Optimiser le HTML

Luc Maranget *

Important : Il existe une page de suivi, en `http://www.enseignement.polytechnique.fr/profs/informatique/Luc.Maranget/htmlopt/suivi.html`

1 HTML, très rapidement

HTML est le langage des pages web, c'est du texte, enrichi par des balises, les balises servent à donner au brouteur (*browser*) des indications de formatage du texte. On peut voir le HTML des pages web en les sauvant puis en les regardant avec *emacs*, ou à l'aide de la fenêtre idoine du brouteur.

Les balises (*tags*) commencent toujours par le caractère "`<`" et se terminent toujours par le caractère "`>`". Il existe des balises d'ouverture, par exemple "`<B>`" et des balises de fermeture, par exemple "`</B>`". Les balises d'ouverture et de fermeture doivent être bien parenthésées. Par exemple, pour mettre du texte en gras (*bold*), on écrira :

```
<B>Texte en gras</B>
```

Il existe d'autre balises, pour mettre le texte en italique, en style "machine à écrire", etc. (`I`, `TT`, etc.). L'ensemble du texte entre deux balises est dit un élément (ci-dessus on a un élément "`B`", à rendre en gras donc).

Les outils automatiques de génération de pages web font un usage inconsidéré de ces balises. Par exemple il ne sera pas rare de rencontre, dans le HTML généré par ces outils, ce genre de texte :

```
<B>Texte en gras, </B><B>texte encore en gras</B>
```

Il est clair, que le source HTML ci-dessus est équivalent au source simplifié suivant :

```
<B>Texte en gras, texte encore en gras</B>
```

Remplacer le premier source par le second donne un source plus court, donc qui passera plus vite sur le réseau et sera formaté plus rapidement par le brouteur. Il est important de remarquer dès maintenant que l'effet final (afficher le texte en gras) est inchangé et donc que la transformation est correcte.

2 HTML un peu moins rapide

Évidemment les choses sont un rien plus compliquées que dans la description rapide précédente, à savoir :

– Certaines balises (toujours des balises d'ouverture) ont des attributs. C'est le cas de la balise "`<FONT>`" (modifier la police de caractères), qui peut, par exemple prendre un attribut de taille :
  `<FONT SIZE=5>Texte un peu plus gros que la normale (3)</FONT>`

* `Luc.Maranget@inria.fr`

De notre point de vue, il s'agit d'un élément "FONT". On remarquera que les noms des éléments sont donc des suites de lettres de l'alphabet (ou de chiffres). On apprendra au passage que la casse de ces lettres (majuscules/minuscule) n'a aucune importance (<FONT>...</FoNt> est valide !).
- Un approche un peu moins simpliste des éléments est la suivante. On distingue...
  - Les éléments de niveau texte qui modifient l'apparence du texte, ce sont "B", "FONT" etc.
  - D'autres éléments de niveau texte qui << génère du contenu >>, comme "IMG", "HR", etc. Ces éléments ne possèdent pas de balise de fermeture, ils sont remplacés par une image, un gros trait horizontal etc.
  - Les liens hypertexte "A" qui se comportent en gros comme un élément de niveau texte.
  - Les éléments de niveau bloc, qui constituent grosso-modo une unité de l'ordre du paragraphe. Ce sont par exemple, les éléments "P" un paragraphe, "BLOCKQUOTE" une citation, "TABLE" un tableau, etc.). En outre, les balises de fermeture de certains éléments (par exemple "P") sont optionnelles. Heureusement pour nous, les éléments de niveau texte comprennent toujours des balises de fermeture.
  - Quelques autres éléments, par exemple ceux qui structurent le document HTML : "BODY","HEAD", etc.
  - Il y a des commentaires, <!--*text*-->. On se simplifiera la vie en interdisant les commentaires imbriqués.
- Il y a des *règles de* bonne formation du HTML. Voici les deux plus importantes :
  - Les éléments de niveau texte sont bien parenthésés. Par exemple, le HTML suivant est incorrect.
    ```
    <I>En italique, <B> en gras-italique</I>, en gras</B>
    ```
    On ne peut pas fermer l'élément I à l'intérieur de l'élément B. Autrement dit les balises de fermeture doivent apparaître dans l'ordre inverse des ouvertures. Ici, un HTML correct est :
    ```
    <I>En italique, <B> en gras-italique</B></I><B>, en gras</B>
    ```
  - Il est interdit d'ouvrir un élément de niveau bloc lorsque des éléments de niveau texte sont actifs. Ainsi le HTML suivant est incorrect.
    ```
    <I><BLOCKQUOTE>Citation en italique.</BLOCKQUOTE></I>
    ```
    Il faut écrire :
    ```
    <BLOCKQUOTE><I>Citation en italique.</I></BLOCKQUOTE>
    ```

Du point de vue de l'optimisation on supposera :
- Que le HTML donné en entrée est correct. Ce qui permettra d'échouer sans autre forme de procès en cas d'entrée incorrecte.
- On ne cherchera à optimiser que les seuls éléments de niveau texte.
- On produira du HTML correct.

3 Exemples d'optimisations

L'optimisation déjà présentée de changer <B>*texte1*</B><B>*texte2*</B> en <B>*texte1 texte2*</B> est simple à réaliser : il suffit de supprimer "</B><B>". Un autre optimisation du même style est de supprimer les éléments de niveau texte vides, par exemple "<B></B>".

D'autres optimisations vont demander une certaine interprétation du HTML. Par exemple, il est inutile de mettre un texte en italique deux fois. Ainsi, le texte

```
<I>Texte en italique, <I>texte encore en italique</I></I>
```

S'optimisera en :

```
<I>Texte en italique, texte encore en italique</I>
```

De même, certaines modifications du l'affichage en anullent d'autres ce qui peut conduire à des suppressions de balises, c'est le cas de la taille du texte :

```
<FONT size=4><FONT size=5>Texte en gros.</FONT></FONT>
```

S'optimisera en :

```
<FONT size=5>Texte en gros.</FONT>
```

Les attributs peuvent être groupés, ainsi :

```
<FONT size=5><FONT color=red>Texte en gros et en rouge.</FONT></FONT>
```

S'optimisera en

```
<FONT size=5 color=red>Texte en gros et en rouge.</FONT>
```

Enfin, l'ordre des balises importe parfois peu. Ainsi le HTML :

```
<B><I>Texte en gras et en italique,
</I></B><I><B>la même chose</B></I>
```

S'optimisera en :

```
<B><I>Texte en gras et en italique,
la même chose</I></B>
```

Vous pouvez trouver d'autres optimisations en regardant le HTML produit par les outils automatiques. Par exemple les sujets de projet informatiques, ou produire vous même de telles pages à l'aide de *Netscape Composer*.

4 Comment procéder ?

Il conviendra d'écrire d'abord un *analyseur lexical* qui extrait les balises. Cet analyseur est suffisant pour programmer la première optimisation présentée dans les cas simples où les balises d'ouverture n'ont pas d'attributs.

Pour toutes les autres optimisations, il faut interpréter le HTML. C'est à dire concevoir un modèle simple qui détermine les caractéristiques de présentation (taille, couleur, styles actifs ou inactifs) de chaque bout de texte. Le modèle définit aussi les relations entre les styles (tel style anulle tel autre, tels styles sont exprimables en un seul élément, etc.). Il faudra sans doute réaliser une analyse syntaxique du HTML, mais cette analyse sera partielle. En se contentant d'organiser le document selon les éléments de niveau texte, ce qui revient à analyser un langage de parenthèses, on évitera soigneusement la question plus difficile d'analyser tout HTML.

Au final, le programme doit se présenter comme un filtre c'est à dire un programme qui lit le HTML sur son entrée standard et écrit le HTML optimisé sur sa sortie standard. Cette interface utilisateur simple est obligatoire et doit être respectée.

J'attire votre attention sur quelques points cruciaux :

- Le monde de la spécification des langages broutables est en constante effervescence. Ne vous laissez pas submerger, la référence est HTML 4.01, rien n'étant déprécié.
- Il est demandé un outil simple et pratique. Il est donc conseillé de restreindre votre interprétation de HTML à ce est suffisant (les éléments de niveau texte). Il est parfaitement autorisé et même recommandé de vous servir d'outils existants (tels des validateurs HTML) pour mettre le vôtre au point.

5 Ce qui est demandé

Le minimum est d'écrire un optimiseur *correct* (qui produit du HTML correct à partir de HTML correct et laisse l'aspect du document inchangé) et aussi performant que possible. Le rapport détaillera votre interprétation de HTML, les optimisations réalisées et les outils utilisés.

Votre optimiseur sera essayé sur des pages HTML, dont certaines seront disponibles à partir de la page de suivi du projet. Notez que cette page contient aussi des informations utiles, par exemple sur HTML ou sur l'analyse lexicale.

Il y a au moins deux extensions possibles (attention elles demandent pas mal de travail) :

- La spécification 4.01 de HTML considère certains éléments de style (essentiellement "`FONT`" et pas mal d'attributs) comme dépréciés (*deprecated*) et suggère d'utiliser les feuilles de style (*style sheets*) à la place. Réaliser un traducteur de HTML 4.01 transitionnel (avec des éléments dépréciés) vers HTML rigoureux (sans ces éléments).
  Cela demande d'introduire des éléments "`SPAN`" avec des attributs "`STYLE`" ou "`CLASS`" qui font appel aux << feuilles de style >>. La page de suivi comprend des pointeurs vers touts ces normes.
- Traiter les pages incorrectes en signalant, voire réparant les erreurs. Il faudra alors justifier les réparations.