mode psgml** (pour insérer des éléments et créer des attributs) ajouter des livres et des catégories au document exemple
(cf. `http://www.lavoisier.fr/fr/livres/index.asp?texte=xml&select=motcle&exact=on&togo=&support=NULL&from=`).

La documentation du mode psgml se trouve dans le fichier `Emacs-PSGMLQuickReference.html` , et le manuel ici `http://www.lysator.liu.se/~lenst/about_psgml/psgml.html`. Vous utiliserez en particulier :

- C-c C-v (sgml-validate) validates a document. It must be done from the main document file, the one with the DTD declaration at the top. (For our documents that's usually doc/o/*/main.sgml.)
- C-c C-r (sgml-tag-region) tags a region of marked text. The system prompts for a tag name and will not accept an invalid one. You can use the Tab key for auto-fill for the name.
- C-c C-e (sgml-insert-element) prompts for a tag name, and then puts an empty element (start and end tags) at the point at the cursor location. The system checks for valid entries, and offers auto-fill.
- C-c = (sgml-change-element-name) only works if the cursor is inside the element's start or end tag. It prompts for a new tag name and changes the element to what you enter. The system checks for valid entries, and offers auto-fill.
- C-c - (sgml-untag-element) removes the tags of an element. The cursor must be inside the element's start or end tag.
- ESC C-k (sgml-kill-element) removes the entire element with all its children, text, and tags intact. This is useful for moving whole chunks of tagged text within and between documents. The cursor must be positioned before the start tag of the element.
- C-c RET (sgml-split-element) inserts an end tag for the current element followed by a beginning tag for the same kind of element, effectively splitting the element at that point. This is an easy way to make two paragraphs. The cursor must be between (and not inside) the start and end tags of the element.

Vous mettrez éventuellement la DTD à jour, et vous validerez le document modifié.

4 Création d'un définition de type de documents: DTD

4.1 Baliser des documents

Soient les documents suivants `bda.txt` et `inforsid.txt`, qui décrivent un programme de conférence. Proposer un balisage XML de ces documents en créant les documents `bda.xml` et `inforsid.xml`. Le contenu de ces documents est fortement lié à la présentation, vous pourrez (devez ?) l'adapeter pour que les documents XML décrivent la "sémantique".

4.2 Construire une DTD

A partir des balisages que vous venez de définir, proposer une DTD `programme.dtd` et valider les deux documents.

5 Une DTD industrielle, Docbook

La DTD docbook (ainsi qu'une version simplifiée) sont présentées sur les sites `http://www.docbook.org` et `http://www.oasis-open.org/docbook/`. Un exemple simple de document se trouve dans le fichier `sample-docbook.xml`.

La documentation complète de docbook écrite en XML se trouve ici :
`http://cvs.sourceforge.net/cgi-bin/viewcvs.cgi/docbook/defguide/en/`, une version HTML (générée automatiquement) est disponible ici : `http://www.docbook.org/tdg/en/html/docbook.html`.

5.1 Créer un document conforme à la DTD docbook

Ecrire le manuel d'utilisation d'un projet que vous avez réalisé en utilisant la DTD docbook.

5.2 Transformer un document XML

Une document XML peut être transformé en un autre document XML (ou en un fichier texte) en utilisant des feuilles de style XSLT (celles-ci seront étudiées ultérieurement en cours).

Trois feuilles de style sont proposées pour transformer un document XML valide pour la DTD docbook en

- HTML : /usr/share/xml/docbook/stylesheet/nwalsh/html/docbook.xsl
- XHTML : /usr/share/xml/docbook/stylesheet/nwalsh/xhtml/docbook.xsl
- XSL-FO : /usr/share/xml/docbook/stylesheet/nwalsh/fo/docbook.xsl

La commande pour appliquer une feuille de style est : `/usr/local/bin/xalan -IN <source.xml> -XSL <stylesheet.xsl> [-OUT result.???]` (Regarder le script et les autres paramètres !)

Transformer le document de l'exercice précédent en un document HTML, puis en un document XHTML. La troisième feuille de style produit un document XML dans un format appelé FO (Formating Object) qui décrit un document formaté. Ce document peut ensuite être converti vers un format propriétaire (dvi, ps, pdf, ...). Pour réaliser cette transformation, deux outils (en cours de développement) sont disponibles FOP (`http://xml.apache.org/fop/`) et PassiveTeX (`http://www.tei-c.org.uk/Software/p`
Deux commandes possibles `fop` et `pdfxmltex`:

- `fop <fichier.fo> -pdf <fichier.pdf>`
- `pdfxmltex <fichier.fo>`

Les commandes `docbook2{pdf, html, rtf, ...}` utilisent un autre langage de feuilles de style.