Université de Nice-Sophia Antipolis
Polytech' CiP1

Jeudi 8 février 2007
Durée: 1h30

Création et Manipulation de documents

(Hélène Renard / Sylvain Schmitz)

Travaux Dirigés – Séance n^{o}2

---

1 Objectifs du TD

Documents structurés. Notion de syntaxe, et éléments de syntaxe wiki et XHTML.

2 Formats textuels structurés

Si tous les documents textuels peuvent être lus comme un suite de caractères (grâce à un codage de caractères, comme ceux vus la semaine dernière : ASCII, ISO-LATIN, UTF-8,...) leur format réel va souvent bien au-delà du texte brut. Vous connaissez déjà au moins deux formats textuels structurés :

scripts shell donnent en premier lieu quel interpréteur shell employer (`#!/bin/bash` par exemple), puis suivent la syntaxe spécifique à cet interpréteur (sans quoi celui-ci signale une erreur),

code java dont la syntaxe respecte la syntaxe de la spécification Java (sans quoi le compilateur `javac` signale une erreur).

Ces formats textuels respectent une *syntaxe* précise, à laquelle correspond un sens, une *sémantique* : `argument=$1` dans un script shell et `String argument = args[0] ;` dans le `main` d'un fichier Java assignent tous les deux la valeur du premier argument à une variable `argument`.

L'emploi d'une syntaxe précise est indispensable pour tous les langages de programmation. Nous allons voir que la plupart des documents textuels sont eux aussi structurés.

3 Structure hiérarchique d'un document

Vous avez appris au cours de votre scolarité à organiser votre pensée dans des dissertations écrites, souvent dans un découpage introduction - thèse - antithèse - synthèse - conclusion. Chacune de ces parties était à son tour subdivisée en paragraphes portant sur une notion que vous développiez. Ce besoin d'un découpage pour moduler un écrit n'est pas qu'un objectif scolaire, mais un principe général de communication écrite.

On peut considérer pour s'en convaincre des documents aussi divers qu'une lettre de motivation, une documentation technique, un article de journal, un livre, ou encore ce sujet de TD. À chaque fois, un découpage, utilisant paragraphes, sections, voire chapitres et parties, est employé pour associer des idées proches. Cette organisation est particulièrement explicite lorsque l'on emploie des titres. Au sein d'un paragraphe, on peut encore employer des listes pour clarifier la présentation d'éléments associés. Ce type d'organisation est *hiérarchique*, comme nous le montre la figure 1.

Au-delà de la structure en tant que telle, un langage de description de documents devrait permettre de faire référence à d'autres documents (liens hypertextes, images, ...) ou à une portion du document lui-même (index, table des matières, ...).

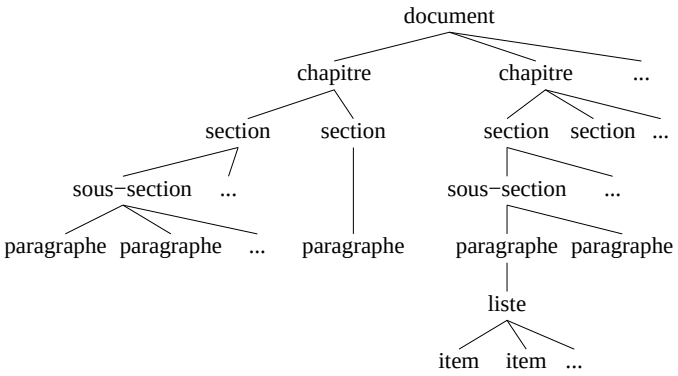

FIG. 1 – Structure hiérarchique, ou *arborescence* d'un document.

3.1 Extraction d'une table des matières

Le fichier `~schmitz/informatique.txt` reprend le contenu de l'article Informatique de l'encyclopédie en ligne Wikipédia. Cet article est assez long pour mériter une table des matières, de manière à ce qu'un lecteur puisse en déterminer le contenu et chercher directement les sections qui l'intéressent.

Les sections sont des lignes isolées du texte commençant par un nombre, potentiellement suivi par un point et un second nombre s'il s'agit en fait d'une sous-section, puis par un espace et le titre de section qui commence par une majuscule. Par exemple, on a la section

```
1.1 Évolution récente
```

Nous avons bien affaire ici à une syntaxe particulière des titres de sections, qui est exploitable de manière automatique.

La table des matières complète que l'on souhaiterait obtenir est donnée dans le tableau 1. Une façon relativement simple d'obtenir cette table des matières est de donner une expression rationelle à `grep` qui corresponde à cette syntaxe. Pour rappel, une expression `grep` peut être :

`^` pour identifier le début d'une ligne,

`$` pour identifier la fin d'une ligne,

`[0-9]` pour identifier les chiffres 0 à 9,

`[A-Z]` pour identifier les lettres majuscules A à Z,

`\( \)` pour grouper plusieurs expressions en une seule,

`e*` pour identifier l'expression `e` zero ou plusieurs fois,

`e\+` pour identifier l'expression `e` une ou plusieurs fois,

`e\?` pour identifier l'expression `e` zero ou une fois.

Par exemple, l'expression `"[A-Z]\+\(.[A-Z]\)\?"` identifiera toutes les séquences d'une ou plusieurs majuscules (`[A-Z]\+`), suivies optionnellement par un point et une majuscule seule (`\(.[A-Z]\)\?`).

Exercice n°1 : Extrayez la table des matières du fichier `~schmitz/informatique.txt` dans un fichier `informatique.toc`.

3.2 Évolutions du document

Un document, et particulièrement un article de Wikipédia, n'est pas figé dans le temps, mais connait des révisisons et des corrections. Nous allons voir deux changements possibles et leurs implications sur notre syntaxe des titres.

Tab. 1 – Table des matières de l'article Informatique de Wikipédia.

3.2.1 Changement de l'arborescence du document

La section « Notes » de l'article décrit les termes anglais pour désigner l'informatique ; à ce titre, elle serait mieux placée en deuxième sous-section de la section « Origine du terme ».

Exercice n°2 : Copiez `~schmitz/informatique.txt` dans votre répertoire `CMDocs` et faites les changements nécessaires. Régénérez la table des matières.

3.2.2 Changement d'un titre de section

La section « La distribution de matériels et logiciels informatiques » traite en fait de spécificités du monde occidental, et il faudrait la renommer en « La distribution de matériels et logiciels informatiques en France, en Europe, et aux États-Unis ». Comme la ligne est trop longue, et on doit la couper entre « France, » et « en Europe ».

Exercice n°3 : Faites cette nouvelle modification dans votre fichier. Que se passe-t'il lorsque vous régénérez la table des matières ?

3.3 Syntaxe wiki

Au vu de ces problèmes, il semble clair que notre syntaxe des titres de sections n'est pas très pratique. La syntaxe wiki réellement employée par Wikipédia pour ses titres de sections est la suivante [1] :

`==Titre de section==` pour les sections,

[1] Wikipédia utilise un joyeux mélange de syntaxe wiki, avec des macro-expansions, des balises XHTML, et même un peu de LaTeX : `http://fr.wikipedia.org/wiki/Aide:Syntaxe`.

`===Titre de sous-section===` pour les sous-sections,

`====Titre de sous-sous-section====` et ainsi de suite sur cinq niveaux.

On appelle un tel encadrement entre signes « = » un *balisage*.

Exercice n°4 : En quoi cette syntaxe permet-elle de résoudre nos deux problèmes ? Regardez le fichier `~schmitz/informatique.wiki` et sa traduction à l'adresse `http://fr.wikipedia.org/wiki/Informatique`. Quelle syntaxe employer pour faire un lien dans un document wiki ?

4 Syntaxes XML

La syntaxe wiki, très simple, a néanmoins plusieurs défauts :

1. le nombre de balisages différents reste limité, ce qui explique l'import de certains éléments syntaxiques du XHTML et de LaTeX dans la syntaxe de Wikipédia ;
2. ces balisages sont avant tout présentationnels avant d'être structurels : par exemple, le seul moyen de connaître la fin d'une section est d'identifier le titre de la section suivante.

Ces défauts sont généralement partagés avec les autres syntaxes de documents usuelles, comme RTF (*Rich Text Format*).

4.1 La syntaxe des balises XML

XML (*eXtensible Markup Language*) est une syntaxe de balises. Les balises XML sont de la forme `<balise> contenu </balise>`. Une balise sans contenu `<balise></balise>` s'écrit plus simplement `<balise/>`. Le contenu entre deux balises peut être en général d'autres balises ou du texte simple. Une balise XML peut avoir des attributs associés, sous la forme `nom="valeur"`, qui la raffinent. Par exemple, on pourrait avoir une balise `<chapitre classe="bibliographie"> contenu </chapitre>` pour préciser que le chapitre en question est une bibliographie. On utilise des *entités* particulières pour représenter les caractères réservés

`&lt;` pour `<` (*less than*),

`&gt;` pour `>` (*greater than*),

`&amp;` pour `&` (*ampersand*).

Cette syntaxe est particulièrement adaptée aux données hiérarchiques. On peut aisément représenter l'arbre de la figure 1 par le texte XML de la figure 2.

Exercice n°5 : Dessinez l'arbre correspondant au texte XML suivant :

```
<phrase>
  <groupe-nominal>
    <determinant mode="possessif" genre="masculin" nombre="singulier">
      Son
    </determinant>
    <nom genre="masculin" nombre="singulier">
      chat
    </nom>
  </groupe-nominal>
  <groupe-verbal>
    <verbe mode="indicatif" temps="présent" genre="masculin" nombre="singulier">
      est
    </verbe>
    <complement classe="direct" genre="masculin" nombre="singulier">
      noir
    </complement>
  </groupe-verbal>
  .
</phrase>
```

```
<document>
  <chapitre>
    <section>
      <sous-section>
        <paragraphe/>
        <paragraphe/>
        ...
      </sous-section>
      ...
    </section>
    <section>
      <paragraphe/>
    </section>
  </chapitre>
  <chapitre>
    <section>
      <sous-section>
        <paragraphe>
          <liste>
            <item/>
            <item/>
            ...
          </liste>
        </paragraphe>
        <paragraphe/>
        ...
      </sous-section>
      ...
    </section>
    <section>

    </section>
    ...
  </chapitre>
  ...
</document>
```

Fig. 2 – Texte XML pour l'arborescence de la figure 1.

4.2 Dialectes XML

Un *dialecte* XML est un langage qui emploie une syntaxe de balises XML pour décrire un type de hiérarchie précis. On obtient un tel dialecte en définissant exactement quelles sont toutes les balises permises, quels attributs chacune peut avoir, et quelles sous-balises sont admissibles pour chacune. Pour l'exemple de la figure 2, on pourra ainsi imposer que

- un document ne peut contenir que des chapitres,
- un chapitre ne peut contenir que des sections, des paragraphes, ou des listes,
- une section ne peut contenir que des sous-sections, des paragraphes, ou des listes, *etc.*

Ces règles sont décrites par une référence adoptée par tous ceux qui emploient le dialecte en question, habituellement sous la forme d'une DTD (*Document Type Definition*) ou d'un schéma XML.

Dans le cadre des langages de documents, on dispose de nombreux dialectes XML (XHTML, OpenDocument, TEI, DocBook, ...). Mais il y a des dialectes pour des données relationelles (RDF), des graphiques vecoriels (SVG), des formules mathématiques (MathML), des définitions d'interfaces graphiques (XUL, XAML), ...

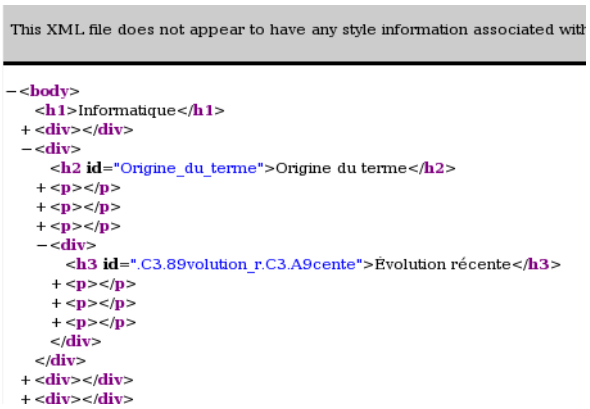

FIG. 3 – Affichage d'un fichier XML pour l'article Informatique.

4.3 Un premier aperçu de XHTML

XHTML (*eXtensible HyperText Markup Language*) est la mise en forme XML du format HTML des pages web. Il reprend les balises d'HTML, mais impose la forme correcte d'XML.

Exercice n°6 : Le fichier `~schmitz/informatique.xml` contient une traduction de la syntaxe wiki en XHTML. Ouvrez ce fichier avec Firefox ; vous devriez voir un affichage ressemblant à celui de la figure 3. Utilisez les signes « - » et « + » à gauche des balises pour cacher/afficher leur contenu. Quelles sont les balises XHTML pour les titres et sous-titres de sections ? Pour les paragraphes ?

Notre fichier XHTML n'est pas correct : il lui manque un en-tête.

Exercice n°7 : Ouvrez maintenant le fichier `~schmitz/informatique.xhtml`. Firefox le reconnaît bien comme un document XHTML, et l'affiche avec un style par défaut. Comment écrit-on un lien hypertexte vers un autre document ?