

Cahiers **GUT** *enberg*

☞ FEUILLES DE STYLE POUR LE WEB

☞ Hélène RICHY

Cahiers GUTenberg, n° 26 (1997), p. 133-145.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1997__26_133_0>

© Association GUTenberg, 1997, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

Feuilles de style pour le Web*

Hélène RICHY

Irisa/CNRS

Campus de Beaulieu, F-35042 Rennes cedex

helene.richy@irisa.fr

Résumé. Les feuilles de styles vont-elles s'imposer sur le *Web*? En attendant de pouvoir répondre à cette question, nous analysons ici cette « nouvelle » approche et ses conséquences dans le contexte du *Web* et des *Cascading Style Sheets*.

Mots-clé : feuille de style, CSS, HTML, *Web*

Les utilisateurs de \LaTeX qui sont habitués à pouvoir changer le style d'un document en ne modifiant a priori qu'une seule ligne (`\documentstyle` ou `\usepackage`) dans leur fichier \LaTeX seront sans doute moins surpris de l'apparition des feuilles de style que les utilisateurs du *Web*.

Nombreux sont en effet les utilisateurs du *Web* qui éditent leurs pages *Web* – en jouant sur les balises de leurs documents HTML (*Hypertext Markup Language* [11]) – dans le seul but d'en modifier la présentation. Et, s'ils n'obtiennent pas l'effet désiré avec leur navigateur favori, ils composent une image (en PostScript, par exemple) pour être sûrs que le tableau ou la formule de mathématique sera convenablement affiché : le document HTML a alors pour seule utilité la communication. Toute utilisation ultérieure des informations – hormis l'affichage tel quel des images – est impossible : impossible de modifier la taille des caractères ou les couleurs, impossible de rechercher des informations par le contenu, par exemple.

En recommandant l'usage de **feuilles de styles** sur le *Web* [7], le consortium W3 bouscule donc ces habitudes, mais renoue avec la tradition de l'édition électronique : une feuille de style permet de mieux contrôler l'apparence des documents et de contribuer à une meilleure qualité typographique et esthétique de ces documents (voir [13] pour une introduction à la notion de feuille de style). Avec cette approche, des styles de différentes origines peuvent être

* Exposé fait lors des *Journées GUTenberg*, Strasbourg, 26–27 mai 1997.

appliqués à un document et même combinés entre eux, si le langage de style le permet.

L'objectif de cette note est de faire comprendre l'intérêt de cette approche et de présenter les *Cascading Style Sheets* (CSS), un langage de style pour le *Web*, en essayant de répondre aux trois questions suivantes :

1. Pourquoi utiliser des feuilles de style sur le *Web*?
2. Comment définir des styles de documents HTML?
3. Quelles sont les principales caractéristiques des CSS?

1. Pourquoi utiliser des feuilles de style sur le *Web* ?

Les documents électroniques du *Web* ne contiennent pas uniquement des informations visuelles (texte, image, graphique), mais aussi des informations sonores et animées (son, vidéo, animation). L'objectif des feuilles de style est de décrire comment ces informations peuvent être affichées sur un écran, imprimées sur un papier ou prononcées et animées. L'indépendance entre le contenu et l'apparence facilite l'adaptation à de nouveaux supports de restitution : par exemple, en permettant de restituer les mêmes informations, soit visuellement (texte, image fixe ou animée sur écran), soit phoniquement (parole, musique).

Cette approche n'est pas totalement nouvelle. Avant d'apparaître dans le *Web*, le concept de feuille de style a été utilisé en PAO et en particulier pour la documentation structurée : le langage SGML (*Standard Generalized Markup Language* [5]) et XML, langage de balisage extensible proposé récemment pour le *Web* (*Extensible Markup Language* [2] [4]) supposent l'usage des feuilles de style.

En se limitant aux objectifs de l'affichage sur écran et de l'impression sur papier, on considère généralement qu'une feuille de style [13] doit décrire à la fois :

la maquette de mise en page (*layout*), à savoir le positionnement et le dimensionnement des pages, des colonnes, l'emplacement des illustrations, des notes, etc.

le protocole typographique qui précise le style des caractères, les couleurs et les formats des éléments qui composent ce document (espace entre paragraphes, titres, sous-titres, etc.).

En dissociant structure et présentation, l'approche des feuilles de style présente les avantages suivants :

- meilleur contrôle sur l'apparence des documents,
- possibilité de définir des consignes de mise en forme sans modifier le contenu des documents¹,
- facilité de spécification, modularité et partage de spécifications entre documents.

En permettant l'application conjointe de feuilles de style, le mécanisme des cascades – supporté par certains langages de style – permet à la fois le contrôle par l'auteur et par le lecteur : les préférences individuelles (contrastes des couleurs ou taille des caractères) peuvent ainsi être respectées.

Telle qu'elle est proposée par le consortium W3, la mise en œuvre des feuilles de style est simple et progressive (voir section 2) : dans certains cas, le contenu d'une feuille de style peut être réduit à une seule règle. De plus, la possibilité de réutiliser une feuille de style pour plusieurs documents facilite la maintenance des sites *Web* qui ont de nombreux documents à gérer.

Non seulement le partage des feuilles de styles, mais aussi le simple fait d'utiliser des feuilles de style à la place d'images volumineuses peut contribuer à diminuer notablement le volume des documents transmis [10] et ainsi permettre un accès plus rapide aux documents.

En effet, depuis un certain temps déjà, les documents électroniques du *Web* contiennent une grande quantité d'images ; beaucoup sont des illustrations (photos, schémas, logos, ...) qui ont pour vocation de faciliter la compréhension internationale des documents électroniques. Mais on peut aussi constater que dans de nombreux cas l'utilisation des images est une façon pour les auteurs d'imposer le *look* de leur documents, même lorsqu'il s'agit de documents essentiellement textuels. Cet usage est fréquent lorsqu'il s'agit d'introduire des tableaux ou des formules de mathématiques, par exemple, dans des documents HTML. En utilisant l'image d'un document, produite par exemple par un programme PostScript, l'auteur est sûr de l'effet produit. Toutefois, cette solution peut nuire à la lisibilité des documents en ne tenant pas compte des contraintes technologiques (caractéristiques des écrans) ou humaines (mal voyants). Le remplacement de telles images par HTML+CSS peut palier ces inconvénients tout en améliorant les performances du *Web*.

1. Sans ajout de balises HTML dans le document : le break (BR), par exemple, est souvent utilisé pour aérer certains textes au milieu de documents HTML !

2. Comment définir des styles de documents HTML ?

Les extensions proposées pour HTML [12] facilitent l'intégration de feuilles de style dans les documents HTML à trois niveaux : soit en plaçant ces règles dans des fichiers séparés qui sont associés au document, soit en regroupant ces règles en tête du document, soit en incluant ces règles dans les éléments HTML qui requièrent une présentation spéciale. Ces extensions sont indépendantes du langage de spécification de style utilisé².

Comme le montrent les exemples décrits ci-dessous, le choix d'une méthode d'intégration dépend du mode d'utilisation des documents et de leurs feuilles de style.

Feuille de style externe Cette méthode offre l'avantage de pouvoir utiliser la même feuille de style pour plusieurs documents. Ainsi, avec les navigateurs capables de stocker ces feuilles de style, il est inutile d'attendre l'arrivée d'une nouvelle feuille de style pour présenter des documents qui font référence à la même feuille de style externe. De plus, la maintenance des styles est facilitée : une seule mise à jour modifie la présentation de tous les documents qui appliquent cette feuille de style. Cette méthode permet, en outre, d'offrir au lecteur un choix entre plusieurs styles.

Feuille de style intégrée Au contraire, si une feuille de style est réduite à quelques règles applicables à un seul document, l'intégration de la feuille de style dans le document est conseillée. Cette intégration est soit directe, soit par importation³ : si certaines règles sont utilisées par d'autres documents, il est possible de constituer des modules indépendants qui seront importés dans ces documents. De nouveaux éléments⁴ et des at-

2. Aucune supposition n'est faite sur la manière de définir des feuilles de style pour les navigateurs : par exemple, un document HTML peut faire référence à une feuille de style utilisant le langage CSS (style de l'auteur), alors que le navigateur peut permettre au lecteur de définir des styles à l'aide d'un autre langage. Si le mécanisme des cascades (voir en 3.5) est prévu par les deux langages, alors le processus de formatage du navigateur appliquera ces deux spécifications « en cascade » pour mettre en forme le document.

3. Cette facilité a toutefois disparu des récentes propositions [12].

4. DIV permet de regrouper des éléments dans une division (chapitre, section, ...) et SPAN permet de désigner séparément certains caractères d'un texte pour leur appliquer un style particulier.

tributs communs⁵ sont définis par HTML pour faciliter la spécification des styles dans les documents HTML.

Style en ligne Au cas où seules quelques modifications de style sont souhaitées, elles sont indiquées directement dans le document HTML (voir aussi certaines extensions de HTML qui peuvent faciliter l'application de styles à certaines parties d'un document).

Ces différentes méthodes sont résumées dans la table 1 et illustrées par les exemples ci-dessous.

– Feuilles de style externe

En utilisant plusieurs éléments LINK, l'auteur peut proposer un choix de feuilles de style au lecteur. Le titre (TITLE) permet de les identifier. Lorsqu'un style est défini sans titre, il est considéré comme un **style permanent** (*persistent style*) que l'utilisateur ne peut pas désactiver. Un attribut TYPE permet d'indiquer le langage de spécification de style utilisé (CSS, , ...). HREF donne l'adresse de la feuille de style. L'attribut MEDIA permet de préciser un médium particulier (*screen, print, aural, projection, braille, all*) pour lequel est définie cette feuille de style.

```
<LINK REL=stylesheet HREF="pretty.css"
      MEDIA=print TYPE="text/css"
      TITLE="Style multicolore">
```

– Feuille de style intégrée

L'élément STYLE est placé dans l'élément HEAD et peut porter les attributs TYPE, TITLE et MEDIA.

```
<HEAD>
  <TITLE>Exemple</TITLE>
  <STYLE TYPE="text/css">
    @import url(http://style/exotic)
    @import url(http://style/national)
    H1 { background: white }
  </STYLE>
</HEAD>
```

5. ID (identifiant unique d'un élément), CLASS (attribut commun permettant de qualifier les éléments) et STYLE (pour définir les styles d'un élément).

TABLE 1 – *Intégration du style dans un document HTML*

Feuille de style (FdS)		HTML	suivi de ...	placé dans ...
FdS externe		élément LINK	adresse de la FdS	élément HEAD
FdS intégrée	directe	élément STYLE	règles de style	élément HEAD
	importée	@import	adresse de la FdS	élément STYLE
Style en ligne		attribut STYLE	styles	élément quelconque

– Style en ligne

```
<BODY>
  <P STYLE="color: green">Paragraphe vert.
</BODY>
```

3. Quelles sont les caractéristiques des CSS?

Nous nous intéressons ici uniquement aux CSS qui font l'objet d'une recommandation récente du consortium W3 (CSS1 [7]). Toutefois, il faut noter qu'il existe d'autres langages permettant de spécifier des feuilles de style. En particulier, le langage DSSSL (*Document Style Semantics and Specification Language* [6]), normalisé en avril 1996, est sans doute actuellement le plus puissant, mais aussi le plus complexe à mettre en œuvre. Il permet de spécifier toute sorte de traitements (restructuration, mise en forme, etc.) concernant des documents SGML ou XML.

CSS est un mécanisme simple de feuille de style qui permet de définir des règles de typographie pour des documents HTML. Ces règles, appliquées par un programme de formatage respectueux de l'ordre des cascades, permettent de contrôler la mise en forme et la présentation des documents.

Des extensions, qui ne sont pas décrites dans ce papier, sont prévues notamment pour améliorer l'impression [14] et la restitution sonore [8]. Seules les caractéristiques générales sont présentées dans cette section. Pour une information complète, le lecteur peut se reporter à la recommandation de W3C ainsi qu'aux nombreux travaux en cours (voir aussi les notes sur le positionnement [3] et sur les *frames* [1] [9]).

3.1. Modèle de formatage

CSS1 suppose que le modèle de formatage est à base de boîtes rectangulaires : à chaque élément correspondent une ou plusieurs boîtes⁶. Du point de vue du formatage, on considère deux catégories d'éléments HTML :

- les éléments-blocs (*block-level elements*) qui, lors du formatage, créent un nouveau paragraphe (c'est à dire coupent le paragraphe courant). Dimension et positionnement sont spécifiés par des propriétés définissant les marges, bordures, remplissage, fond, couleur, etc. (voir en 3.3). Ce sont soit des éléments de liste (*list-item*), soit des éléments flottants (*float* ≠ "none"), soit des éléments appartenant explicitement à cette catégorie (*display="block"*).
- les éléments-en-ligne (*inline elements*) qui peuvent partager une ligne avec d'autres éléments (c'est à dire ne coupent pas le paragraphe courant), ou être coupés sur plusieurs lignes si nécessaire. Ces éléments sont des éléments de texte (*text-level elements*). Ils ne peuvent contenir que d'autres éléments de texte.

Par exemple, les éléments raccrochés (*replaced elements*) peuvent être considérés soit comme des éléments-blocs, soit comme des éléments-en-ligne. Il s'agit d'éléments dont le contenu est spécifié par une adresse. C'est le cas, par exemple, des images (IMG) pointées par un attribut SRC. Ces éléments peuvent avoir leur dimension propre ou être redimensionnés.

3.2. Propriétés

CSS1 définit une cinquantaine de propriétés, ainsi que les valeurs permises pour toutes ces propriétés.

- Ces propriétés permettent de définir : le style des caractères, les couleurs et les fonds, les caractéristiques des textes (espacement, indentation, alignement, ...), les propriétés des boîtes (marges, bordures, dimension et position) et des catégories utiles pour l'affichage (mode de mise en ligne ou en bloc, marque précédant certaines listes, etc.).
- Les valeurs prises par ces propriétés peuvent être exprimées en pourcentage ou en points (pt), en picas (pc), en pouces (in) ou en centimètres

6. Exception faite des éléments ayant un attribut *display* de valeur *one*. Ces élément ne seront pas formatés. Il n'y aura donc pas de boîte correspondante.

(cm)⁷ ou par des mots-clés prédéfinis (pour les couleurs, par exemple) ou comme une adresse *Web* (URL).

De nouvelles propriétés sont proposées dans [14] pour faciliter la mise en page à l'impression (*page-break-before*, *page-break-after*). Par exemple :

```
<STYLE>
  H1 { page-break-before: always }
</STYLE>
```

3.3. Règles de style

Une feuille de style est composée d'un ensemble de **règles de style**. Chaque règle de style est composée d'un sélecteur et d'un style à appliquer aux éléments identifiés par ce sélecteur :

le sélecteur peut être soit l'identification d'un élément particulier (son ID), soit un type d'élément HTML, soit un type contextuel (le contexte est décrit par une séquence d'éléments HTML⁸ dans lesquels doit être placé ce type d'élément pour que la règle de style s'applique), soit un type d'élément dont l'attribut CLASS a la valeur indiquée.

le style est l'association entre une propriété et une valeur pour cette propriété.

Dans l'exemple suivant :

```
#zx789w { line-height: 120% }
H1.section { color: green }
UL UL LI { font-size: 10pt }
```

la première règle s'appliquera au paragraphe dont l'ID a la valeur `zx789w` (décrit en HTML par `<P ID=zx789w>`), la seconde règle s'appliquera aux éléments de type H1 dont l'attribut CLASS a été défini avec la valeur `section` (`<H1 CLASS=section>`). La troisième règle s'appliquera à tous les items de liste (LI) contenus dans une liste non numérotée (UL), elle-même contenue dans une liste non numérotée.

7. 72 pt = 6 pc = 1 in = 2,54 cm

8. Les nouveaux éléments SPAN et DIV sont particulièrement utiles pour harmoniser la présentation de certaines parties d'un document (DIV) ou obtenir certains effets typographiques sur des caractères (SPAN).

Afin d'éviter les répétitions inutiles, la syntaxe de CSS1 autorise les regroupements ; plusieurs styles peuvent être associés à un même sélecteur, plusieurs sélecteurs peuvent être associés au même style, comme dans l'exemple ci-dessous :

```
H1, H2, H3 { font-family: helvetica ; color: red }
```

3.4. Pseudo-classes et pseudo-éléments

Les concepts de **pseudo-élément** et de **pseudo-classe** sont définis par CSS1 pour permettre à des informations externes d'influer sur le formatage du document : les pseudo-éléments désignent des sous parties d'éléments tandis que les pseudo-classes permettent de différencier des types d'éléments. Ils peuvent être utilisés dans les sélecteurs (voir section 3.3) des CSS.

Pseudo-classe *anchor*

Toutes les origines de liens (c'est à dire les éléments A portant un attribut HREF) peuvent appartenir à une et une seule des trois pseudo classes présentées ci-dessous (*link*, *visited*, *active*). Il est ainsi possible d'associer un style différent à ces pseudo-classes de telle sorte que l'ancre apparaisse d'une couleur différente, par exemple, si elle a ou n'a pas été visitée ou si elle est sélectionnée :

```
A:link { color: red } /* lien non visité */  
A:visited { color: blue } /* lien visité */  
A:active { color: lime } /* lien actif */
```

Cette pseudo-classe peut être utilisée dans les règles de style pour préciser un sélecteur. Un sélecteur peut mélanger classes et pseudo-classes, comme dans l'exemple ci-dessous :

```
A.internal:visited { color: green }
```

qui a pour effet de faire apparaître l'origine du lien suivant en vert lorsque ce lien a été visité :

```
<A CLASS=internal HREF="http://www.irisa.fr"  
  Ici, l'Irisa</A>
```

Pseudo-élément *first-line*

Le pseudo-élément *first-line* est utile pour appliquer un style particulier à la première ligne d'un élément après formatage. Par exemple, la règle suivante

```
<STYLE TYPE="/text/css">
P:first-line { font-style: small-caps }
</STYLE>
```

aura pour effet de mettre en petites capitales la première ligne des paragraphes.

Pseudo-élément *first-letter*

Le pseudo-élément *first-letter* est utile pour appliquer un style⁹ particulier au premier caractère d'un texte.

```
BODY P:first-letter { color:purple }
```

En définissant ce pseudo-élément pour des éléments SPAN, il est possible de décrire des effets typographiques plus complexes tels que les *drop-caps*, par exemple.

Pseudo-classe *page box*

La pseudo-classe *page box* est proposée dans [14] en prévision d'extension des CSS pour améliorer l'impression : une page¹⁰ est classifiée automatiquement comme une page gauche (*left*) ou une page droite (*right*). D'autres catégories (*first, ...*) devront être prévues pour compléter la spécification de la maquette des documents.

```
@page : left {
margin-left: 4cm; margin-right: 4 cm; }
```

L'utilisation de ces pseudo-élément et pseudo-classes offre des possibilités typographiques nouvelles qui pourront résoudre un certain nombre de problèmes de présentation. Toutefois, ces extensions sont volontairement limitées afin de ne pas transformer CSS en un complexe langage de description de page.

9. Certaines restrictions décrites dans CSS1 limitent l'application de ces pseudo-éléments typographiques à certaines propriétés (typographiques).

10. Il n'existe pas dans HTML d'élément page. Page est sélectionnée par une règle @ (*at-rule*) définie par le mécanisme d'extension de CSS1.

3.5. Héritage et cascade

Le modèle HTML est particulièrement approprié pour décrire des documents multimédia et hypertextes. HTML dans sa version 3.2 est spécifié par une DTD (*Document Type Definition*) SGML [5] : un document HTML est vu comme un document structuré arborescent. Le corps (BODY) contient des paragraphes (P), des listes (UL ou OL) ou des tables (TABLE). Les listes contiennent des *items* (LI), qui contiennent des paragraphes, etc. Il est possible – et même conseillé – de tirer profit de cette structure pour définir des styles à un niveau tel qu'ils puissent ensuite être appliqués par défaut à tous les descendants dans l'arborescence du document. Ainsi, en définissant des styles pour l'élément BODY, tous ces styles seront considérés comme les styles par défaut de tous les éléments contenus dans l'élément BODY, sauf si certains styles sont redéfinis explicitement pour un élément donné : dans ce cas, c'est ce nouveau style qui s'appliquera à cet élément ainsi qu'à tous ses descendants. Telle est la règle d'héritage.

Par ailleurs, comme on l'a vu dans la section 2, à un même document peuvent être appliquées plusieurs feuilles de style : le style peut être défini dans des feuilles de styles externes ou dans une feuille de style intégrée ou bien défini en ligne dans le corps du document HTML. Les conflits entre ces différentes définitions sont résolus en respectant l'**ordre de cascade** (*Cascading order*) qui est indiqué dans la table 2 : à chaque ligne du tableau correspond une catégorie de critère. Pour déterminer quel est le style qui s'applique, ces critères sont analysés dans l'ordre des lignes. Pour chacun des critères, le style défini en colonne 1, l'emporte sur le style défini en colonne 2, qui l'emporte sur le style défini en colonne 3. En résumé, la méthode de résolution des conflits est la suivante :

- a. En premier lieu, il faut essayer d'appliquer les styles définis pour l'élément ou pour un sélecteur qui convient à cet élément. Si aucune déclaration ne convient, il faut rechercher le style hérité ou, à défaut, la valeur initiale définie par HTML.
- b. L'attribut poids lorsqu'il est utilisé dans une règle de style permet de garantir qu'une règle importante (! *important*) sera prioritaire sur une règle normale.
- c. C'est la feuille de style de l'auteur qui l'emporte sur la feuille de style du lecteur et la feuille de style du lecteur l'emporte sur celle du navigateur.
- d. Lorsqu'un style est spécifié pour un type d'élément et pour un type contextuel, c'est la règle de style considérée comme la plus spécifique (spécifiée pour le type contextuel) qui s'applique.

TABLE 2 – *Les cascades*

Ordre de cascade	Priorité 1	Priorité 2	Priorité 3
a) héritage	style en ligne	sélecteur	style hérité
b) poids	important		normal
c) feuille de style	auteur	lecteur	navigateur
d) spécificité du sélecteur	plus spécifique		plus général
e) ordre de spécification	dernier spécifié		premier spécifié

- e. En dernier ressort, à poids égal, c'est l'ordre dans lequel ces règles sont spécifiées qui est déterminant : la dernière règle spécifiée l'emporte.

4. Conclusion

L'utilisation de feuilles de style pour les documents HTML en est encore à ses débuts : la recommandation des CSS par le consortium W3 ne date que de décembre 1996. Mais de nombreux navigateurs et éditeurs les prennent déjà en compte¹¹. Il ne reste plus qu'à les utiliser efficacement !

Toutefois, la volonté évidente de proposer quelque chose de simple a, pour l'instant, fortement limité les possibilités de mise en page. Espérons qu'une prochaine version soit bientôt définie afin que les expérimentations actuelles puissent converger rapidement et que la notion de feuille de style s'impose. Le *Web* a beaucoup à y gagner : à la fois pour l'esthétique des documents et l'amélioration des performances.

Bibliographie

- [1] Bert Bos, Dave Raggett, Håkon Wium Lie, *Frame-based layout via Style Sheets*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 8 juin 1996.
- [2] T. Bray, C.M. Sperberg-McQueen, *Extensible Markup Language (XML)*, Working Draft, <http://www.w3.org/pub/WWW/TR>, 31 mars 1997.
- [3] Scott Furman *et al.* (Netscape), Scott Isaacs, *et al.* (Microsoft), *Positioning HTML Elements with Cascading Style Sheets*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 31 janvier 1997.

11. Les navigateurs suivants supportent les CSS : Netscape Navigator 4.0 (2/3/97), Microsoft Internet Explorer 3.0 (8/96), Arena W3C testbed browser, Emacs-w3 Gnuscape Navigator. Les éditeurs suivants supportent les CSS : HotDog de Sausage Software (26/3/97), HomeSite de Allaire, HoTMetal Intranet Publisher, QuickSite 2.0, MoTaMaLe, Amaya.

-
- [4] François Chahuneau, « XML : une voie convergente entre SGML et HTML », *Document numérique*, vol. 1, n° 1, p. 69–73, 1997.
- [5] ISO, *Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*, ISO 8879, octobre 1986.
- [6] ISO, *Information processing – Text and Office systems – Document Style Semantics and Specification Language (DSSSL)*, ISO/IEC DIS 10179, 1996.
- [7] Håkon Wium Lie, Bert Bos, *Cascading Style Sheets, level 1*, W3C recommendations, <http://www.w3.org/pub/WWW/TR/REC-CSS1>, 17 décembre 1996.
- [8] Chris Lilley, T. V. Raman, *Aural Cascading Style Sheets (ACSS)*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 28 mars 1997.
- [9] Murray Maloney, *Implementing HTML Frames*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 31 mars 1997.
- [10] Henrik Frystyk Nielsen, *et al. Network Performance Effects of HTTP/1.1, CSS1, and PNG*, W3C Note, 14 février 1997, <http://www.w3.org/PUB/WWW/Protocols/HTTP/Performance/Pipeline.html>.
- [11] Dave Raggett, *HTML3.2 Reference Specification*, W3C Recommendation, <http://www.w3.org/pub/WWW/TR/REC-html32.html>, 14 janvier 1997.
- [12] Dave Raggett, Bert Bos, Håkon Wium Lie, *HTML and Style Sheets*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 24 mars 1997.
- [13] Hélène Richy, Chrystèle Hérault et Jacques André, « Notion de feuille de style », *Cahiers GUTenberg*, n° 21, p. 127–133, juin 1995.
- [14] Robert Strvahn, Stephen Waters, *CSS Printing Extensions*, W3C Working Draft, <http://www.w3.org/pub/WWW/TR>, 31 mars 1997.