

Cahiers **GUT** *enberg*

☞ COMPOSITION EN COULEUR AVEC L^AT_EX

☞ Michel GOOSSENS, Sebastian RAHTZ

Cahiers GUTenberg, n° 16 (1994), p. 5-20.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1994__16_5_0>

© Association GUTenberg, 1994, 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.

Composition en couleur avec L^AT_EX

Michel GOOSSENS

CERN, Division CN, CH-1211 Genève 23, Suisse, <Michel.Goossens@cern.ch>

Sebastian RAHTZ

ArchaeoInformatica, 12 Cygnet Street, York YO2 1AG, Grande Bretagne,
<spqr@ftp.tex.ac.uk>

Résumé. Cet article passe en revue les notions de base de l'utilisation de la couleur avec L^AT_EX. Nous expliquons les principes de la production de la couleur sur les imprimantes. Puis nous présentons l'approche adoptée dans L^AT_EX 2_ε pour spécifier la couleur à l'aide de quelques exemples. Nous terminons par une description de la séparation des couleurs en PostScript en utilisant **dvips**.

Abstract. *In this article we show some basic principles underlying the use of colour. After explaining how colour is reproduced in printers, we show one implementation with L^AT_EX of using it in your work and give a few examples of colour use. We also explain how to make simple colour separations using PostScript and **dvips**.*

1. Modèles de couleur

On sait depuis longtemps que la lumière est composée de trois composantes «primaires», avec lesquelles on peut réaliser toutes les autres couleurs existantes. Sur un tube cathodique, ces couleurs sont le rouge, le vert et le bleu, et un des modèles les plus communs est le modèle RGB (*Red*, *Green* et *Blue* en anglais). Dans les arts graphiques on n'utilise guère ces couleurs primaires mais plutôt leurs compléments, le jaune, le magenta et le cyan. La raison est que les encres «soustraient» leurs couleurs complémentaires de la lumière blanche qui tombe sur la surface illuminée, par exemple l'encre cyan absorbe la composante rouge de la lumière blanche et, en terme du modèle additif des couleurs primaires, le cyan est donc le blanc moins le rouge, ou le bleu plus le vert. De même, le magenta absorbe la composante verte et correspond donc au rouge plus le bleu, et le jaune, qui absorbe le bleu, correspond au rouge plus le vert. Pour donner plus de profondeur ou de relief aux reproductions couleurs, dans l'imprimerie on utilise le procédé de la quadrichromie qui consiste à ajouter une quatrième «couleur», le noir. La composante noire est obtenue en passant l'original de l'image sous

des filtres rouge, vert et bleu, ce qui soustrait une intensité égale des composantes cyan, magenta et jaune. De cette façon on obtient un noir plus absorbant que par la superposition des trois autres encres. Ce modèle des couleurs est appelé le «modèle CMYK»¹.

La figure 1 représente d'une façon simplifiée la relation entre les modèles RGB et CMYK, alors que la figure 2 à la page suivante, contenant les anneaux olympiques et une ellipse avec les couleurs du spectre visible, montre comment en ajoutant les différentes encres une à une on obtient les couleurs finales. On commence par l'encre cyan (en haut à gauche), puis on ajoute respectivement le magenta (en haut à droite), le jaune (en bas à gauche) et finalement le noir (en bas à droite). Le processus est montré dans sa version décomposée dans la partie gauche de la figure, et en montrant l'effet cumulatif de l'application de chaque couleur à droite.

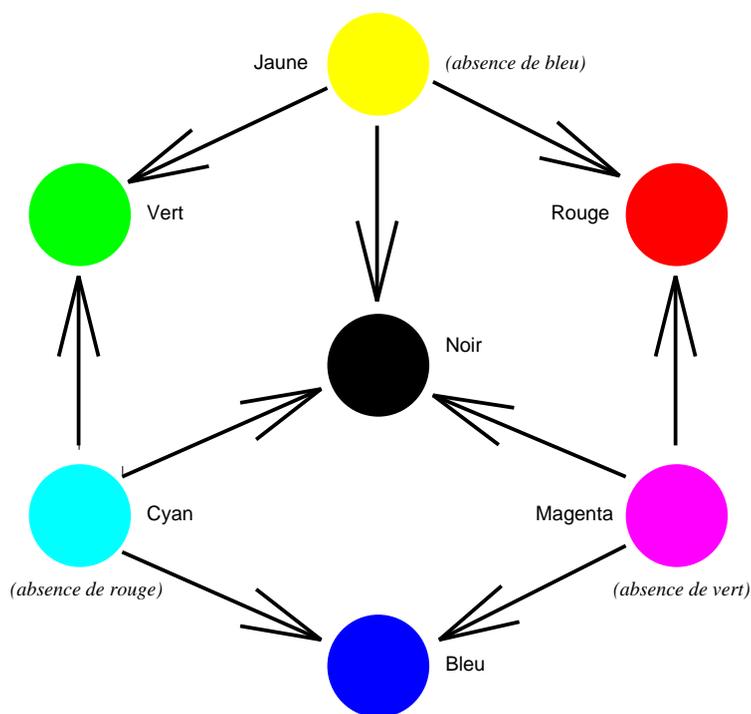
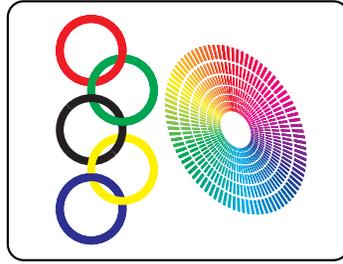
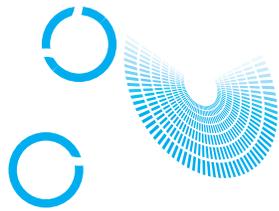
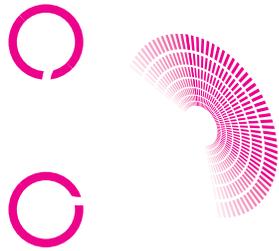
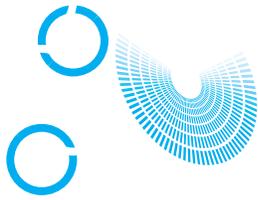
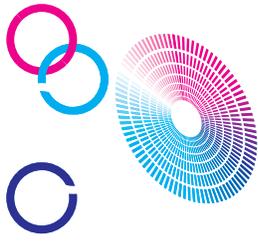


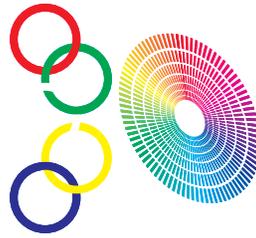
FIG. 1 - Relation entre les modèles RGB et CMYK

Pour piloter leurs imprimantes ou photocomposeuses la plupart des imprimeurs dans le monde utilisent le langage PostScript, qui offre à présent également un bon support de la couleur. En effet, avec la parution du niveau 2 de PostScript en

1. CMYK vient de l'anglais *Cyan, Magenta, Yellow and black*.



October 4, 1995 / 1



October 4, 1995 / 1



FIG. 2 - La séparation des couleurs dans le modèle CMYK

1991 [1], ce langage dispose actuellement de plusieurs modèles supplémentaires pour décrire la couleur, qui sont venus s'ajouter aux modèles RGB et CMYK, discutés ci-dessus. Parmi ceux-ci mentionnons les modèles HSB (de l'anglais *Hue, Saturation, Brightness*, ou teinte, saturation, luminosité) et CIE (le standard *Commission Internationale de l'Éclairage*). En plus, des espaces spéciaux de couleurs peuvent être définis². Il est important de réaliser que seul le PostScript du niveau 2 dispose des commandes nécessaires pour simplifier considérablement l'utilisation de la couleur et la préparation des séparations, qui seront discutées dans la section 3. Notons également que dans l'industrie et les arts graphiques il existe d'autres modèles pour spécifier les couleurs, comme les modèles Munsell et Pantone, ou plus récemment, Focoltone et Trumatch. Dans ces cas on essaie surtout de garantir une harmonie des couleurs entre différents supports possibles.

Utiliser les couleurs avec L^AT_EX 2_ε

Le pilote d'imprimantes **dvips** [7] de Tomas ROKICKI, qui traduit les fichiers DVI en PostScript, est maintenant disponible pour pratiquement tous les ordinateurs et systèmes d'exploitation, allant de VM/CMS sur les gros calculateurs IBM jusqu'à MSDOS sur PC ou Système 7 sur Macintosh. Ce programme autorise l'utilisation de la couleur à travers des commandes `\special` spécifiques. Il y a quelques années, James HAFNER, du Centre de Recherche IBM à Almaden (É.U.d'A.), a développé un style `colordvi` qui utilise ces commandes T_EX de bas niveau pour réaliser une interface utilisateur, qui simplifiait sensiblement la spécification de la couleur dans les documents L^AT_EX. Récemment l'interface pour la couleur de L^AT_EX 2_ε a adopté essentiellement les mêmes principes, et c'est donc la syntaxe de `colordvi` qui sera présentée dans cet article. Dans `colordvi` HAFNER définit 68 couleurs CMYK, qui correspondent aux couleurs des crayons disponibles dans une boîte «Crayola». Ces 68 couleurs sont pré-définies dans les fichiers d'en-têtes de **dvips** et sont chargés avant le code PostScript généré par **dvips** pour représenter le document de l'utilisateur. L'utilisateur peut, évidemment, définir ses propres couleurs en fonction des modèles couleurs RGB, CMYK ou autres. La figure 3 montre les 68 couleurs définies par HAFNER et leurs noms, comme il faut les spécifier, et que nous utiliserons aussi dans nos exemples. La définition d'une couleur doit fixer les intensités des quatre composantes cyan, magenta, jaune et noir, comme des valeurs entre 0 et 1, par exemple la couleur *ForestGreen* (le vert de la forêt) correspond aux valeurs 0.91, 0, 0.88 et 0.12, respectivement.

2. Vous trouverez plus de détails sur les modèles de couleur et les algorithmes de conversion entre les différents modèles dans [1] pour le PostScript, et [2] pour la théorie générale. D'autres livres que nous avons consultés au sujet de PostScript et la couleur sont [6], le plus récent [5] et surtout [4], qui donne beaucoup d'exemples, notamment dans le domaine de l'utilisation de la couleur pour la présentation graphique des données.

| | | | |
|--------------------------------|----------------------------|----------------------------------|------------------------------|
| GreenYellow GreenYellow | Yellow Yellow | Goldenrod Goldenrod | Dandelion Dandelion |
| Apricot Apricot | Peach Peach | Melon Melon | YellowOrange YellowOrange |
| Orange Orange | BurntOrange BurntOrange | Bittersweet Bittersweet | RedOrange RedOrange |
| Mahogany Mahogany | Maroon Maroon | BrickRed BrickRed | Red Red |
| OrangeRed OrangeRed | RubineRed RubineRed | WildStrawberry WildStrawberry | Salmon Salmon |
| CarnationPink CarnationPink | Magenta Magenta | VioletRed VioletRed | Rhodamine Rhodamine |
| Mulberry Mulberry | RedViolet RedViolet | Fuchsia Fuchsia | Lavender Lavender |
| Thistle Thistle | Orchid Orchid | DarkOrchid DarkOrchid | Purple Purple |
| Plum Plum | Violet Violet | RoyalPurple RoyalPurple | BlueViolet BlueViolet |
| Periwinkle Periwinkle | CadetBlue CadetBlue | CornflowerBlue CornflowerBlue | MidnightBlue MidnightBlue |
| NavyBlue NavyBlue | RoyalBlue RoyalBlue | Blue Blue | Cerulean Cerulean |
| Cyan Cyan | ProcessBlue ProcessBlue | SkyBlue SkyBlue | Turquoise Turquoise |
| TealBlue TealBlue | Aquamarine Aquamarine | BlueGreen BlueGreen | Emerald Emerald |
| JungleGreen JungleGreen | SeaGreen SeaGreen | Green Green | ForestGreen ForestGreen |
| PineGreen PineGreen | LimeGreen LimeGreen | YellowGreen YellowGreen | SpringGreen SpringGreen |
| OliveGreen OliveGreen | RawSienna RawSienna | Sepia Sepia | Brown Brown |
| Tan Tan | Gray Gray | Black Black | White |

FIG. 3 - Les couleurs prédéfinies avec L^AT_EX et `colordvi`

$\text{\LaTeX}2_{\epsilon}$ propose plusieurs nouvelles fonctions dans le domaine de la couleur, mais dans ce qui suit nous nous limiterons à la façon de colorier du texte. Par analogie avec les commandes pour changer les polices, $\text{\LaTeX}2_{\epsilon}$ donne le choix entre deux types de commandes pour spécifier la couleur. La première a la forme d'une commande :

```
\textcolor{<nom_de_la_couleur>}{<texte>}
```

Elle prend deux arguments, le premier spécifiant le nom de la couleur à utiliser pour composer le texte contenu dans le deuxième argument. Cette commande sera utilisée pour des modifications de couleur locales (limitées) ou imbriquées. Au terme de cette commande on retrouve l'état précédent de la couleur, par exemple :

| | |
|--|--|
| <p>Ce texte est en noir. Ici le texte est en bleu et revoici du texte en noir.</p> | <pre> Ce texte est en noir. \textcolor{Blue}{Ici le texte est en bleu} et revoici du texte en noir. </pre> |
|--|--|

Le deuxième type de commande pour définir la couleur a la forme :

```
\color{<nom_de_la_couleur>}
```

Ici nous avons affaire à une modification globale de la couleur qui spécifie la couleur du document à partir du point où la commande est donnée. L'état de la couleur n'est pas sauvegardé, par exemple :

| | |
|--|--|
| <p>Tout le texte qui suit est en rouge. Maintenant le texte est en noir.</p> | <pre> \color{Red}Tout le texte qui suit est en rouge. \color{Black}Maintenant le texte est en noir. </pre> |
|--|--|

Ce type de commande respecte néanmoins les blocs \TeX^3 comme le montre l'exemple suivant :

| | |
|---|---|
| <p>Nous commençons en noir, mais maintenant tout le texte est en rouge, quoique ceci devrait être en bleu et ceci à nouveau en rouge. Pour finir en noir.</p> | <pre> Nous commençons en noir, mais maintenant \bgroup \color{Red} tout le texte est en rouge, {\color{Blue}quoique ceci devrait être en bleu} et ceci à nouveau en rouge. \egroup Pour finir en noir. </pre> |
|---|---|

3. Notons cependant que ceci n'est pas un point trivial, parce que \TeX n'a aucun support interne pour les attributs couleurs. Ceci peut avoir des conséquences inattendues pour les couleurs obtenues dans un bloc qui enjambe plusieurs pages, qui se trouve dans un élément flottant, une note de bas de page, etc.

2. Couleur, élément de communication

La présence de la couleur dans un document constitue une valeur ajoutée. Elle accroche l'attention du lecteur et modifie la perception du message, en y introduisant une dimension affective. La discussion de la section précédente montre qu'il est relativement facile de colorier un morceau de texte. Mais pour tirer vraiment profit de la couleur comme outil de communication, il faut pouvoir appliquer des couleurs de fond, pour mettre en évidence certaines parties de la page. $\text{\LaTeX} 2_{\epsilon}$ offre certaines commandes de base pour réaliser des fonds coloriés et dégradés, mais il est préférable d'utiliser l'extension **PSTricks** [9] de Timothy Van ZANDT, qui offre une panoplie de commandes qui facilitent la mise en œuvre de la couleur (voir l'article de Denis GIROU dans le présent numéro). Voici un exemple simple mettant en évidence une partie d'un texte :

Attention !

```
\psframebox*[fillcolor=YellowGreen]%
{\textbf{Attention !}}
```

Il est souvent nécessaire de combiner du texte et un fond coloriés dans un tableau. Contre toute attente, il est extrêmement difficile de réaliser une telle présentation avec \LaTeX . À nouveau, c'est une extension de Timothy Van ZANDT, **ColorTab**, qui permet de composer de tels tableaux sans trop de problèmes. Comme la mise en œuvre de ce style est d'une certaine complexité, nous nous limiterons dans ce qui suit à deux exemples, et renvoyons le lecteur intéressé à la documentation [8]. Notons que **ColorTab** fonctionne avec *plain* \TeX , et les environnements **tabular** standard de \LaTeX et **longtable** de David CARLISLE.

Notre premier exemple, tableau 1, est inspiré d'une brochure touristique et montre les principes du balisage à introduire.

Le tableau 2 à la page 13 est un exemple plus complexe (une description des commandes HTML) où il est montré comment utiliser des couleurs pour mettre en évidence d'une façon cohérente des éléments structurels similaires d'un tableau⁴. La couleur guide le lecteur à travers l'information présentée et lui permet de suivre des liens visuels qui lient les différents éléments. Aussi ce tableau est-il un exemple d'une des règles d'or de la composition en couleurs, c'est-à-dire de ne jamais utiliser plus de trois ou quatre couleurs sur une page de texte, autrement la codification (la signification) des différentes couleurs se perd.

Dans notre tableau, nous avons utilisé du jaune pour les titres, du rouge pour les commandes HTML, du vert pour les textes que nous voulons mettre en évidence,

4. Ce tableau est réalisé à l'aide de l'extension **longtable** de David CARLISLE (voir par exemple [3]).

TAB. 1 - Exemple de tableau colorié

| ORLANDO | | | | | MCO |
|---------|------|----------|-------------------|----------------|--------|
| Genève | OG4G | Je 21/10 | Lu 01/11 ou 08/11 | 11/11 ou 18/11 | 799CHF |
| | | Je 28/10 | Lu 01/11 ou 08/11 | 04/11 ou 11/11 | 799CHF |
| | OG7A | Di 24/10 | Lu 01/11 ou 08/11 | 08/11 ou 15/11 | 799CHF |
| | | Di 31/10 | Lu 08/11 | 08/11 | 799CHF |

```

\setlength{\extrarowheight}{1mm}\setlength{\tabcolsep}{2mm}
\newsavebox{\Orlando}\newlength{\OrlandoWidth}
\savebox{\Orlando}{% Sauvegarder le tableau dans une boîte
\begin{tabular}{|l|l|l|l|c|>{\bfseries}c<{\small CHF}}
\SP\hline
\LCC
&\color{Yellow}&          & \color{Yellow}&          & \color{Yellow}\&
Genève & OG4G & Je 21/10&Lu 01/11 ou 08/11&11/11 ou 18/11& 799\&
&      & Je 28/10&Lu 01/11 ou 08/11&04/11 ou 11/11& 799\&
& OG7A & Di 24/10&Lu 01/11 ou 08/11&08/11 ou 15/11& 799\&
&      & Di 31/10&Lu 08/11          & 08/11          & 799\&
\ECC
\RP\hline
\end{tabular}}
% Ajouter cadre rouge entourant le texte en blanc (bord à 2 mm)
\setlength{\OrlandoWidth}{\usebox{\Orlando}}% largeur tableau
\addtolength{\OrlandoWidth}{-4mm}% soustraire 2 X 2mm pour bord
\psframebox*[fillcolor=Red,framesep=2mm]{\textcolor{White}%
{\makebox{\OrlandoWidth}{\textsf{ORLANDO} \hfill \textsc{mco}}}}
\usebox{\Orlando}

```

et du bleu pour les exemples. Le début du code pour la réalisation de ce tableau est montré ci-dessous. Une excellente discussion de l'utilisation de la couleur dans la préparation de documents informatiques se trouve dans [11].

```

\newcommand{\RLit}[1]{\texttt{\color{Red}{#1}}} % rouge (balises)
\newcommand{\BLit}[1]{\texttt{\color{Blue}{#1}}} % bleu (exemples)
\newcommand{\Attr}{\color{Green}{Attributes:}} % vert
% "strut" (filet invisible) contrôlant l'espacement dans les titres
\newcommand{\RR}{\rule[-.35em]{0pt}{1.2em}}
% Constructions des titres en jaune
\newcommand{\MULT}[1]{\LColors\color{Yellow}\&          % fond jaune
\hfil\large\bf\RR#1\hfil\\\hline% texte
\LColors\color{White}\&} % fond blanc
\setlength{\arraycolsep}{2pt}
\begin{longtable}{|!{\LCi}p{.97\linewidth}}|% début de longtable

```

TAB. 2 - Exemple d'utilisation de PStricks, ColorTab et longtable

| Overview of HTML commands | |
|---|--|
| General structure of an HTML tag | |
| <code><tag_name></code> e.g. <code><p></code> | |
| <code><tag_name></code> text <code></tag_name></code> e.g. <code><title></code> <code></title></code> | |
| <code><tag_name attr1=arg1 attr2=arg2></code> text <code></tag_name></code> e.g. <code><pre width=60></code> <code></pre></code> | |
| General structure of an HTML document | |
| <code><html></code> | |
| <code><head></code> <code></head></code> | |
| <code><body></code> <code></body></code> | |
| <code></html></code> | |
| Elements used inside the head | |
| <code><link></code> Specify index file | |
| <code><title></code> <code></title></code> Specify document title | |
| <code><meta id=...></code> Set a variable value. Attributes: variable name | |
| <code><link></code> Relationships to other documents. Attributes: see "Anchors" below | |
| <code><base></code> Name of the file in which the current document is stored | |
| Elements used inside the body | |
| Text Elements | |
| <code>&</code> Indicates end of paragraph | |
| <code><p></code> <code></p></code> Identifies text that has already been formatted (preformatted) by some other system and must be displayed as is. Preformatted text may include embedded tags, but not all tag types are permitted. | |
| Attributes: <code>width</code> | |
| Headers | |
| <code><h1></code> <code></h1></code> First level header | |
| <code><h2></code> <code></h2></code> Second level header | |
| <code><h3></code> <code></h3></code> Third level header | |
| <code><h4></code> <code></h4></code> Fourth level header | |
| <code><h5></code> <code></h5></code> Fifth level header | |
| <code><h6></code> <code></h6></code> Sixth level header | |
| Logical styles | |
| <code></code> <code></code> Emphasis | |
| <code><code></code> <code></code></code> Display code example | |
| <code><math></code> <code></math></code> Include simple output | |
| <code><input type="checkbox"></code> <code></input type="checkbox"></code> Display keyboard input | |
| <code><var></code> <code></var></code> Define a variable | |
| <code><dfn></code> <code></dfn></code> Display a definition | |
| <code><cite></code> <code></cite></code> Display a citation | |
| Physical styles | |
| <code></code> <code></code> Bold font | |
| <code><i></code> <code></i></code> Italics | |
| <code><u></code> <code></u></code> Underline | |
| <code><tt></code> <code></tt></code> Typewriter font | |
| Definition list | |
| <code><dl></code> or <code><dl compact></code> Beginning of definition list | |
| <code><dt></code> term <code><dd></code> definition | |
| <code><dt></code> term <code><dd></code> definition | |
| <code></dl></code> End of definition list | |
| Unordered list | |
| <code></code> Beginning of unordered list | |
| <code></code> First item in list | |
| <code></code> Next item in list | |
| <code></code> End of unordered list | |
| Ordered list | |
| <code></code> Beginning of ordered list | |
| <code></code> First item in list | |
| <code></code> Next item in list | |
| <code></code> End of ordered list | |
| Hyperlinks and Anchors | |
| <code></code> <code></code> Define a target location in a document | |
| <code></code> <code></code> Link to a location in the same file | |
| <code></code> <code></code> Link to another file | |
| Attributes: one of <code>name</code> or <code>href</code> are required, <code>rel</code> , <code>rev</code> , <code>urn</code> , <code>title</code> , <code>methods</code> are optional | |
| Structure of a Universal Resource Location (URL): <code>resource_type://host.domain:port/path/name</code> | |
| Entities | |
| <code>&keyord;</code> Display character identified by keyword, e.g. <code>&amp;</code> ; (for <code>&</code>) or <code>&l1;</code> ; (for <code><</code>) ¹ | |
| <code><!-- text --></code> Place a comment in the HTML source | |
| <code><address></code> <code></address></code> Address information | |
| Images | |
| <code></code> <code></code> Include graphics image ² | |

¹ A full list of entities is on WWW in `http://info.cern.ch/ypertext/WWW/MarkUp/entities.html`.
² Supported by some servers only, e.g. Mosaic.

```

\caption*{\textbf{\LARGE Overview of HTML commands}}      \\ \hline
\endfirsthead
                                                            \hline
\endhead
                                                            \hline
\endfoot
\MULT{\Large\RR General structure of an HTML tag}
\RLit{<tag\_name>} e.g. \BLit{<P>}                      \\
\RLit{<tag\_name> \textcolor{Black}{text} </tag\_name>} e.g.
  \BLit{<title> .... </title>}                          \\
\RLit{<tag\_name attribute\_name=argument>\textcolor{Black}{text}
</tag\_name>} e.g. \BLit{<pre width=60> .... </pre>}    \\ \hline
\MULT{\Large\RR General structure of an HTML document}
\RLit{<html>}                                           \\
\quad\RLit{<head> . . . </head>}                        \\
\quad\RLit{<body> . . . </body>}                       \\
\RLit{</html>}                                          \\ \hline

```

Un autre domaine d'application où l'utilisation de couleurs est très répandu est la préparation de transparents. Ici nous conseillons d'utiliser la classe de documents **Seminar** [10], également développée par Timothy Van ZANDT ; nous la décrivons dans notre autre article dans ces *Cahiers*. La figure 4 de la page suivante montre le même texte pour un transparent reproduit en utilisant différentes couleurs pour le fond et le texte. Il faut faire attention de choisir une bonne combinaison pour transmettre votre message efficacement. En particulier, certains contrastes plaisants sur une page imprimée dans un livre (où la lumière agit en réflexion) peuvent être inopérants, voire illisibles sur un transparent. La seule façon garantie de réussir est de projeter quelques transparents qui utilisent les combinaisons choisies et d'essayer de déchiffrer le message à transmettre.

3. Séparation des couleurs avec **dvips**

Supposons que nous ayons composé un document qui contient de la couleur avec \LaTeX et traduit le fichier **dvi** obtenu en PostScript avec **dvips**. Alors nous pouvons visionner le fichier sur un écran ou imprimer des copies individuelles sur une imprimante PostScript couleurs. Mais que devons nous faire si nous voulons reproduire notre document (livre) en beaucoup d'exemplaires en l'imprimant en offset ? Cette procédure d'impression nécessite quatre versions (clichés) pour chaque page, où des niveaux de gris indiquent les intensités du cyan, magenta, jaune et noir sur la page. Ces clichés utilisent un film spécial d'une grande rigidité, parce que chaque page est imprimée quatre fois, et les tolérances pour le film doivent permettre un recouvrement parfait pour les différentes couleurs.



FIG. 4 - Quelques exemples de l'utilisation de la couleur avec Seminar

Certains systèmes de reproduction sont capables de produire les quatre séparations automatiquement, mais assez souvent elles sont obtenues par des actions au niveau de PostScript. Pour garantir un résultat parfait, les professionnels de la branche graphique disposent de plusieurs outils sophistiqués, dont la discussion dépasse le cadre de cet article. Néanmoins, les utilisateurs de \TeX peuvent générer des séparations CMYK simples de leurs fichiers couleurs à l'aide de **dvips**. Notons que la procédure décrite ci-dessous n'utilise que des opérateurs de PostScript de niveau 1 et ne nécessite donc nullement de disposer d'une imprimante PostScript de niveau 2.

Le principe de la séparation des couleurs par **dvips** est que chaque page est produite quatre fois (en utilisant le paramètre **-b 4** avec la commande **dvips** ou en introduisant une ligne contenant **b 4** dans le fichier de configuration) et que les opérateurs couleurs sont redéfinis différemment pour ces quatre pages en incluant un fichier d'en-tête adéquat. Un tel fichier, distribué avec **dvips**, a été développé et est maintenu par Sebastian RAHTZ. La procédure est en grande partie dérivée de [4] et [6] et utilise la procédure utilisateur **bop-hook** de **dvips** pour incrémenter un compteur au début de chaque page, ce qui permet de savoir si l'on est en train de produire une page C, M, Y ou K. L'opérateur **setcmykcolor** est alors redéfini pour produire seulement une des quatre composantes de la couleur en niveaux de gris, en convertissant les couleurs RGB en CMYK avant de passer par la procédure évoquée. L'opérateur **setgray** est désactivé sur les pages «C», «M» et «Y» et influence seulement la page noire («K»). Le code PostScript est donné *in extenso* dans l'annexe.

Le tableau suivant montre la sortie après séparation pour l'exemple contenant **Attention!**, où la boîte est composée en *ForestGreen*, qui correspond à des valeurs CMYK «0.91 0 0.88 0.12». Les quatre cases représentent les quatre «clichés» noir et blanc qui donnent les intensités de l'image dans chacune des quatre couleurs de base. Notons le fait que le cliché «M» sera blanc, parce que ni la boîte verte, ni le texte noir n'ont une composante magenta.

TAB. 3 - Les quatre «clichés» noir et blanc avec les composantes CMYK séparées

| | | | |
|-------------------|-------------|-------------------|-------------------|
| cyan (C) | magenta (M) | jaune (Y) | noir (K) |
| Attention! | | Attention! | Attention! |

Références bibliographiques

- [1] Adobe Systems Incorporated. *Manuel de référence du langage PostScript (deuxième édition)*, Addison-Wesley France SA, 1992.
- [2] James D. FOLEY, Andries van DAM, Steven K. FEINER, et John F. HUGHES. *Computer Graphics, Principles and Practice (deuxième édition)*, Addison-Wesley, Reading, MA, USA, 1990.
- [3] Michel GOOSSENS, Frank MITTELBACH et Alexander SAMARIN. *The L^AT_EX Companion*. Addison-Wesley, Reading, MA, USA, 1994.
- [4] Gerard KUNKEL. *Graphic Design with PostScript*. Scott, Foresman and Company, Glenview, Illinois, 1990.
- [5] Henry Mc GILTON et Mary CAMPIONE. *PostScript by Example*. Addison-Wesley, Reading, MA, USA, 1992.
- [6] Glenn C. REID. *PostScript Language Program Design*. Addison-Wesley, Reading, MA, USA, 1988.
- [7] Tomas ROKICKI. *DVIPS: A T_EX Driver*. Manuel d'utilisation distribué électroniquement avec le logiciel. Janvier 1993.
- [8] Timothy Van ZANDT. *colortab.tex: Shading and coloring T_EX tables. Version 0.9*. Manuel d'utilisation distribué électroniquement avec le logiciel. Janvier 1993.
- [9] Timothy Van ZANDT. *PSTricks: PostScript macros for Generic T_EX. User's Guide. Version 0.93*. Manuel d'utilisation distribué électroniquement avec le logiciel. Mars 1993.
- [10] Timothy Van ZANDT. *seminar.sty: A L^AT_EX style for slides and notes. User's Guide. Version 1.0*. Manuel d'utilisation distribué électroniquement avec le logiciel. Avril 1993.
- [11] Jan V. WHITE. *Color for the Electronic Age*. Watson-Guptill Publications, New-York, 1990.

Annexe : le fichier d'en-tête **dvips** pour la séparation des couleurs

```
%!  
% Séparation des couleurs par dvips utilisant la procédure bop-hook.  
%  
% Basé sur le << Green Book >> (Listing 9-5, à la page 153)  
% et sur Kunkel << Graphic Design with PostScript >>  
%  
% Version 1.2 par Sebastian Rahtz 7 Jan. 1994  
%  
/seppages 0 def  
userdict begin  
/Min {% 3 éléments sur la pile  
2 copy lt { pop }{ exch pop } ifelse  
2 copy lt { pop }{ exch pop } ifelse  
} def  
/SetGray {  
1 exch sub systemdict begin adjustdot setgray end  
} def  
/sethsbcolor {systemdict begin  
sethsbcolor currentrgbcolor end  
userdict begin setrgbcolor end}def  
  
/ToCMYK  
% Manuel de référence PostScript (Livre Rouge pp. 315-18)  
{  
% soustraire la composante de couleur de 1  
3 { 1 exch sub 3 1 roll } repeat  
% calculer la composante noire dans les composantes C, M et Y  
% trouver le minimum (k)  
3 copy Min  
% déterminer la composante commune  
blackUCR sub  
dup 0 lt {pop 0} if  
/percent_UCR exch def  
%  
% suppression des couleurs sous-jacentes  
%  
3 { percent_UCR sub 3 1 roll } repeat  
% maintenant nous calculons le noir  
percent_UCR 1.25 mul % 1 exch sub  
% la pile devrait maintenant contenir les composantes C M Y K  
} def  
%  
% définition de la procédure pour les points de repère  
%  
/cX 18 def  
/CM{gsave TR 0 cX neg moveto 0 cX lineto stroke  
cX neg 0 moveto cX 0 lineto stroke grestore}def
```

```

%
/bop-hook{cX dup TR
%
% quelle est la couleur que nous traitons ?
%
seppages 1 add
/seppages exch def
seppages 5 eq { /seppages 1 def } if
seppages 1 eq {
/ColourName (CYAN) def
CYAN setupcolor
/WhichColour 3 def } if
seppages 2 eq {
/ColourName (MAGENTA) def
MAGENTA setupcolor
/WhichColour 2 def } if
seppages 3 eq {
/ColourName (YELLOW) def
YELLOW setupcolor
/WhichColour 1 def } if
seppages 4 eq {
/ColourName (BLACK) def
BLACK setupcolor
/WhichColour 0 def } if
%
% génération des points de repère
%
gsave .3 setlinewidth
3 -7 moveto
/Helvetica findfont 6 scalefont setfont
ColourName show
0 0 CM
vsize cX 2 mul sub dup hsize cX 2 mul sub dup isls
{4 2 roll}if 0 CM
exch CM 0
exch CM
grestore 0 cX -2 mul TR isls
{cX -2 mul 0 TR}if
} def end
%
/separations 48 dict def
separations begin
/cmykprocs [ %def
% cyan
{ pop pop pop SetGray }
% magenta
{ pop pop exch pop SetGray }
% jaune
{ pop 3 1 roll pop pop SetGray }

```

```
% noir
{ 4 1 roll pop pop pop SetGray }
] def
/rgbprocs [ %def
% cyan
{ ToCMYK pop pop pop SetGray }
% magenta
{ ToCMYK pop pop exch pop SetGray }
% jaune
{ ToCMYK pop 3 1 roll pop pop SetGray }
% noir
{ ToCMYK 4 1 roll pop pop pop SetGray }
] def
/screenangles [ 105 75 0 45 ] def
%
% cyan magenta jaune noir
end % séparations

% setupcolor accepte 0, 1, 2, ou 3 comme argument,
% pour le cyan, le magenta, le jaune ou le noir.
/CYAN 0 def /MAGENTA 1 def
/YELLOW 2 def /BLACK 3 def
/setupcolor{ %def
userdict begin
dup separations /cmykprocs get exch get
/setcmykcolor exch def
dup separations /rgbprocs get exch get
/setrgbcolor exch def
separations /screenangles get exch get
currentscreen
exch pop 3 -1 roll exch
setscreen
/setscreen { pop pop pop } def
%
% redéfinition de la commande setgray pour qu'elle affecte
% uniquement la séparation noire.
%
/setgray {
WhichColour 0 eq
{systemdict begin adjustdot setgray end}
{pop systemdict begin 1 setgray end}
ifelse}def
end
} bind def
%
% de Kunkel
%
/adjustdot {dup 0 eq { } {dup 1 exch sub .1 mul add} ifelse} def
```