

Cahiers **GUT** *enberg*

☞ L^AT_EX → HTML ALLER ET RETOUR
☞ Michel GOOSSENS

Cahiers GUTenberg, n° 19 (1995), p. 98-120.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1995__19_98_0>

© Association GUTenberg, 1995, 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.

L^AT_EX – HTML aller et retour*

Michel GOOSSENS

CERN, Division CN, 1211 Genève 23, Suisse,
E-mail: Michel.Goossens@cern.ch

Résumé. L^AT_EX et HTML sont deux des langages qui permettent d'exprimer la fonction des différents éléments structurels d'un document. Nous mettons en évidence le parallèle entre ces deux systèmes. Nous étudions en profondeur l'utilitaire **LaTeX2HTML**, développé par Nikos Drakos, qui permet une traduction quasi-automatique d'un document L^AT_EX en HTML. Nous terminons pas une brève discussion de **HTML2LaTeX** et de **SGML2TeX** qui traduisent un document HTML en L^AT_EX.

Abstract. *Both L^AT_EX and HTML are languages that can express the function of the structural elements of a document, and similarities between these two systems are shown. A detailed study of the LaTeX2HTML program, written by Nikos Drakos, is proposed. LaTeX2HTML allows a quasi-automatic translation of L^AT_EX documents into HTML. A brief discussion of the HTML2LaTeX and SGML2TeX programs that translate HTML into L^AT_EX concludes the article.*

1. L^AT_EX et HTML : similitudes

Le tableau 1 compare les éléments de balisage générique en HTML (niveau 1) et en L^AT_EX. Pour la plupart des entrées, les différences entre les commandes sont triviales. Ceci suggère qu'une conversion de l'un à l'autre pour les éléments en question ne pose pas de problèmes.

Les programmes décrits dans ce qui suit sont basés sur ces similitudes mais, afin d'exploiter les richesses du langage L^AT_EX par rapport à HTML actuel, il faut augmenter la panoplie des commandes traitées correctement en utilisant des techniques *ad hoc* (comme la transformation d'une partie du source L^AT_EX en images dans le cas de **LaTeX2HTML**) ou en suivant les indications explicites de l'utilisateur.

*. Présentation réalisée à Nanterre, le 19 janvier 1995 lors de la journée *diffusion des documents électroniques*.

Description	HTML	\LaTeX
Commandes de sectionnement		
niveau 1	<H1>	<code>\chapter</code> ou <code>\section</code>
niveau 2	<H2>	<code>\section</code> ou <code>\subsection</code>
niveau 3	<H3>	<code>\subsection</code> ou <code>\subsubsection</code>
niveau 4	<H4>	<code>\subsubsection</code> ou <code>\paragraph</code>
niveau 5	<H5>	<code>\paragraph</code> ou <code>\subparagraph</code>
niveau 6	<H5>	<code>\subparagraph</code>
nouveau paragraphe	<P>	<code>\par</code>
Listes		
liste numérotée		<code>\begin{enumerate}</code>
liste non-numérotée		<code>\begin{itemize}</code>
entrée dans la liste		<code>\item</code>
liste de description	<DL>	<code>\begin{description}</code>
terme	<DT>	<code>\item</code>
définition	<DD>	<i>text</i>
Mise en valeur du texte		
emphase	texte	<code>\emph{texte}</code>
italique	<I>texte</I>	<code>\textit{texte}</code>
gras	texte	<code>\textbf{texte}</code>
chasse fixe	<TT>texte</TT>	<code>\texttt{texte}</code>

TABLE 1 - Comparaison des éléments structurels de HTML et \LaTeX

2. Conversion de \LaTeX vers HTML

Avant d'aborder \LaTeX 2HTML de Nikos Drakos, mentionnons un autre programme, `12x`¹ écrit par Henning Schulzrinne (Berlin, Allemagne), qui traduit \LaTeX en plusieurs autres formats. C'est un programme d'analyse écrit en C qui appelle une fonction Tcl[1] pour chaque fonction \LaTeX . Le convertisseur traducteur `html.tcl` est disponible pour traduire un fichier \LaTeX en HTML, par exemple en entrant :

```
12x -p html.tcl article.tex
```

Actuellement, seul un sous-ensemble des commandes \LaTeX est traité (pas d'expressions mathématiques, de tableaux, de verbatim, etc.), mais il est relativement aisé d'augmenter le code du convertisseur `html.tcl` en y ajoutant de nouvelles fonctions Tcl.

Notons également le programme `tex2RTF`² de Julian Smart (Edinburgh, Grande Bretagne) qui peut transformer \LaTeX en HTML. Il marche relativement

1. <http://info.cern.ch/hypertext/WWW/Tools/12x.html>

2. <http://www.aiai.ed.ac.uk/~jacs/tex2rtf.html>

bien pour un sous-ensemble des commandes \LaTeX mais ne traite pas les tableaux, les expressions mathématiques, les tabulations, etc.

Lionel Cons (CERN, Genève) a développé le programme `texihtml`³ qui traduit les sources `texinfo` (le langage \TeX utilisé pour les documentations de GNU) en HTML.

2.1. Convertiseur \LaTeX2HTML : généralités

Le programme `LaTeX2HTML`⁴ écrit en `perl` par Nikos Drakos transforme un document \LaTeX en un ensemble de fichiers HTML interconnectés reflétant la structure du document original et permettant une navigation facile. Les références croisées, renvois bibliographiques, notes de bas de page, tables des matières, listes des figures et tableaux et l'index sont transformés en « hyperliens » (ces liens connectent différentes parties d'une structure hypertexte). Les commandes \LaTeX qui ont un équivalent (approximatif) en HTML (cf. la Table 1) sont converties en utilisant les balises HTML adéquates. Les éléments visuels \LaTeX plus complexes (tableaux, formules mathématiques, environnements graphiques) ou ceux pour lesquels aucune traduction n'est disponible dans le code `perl` de base sont transformés en images qui sont placées automatiquement à l'endroit qui convient dans le document HTML final.

De plus, pour augmenter les capacités hypermédia du système, `LaTeX2HTML` offre la possibilité de définir des hyperliens et renvois symboliques arbitraires entre plusieurs documents et permet l'utilisation de code HTML pur dans la source \LaTeX .

2.1.1. Liste des fonctions

Parmi l'ensemble de ses fonctionnalités `LaTeX2HTML` permet :

- le traitement « convenable » des commandes \LaTeX standard ;
- la division du document en une ou plusieurs composantes (sous-documents) en suivant les instructions de l'utilisateur ;
- l'inclusion à chaque page de pictogrammes (icônes) de navigation contenant des hyperliens avec d'autres parties du document ;
- le traitement correct des formules mathématiques, des tableaux, des figures ou de tout environnement arbitraire en faisant appel à \LaTeX qui sauve sa sortie sous forme d'images ;
- la gestion d'un facteur de mise à échelle arbitraire devant être appliqué aux figures et tableaux qui peuvent être ainsi inclus dans le document HTML sous forme réduite ou comme « croquis miniature » (en anglais : *thumbnail*) ;

3. <http://asis01.cern.ch/infohtml/texi2html.html>

4. <http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html>

- le support pour les programmes de visualisation orientés images (p. ex. **Mosaic**) ou texte simple (p. ex. **Lynx**) ;
- la définition de nouvelles commandes ou environnements, même s'ils sont placés dans un fichier d'extension externe ;
- la génération de notes de bas de page, d'une table des matières, d'une liste des figures ou des tableaux, d'une bibliographie et d'un index ;
- la traduction des renvois L^AT_EX en hyperliens qui autorisent non seulement des références aux éléments à l'intérieur d'un même document, mais également à d'autres documents disponibles localement ou sur un site distant ;
- la traduction, si possible, des caractères spéciaux L^AT_EX en entités ISO-Latin1 ;
- l'extension du langage standard L^AT_EX avec des commandes pour introduire des hyperliens pointant vers des ressources multimédia résidant sur Internet ou des programmes qui doivent être exécutés ;
- la possibilité d'ignorer du texte si une certaine condition est vérifiée ;
- l'utilisation de code HTML « pur » dans le document L^AT_EX.

2.1.2. Options de la commande `latex2html`

Le programme **LaTeX2HTML** peut être piloté à l'aide d'options spécifiées dans la ligne de commande. La syntaxe générale de la commande est :

```
> latex2html [options] nom_de_fichier.tex
```

Les options sont les suivantes :

- split num**
niveau jusqu'auquel on veut séparer le document (0 signifie que toute la sortie doit être dans un même fichier ; la valeur par défaut est 8) ;
- link num**
niveau jusqu'auquel on doit montrer l'arborescence dans la liste des matières (0/1/2/3/4 pour /partie/chapitre/section/sous-section/sous-sous-section ; la valeur par défaut est 4) ;
- nolatex**
ne pas appeler L^AT_EX pour les environnements inconnus, ce qui permet un traitement plus rapide (toutefois sans figures, mathématiques et tableaux) ;
- external_images**
introduire un hyperlien vers une image externe pour chaque image dans le source (par défaut les images sont incluses à l'intérieur du fichier) ;

- ascii_mode**
utiliser seulement des caractères Ascii et ne pas inclure d'images dans le fichier sortie, ce qui permet son utilisation avec des visualisateurs qui ne peuvent traiter la balise ;
- t texte-du-titre**
titre du document ;
- dir répertoire-de-sortie**
déposer la sortie dans le répertoire spécifié ;
- address adresse-auteur**
signature à utiliser dans chaque page ;
- no_navigation**
ne pas introduire les aides à la navigation dans chaque page ;
- top_navigation**
introduire les aides à la navigation en haut de chaque page ;
- bottom_navigation**
introduire les aides à la navigation en bas de chaque page ;
- auto_navigation**
introduire les aides à la navigation en haut de chaque page et, si la page contient plus que \$WORDS_IN_PAGES lignes (valeur par défaut 300), aussi en bas de la page en question ;
- index_in_navigation**
introduire un lien vers l'index dans le tableau de navigation ;
- contents_in_navigation**
introduire un lien vers la table des matières dans le tableau de navigation ;
- next_page_in_navigation**
introduire un lien vers la page suivante dans le tableau de navigation ;
- previous_page_in_navigation**
introduire un lien vers la page précédente dans le tableau de navigation ;
- info desc**
introduire le texte *desc* dans la section *About this document ...* qui contient de l'information sur le document traduit ;
- dont_include fichier(s)**
ne pas inclure le(s) fichier(s) indiqué(s) ; habituellement cette option est utilisée pour exclure certaines extensions contenant des commandes L^AT_EX que L^AT_EX2HTML ne peut pas traiter ;
- reuse**
réutiliser les images créées lors d'un passage précédent ;
- init_file fichier**
lire le fichier *fichier*, qui peut contenir des commandes perl qui changent les commandes ou valeurs par défaut ;
- show_section_numbers**
numéroter les sections (par défaut les sections ne sont pas numérotées pour faciliter l'emploi des document de façon autonome) ;

-h

afficher la liste des options.

2.2. Exemple d'utilisation de LaTeX2HTML

Pour montrer la procédure à appliquer pour générer un document HTML à partir d'un document L^AT_EX, commençons par un exemple simple, c.-à-d. le fichier montré en figure 1. En le traitant avec L^AT_EX on obtient la sortie de la figure 2.

Ce même document L^AT_EX sera maintenant traité par LaTeX2HTML avec la commande

```
>latex2html nom_du_fichier.tex
```

où les options par défaut ont été utilisées. Les résultats sont montrés dans la figure 3. Le document principal est montré au milieu en haut. Des flèches numérotées indiquent les documents secondaires qui sont créés automatiquement. Le document qui contient la table des matières n'est pas montré explicitement puisque son contenu correspond à peu près au document principal. Notons le tableau de navigation en haut de chaque « page », qui correspond à l'option `-top_navigation` choisie par défaut. Le tableau contient cinq boutons-poussoir : **Next** pour aller au document *suivant* (option `-next_page_in_navigation` choisie par défaut), **Up** pour *remonter* d'un niveau, **Previous** pour aller au document *précédent* (option `-previous_page_in_navigation` choisie par défaut), **Contents** qui permet d'aller directement à la *table des matières* (option `-contents_in_navigation` choisie par défaut) et finalement **Index** pour aller à l'index (option `-index_in_navigation` choisie par défaut). Ces valeurs par défaut peuvent être modifiées en redéfinissant des variables `perl` dans le fichier `.latex2html.init`, qui sera décrit dans la section 2.3.

- ❶ la liste des figures, qui contient un hyperlien vers le document ❸ qui contient la figure en question ;
- ❷ la liste des tableaux, qui contient un hyperlien vers le document ❹ qui contient le tableau en question ;
- ❸ la première section, qui contient un peu de texte, une figure dans le texte, et un renvoi ([1]) à la bibliographie (document ❺) ;
- ❹ la deuxième section, qui contient un peu de texte et un tableau ;
- ❺ les références bibliographiques ;
- ❻ l'index, qui contient des hyperliens vers différents autres documents ;
- ❼ une note explicative au sujet de la façon dont la structure HTML a été produite. Le texte peut être modifié avec l'option `-desc` (voir section 2.5).

2.3. Personnaliser et étendre LaTeX2HTML

À part la spécification d'options sur la ligne de commande, LaTeX2HTML lit deux fichiers standard qui peuvent être utilisés pour personnaliser son comportement. Le premier `latex2html.config` est un fichier configuré au niveau système (qui se trouve habituellement dans le répertoire `/usr/local/lib/latex2html`). Il contient les définitions pour toute une installation, c.-à-d. communes à plusieurs utilisateurs, et spécifie où se trouvent certains programmes et utilitaires externes utilisés par LaTeX2HTML sur le système (p. ex. L^AT_EX, dvips, gs, pmbplus). De plus il donne une valeur aux variables `perl` qui contrôlent le comportement par défaut de LaTeX2HTML. À la fin du fichier, on peut spécifier quelles sont les commandes L^AT_EX à ignorer et lesquelles doivent être processées par L^AT_EX et incluses comme fichier image.

Le deuxième fichier, `.latex2html-init` permet une initialisation personnelle. LaTeX2HTML le cherche normalement dans le répertoire principal (`$HOME`) de l'utilisateur. On peut y spécifier la même information que dans le fichier de configuration `latex2html.config` et y définir comment doivent être traduites certaines commandes L^AT_EX en `perl`. Il faut noter que la distribution LaTeX2HTML vient déjà avec quelques fichiers qui contiennent les traductions pour les commandes supplémentaires introduites dans certaines extensions L^AT_EX, comme `german.perl`, `french.perl`, `html.perl` et `makeidx.perl`. Pour assister l'utilisateur, la distribution contient un fichier type `dot.latex2html-init` qui peut servir de modèle pour développer son propre `.latex2html-init`.

2.3.1. Créer son propre fichier de personnalisation `.latex2html-init`

Avant de donner quelques exemples, notons que le fichier `.latex2html-init` (ainsi que les autres fichiers du système LaTeX2HTML) doit contenir des instructions `perl` et qu'il faut avoir au moins quelques notions de ce langage de programmation.

La première partie du fichier-modèle `dot.latex2html-init` donne une valeur aux variables qui déterminent le comportement par défaut de LaTeX2HTML. Si vous ne voulez pas les changer vous pouvez les enlever. Remarquez la correspondance entre ces variables et les options de la commande `latex2html` décrites dans la section 2.1.2.

```
### Comportement par défaut de latex2html #####
$MAX_SPLIT_DEPTH = 8;# arrêter génération de fichiers séparés à ce niveau
$MAX_LINK_DEPTH = 4; # ne plus montrer les noeuds plus bas que ce niveau
$NOLATEX = 0;      # si =1 pas appeler LaTeX pour environnement inconnu
$EXTERNAL_IMAGES = 0;# si =1 laisser images à l'extérieur du document
$ASCII_MODE = 0;  # si =1 pas utiliser d'icônes ou d'images internes
$title = $default_title; # la valeur par défaut est "No Title"
$DESTDIR = '.';    # répertoire cible
# Spécifier format si vous n'aimez pas la valeur par défaut <Nom> <Date>
```

```

\documentclass{article}
\usepackage{makeidx,tlenc}
\usepackage[dvips]{graphics}
\usepackage[french]{babel}
\makeindex
\begin{document}
\begin{center}\Large
  Exemple d'un article en français\ [2mm]\today
\end{center}
\tableofcontents
\listoffigures
\listoftables
\section{Une figure EPS}
\index{section}
Cette section montre comment inclure une figure PostScript\cite{bib-PS}
dans un document \LaTeX. La figure~\ref{Fpsfig} est insérée dans le
texte à l'aide de la commande \verb!\includegraphics{colorcir.eps}!.
\index{figure}\index{PostScript}
\begin{figure}
\centering
  \begin{tabular}{c@{\quad}c}
    \resizebox{3cm}{!}{\includegraphics{colorcir.eps}} &
    \resizebox{3cm}{!}{\includegraphics{tac2dim.eps}}
  \end{tabular}
  \caption{Deux images EPS}\label{Fpsfig}
\end{figure}
\section{Exemple d'un tableau}
Le tableau~\ref{tab:exa} à la page \pageref{tab:exa}
montre l'utilisation de l'environnement \texttt{table}.
\begin{table}
\centering
  \begin{tabular}{cccccc}
    \Lcs{primo} \primo & \Lcs{secundo} \secundo & \Lcs{tertio} \tertio & & & \\
    \Lcs{quatro} \quatro & & 2\Lcs{ieme} \ 2\ieme & & & 
  \end{tabular}
  \caption{Quelques commandes de l'option \texttt{french}
    de \texttt{babel}}\label{tab:exa}\index{tableau}
\end{table}
\begin{thebibliography}{99}
\index{références}
\bibitem{bib-PS}
Adobe Inc. \emph{PostScript, manuel de référence (2ième édition)}
InterÉditions (France), 1992
\end{thebibliography}
\printindex
\index{index}
\end{document}

```

FIGURE 1 - Exemple d'un document \LaTeX qui sera traduit en HTML

Exemple d'un article en français

7 décembre 1994

Table des matières

1	Une figure EPS	1
2	Exemple d'un tableau	1

Liste des figures

1	Deux images EPS	1
---	-----------------	---

Liste des tableaux

1	Quelques commandes de l'option french de babel	1
---	--	---

1 Une figure EPS

Cette section montre comment inclure une figure PostScript[1] dans un document L^AT_EX. La figure 1 est insérée dans le texte à l'aide de la commande `\includegraphics{colorcir.eps}`.

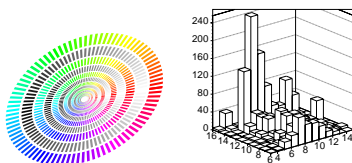


Figure 1: Deux images EPS

2 Exemple d'un tableau

Le tableau 1 à la page 1 montre l'utilisation de l'environnement `table`.

```
\primo 1° \secundo 2° \tertio 3° \quatro 4° 2\ieme 2°
```

Tableau 1: Quelques commandes de l'option french de babel

Références

[1] Adobe Inc. *PostScript, manuel de référence (2ième édition)* InterÉditions (France), 1992

Index

figure, 1	PostScript, 1	section, 1
index, 1	références, 1	tableau, 1

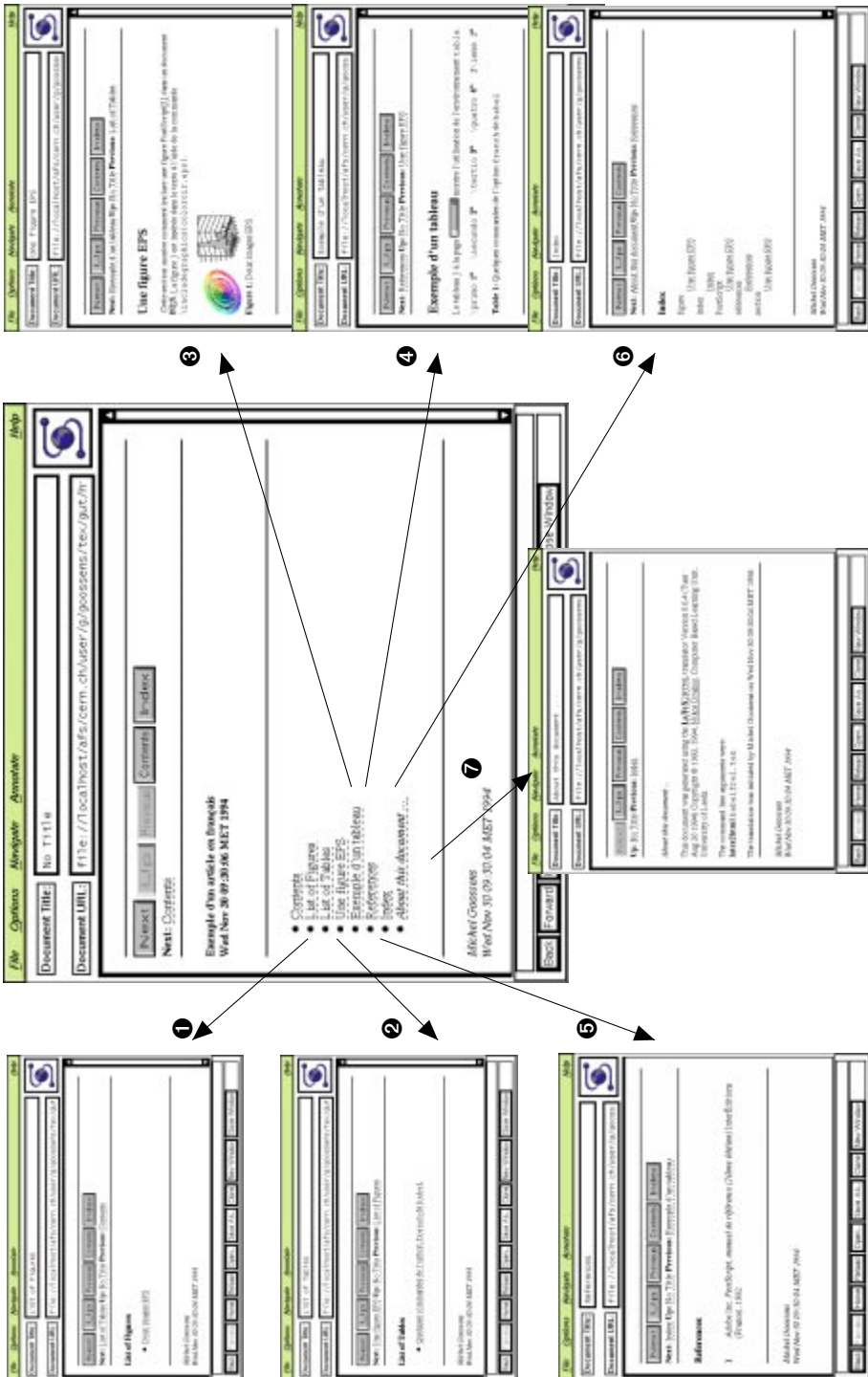


FIGURE 3 - La structure HTML produite à partir du fichier de la figure 1 et visualisée avec *Mosaic*

```
$ADDRESS = "<I>$address_data[0] <BR>\n$address_data[1]</I>";
$NO_NAVIGATION = 0; # si =1 pas générer un tableau de navigation
$AUTO_NAVIGATION = 1;# mettre tableau de navigation en haut de chaque page
$WORDS_IN_PAGE = 300;# si nb. lignes > $WORDS_IN_PAGE, mettre également
# tableau de navigation en bas de cette page
$INDEX_IN_NAVIGATION = 1;# hyperlien vers index dans tableau de navigation
$CONTENTS_IN_NAVIGATION = 1;# hyperlien vers table des matières
$NEXT_PAGE_IN_NAVIGATION = 1;# hyperlien vers document suivant
$PREVIOUS_PAGE_IN_NAVIGATION = 1;# hyperlien vers document précédent
$INFO = 1; # si =0 pas produire document "About this document ..."
```

Si vous désirez donc spécifier une forme précise pour votre adresse et ne jamais inclure les images à l'intérieur des documents, vous pourriez écrire par exemple:

```
$ADDRESS = "<I>Michel Goossens<BR>Division CN<BR>Tél. 3363</I>".
#<BR>\n$address_data[1]</I>";
$MAX_SPLIT_DEPTH = 2; # arrêter séparation au niveau sous-section
$MAX_LINK_DEPTH = 1; # montrer liens descendants juste pour les sections
$EXTERNAL_IMAGES = 1; # laisser images à l'extérieur du document
```

Normalement `LaTeX2HTML` va charger les fichiers d'extension et essayer d'interpréter les commandes qui s'y trouvent. Si cela risque de poser des problèmes ou si vous voulez définir vous-même un équivalent `perl` pour les commandes qui y sont définies, vous pouvez les inclure dans la liste d'extensions à exclure comme suit :

```
$DONT_INCLUDE = "memo:psfig:times:preprint:revtex:eqsecnum:" .
"aps:doublespace:float:harvard:tabs";
```

Les caractères spéciaux, les équations en ligne, etc. sont en général transformés en images internes qui sont placées sur la même ligne dans le texte. D'un autre côté les environnements de mise en évidence, comme les tableaux, les figures, les minipages, les équations multi-lignes, sont transformés en images placées dans un nouveau paragraphe. Les facteurs d'échelle pour ces deux types d'images (en ligne et de mise en évidence) sont contrôlés par les paramètres suivants :

```
$MATH_SCALE_FACTOR = 1.6;# échelle pour les images en ligne
$FIGURE_SCALE_FACTOR = 0;# échelle pour les images mises en évidence
# = 0, garder les dimensions originales
```

Comme indiqué ci-dessus, vous pouvez donner une liste de commandes qui doivent être ignorées ou passées directement à `LATEX`. Chaque commande doit être sur une ligne séparée et avoir la forme

```
nom_de_la_commande#{}# []# {}# [] etc.
```

où chaque paire d'accolades `{}` représente un argument obligatoire et chaque paire de parenthèses carrées `[]` un argument optionnel. On peut aussi spécifier du code `perl` à l'intérieur de chaque paire pour indiquer ce que **LaTeX2HTML** doit faire avec l'argument en question. Notons toutefois que si l'on veut garder le texte dans l'argument de la commande, les parenthèses en question *ne doivent pas être spécifiées*. Dans le cas des commandes à être traitées directement par L^AT_EX les paramètres sont transmis en même temps. Voici quelques exemples.

```
### Commandes à ignorer #####

&ignore_commands (<<_IGNORED_CMDS_);
documentclass # [] # {}
usepackage # [] # {}
linebreak# []
mbox
makebox# [] # []
_IGNORED_CMDS_

### Commandes à passer directement à LaTeX #####

&process_commands_in_tex (<<_RAW_ARG_CMDS_);
fbox # {}
framebox # [] # [] # {}
_RAW_ARG_CMDS_

1; # Ceci doit être la dernière ligne
```

Remarquons le fait que l'argument obligatoire des commandes `\mbox` et `\makebox` n'est pas spécifié, et qu'il reste donc dans le texte, alors que les arguments optionnels de `\makebox` disparaissent. Dans le cas des commandes `\fbox` et `\framebox` qui produisent des boîtes encadrées, les commandes et leurs arguments sont transmis tous deux à L^AT_EX.

Il est important de signaler que la dernière ligne dans le fichier doit impérativement être celle indiquée ci dessus.

2.3.2. Pour les gourous : étendre la fonctionnalité de **LaTeX2HTML**

Si vous avez de solides bases de `perl` et si vous avez le courage de plonger dans le code de `latex2html` (plusieurs milliers de lignes de `perl`), vous pouvez essayer d'ajouter des fonctions à l'interpréteur en définissant directement la traduction des commandes L^AT_EX en HTML.

D'abord il faut savoir que le programme **LaTeX2HTML** lit tous les fichiers associés aux sources L^AT_EX en mémoire, en substituant les commandes L^AT_EX par leur définition. Ceci implique que le processeur `perl` devra traiter plusieurs millions d'octets de données ce qui peut non seulement prendre un temps très long, mais également faire échouer le traitement faute de place en mémoire avec des fichiers

trop longs. Dans ce cas il faut diviser le document en plusieurs morceaux et utiliser les commandes évoquées ci-après pour créer des hyperliens entre les différents fichiers.

Pour donner quand-même une idée de la façon dont les commandes et environnements sont interprétés, nous donnerons deux exemples très simples, mais qui montrent bien la technique extrêmement puissante utilisée permettant de générer des documents HTML qui sauvegardent toute l'information présente dans le document L^AT_EX original.

Considérons d'abord une commande L^AT_EX (`\Ucom`) utilisée pour baliser une commande à entrer par l'utilisateur à sa console. La définition suivante est une réalisation possible :

```
sub do_cmd_Ucom {
  local($_) = @_;
  s/$next_pair_pr_rx//o;
  join(' ', qq+<KBD>$&</KBD>+,$_);
}
```

La variable `perl $next_pair_pr_rx` contient un motif de substitution qui extrait la suite de caractères entourés par la suivante paire de délimiteurs. La suite des caractères et les délimiteurs sont éliminées, puis cette même suite de caractères est recopiée, entourée des balises `<KBD>` et `</KBD>`.

Notre deuxième exemple montre un environnement numéroté `EnumZW` qui utilise des icônes sur un serveur WWW. Le nom de chaque icône dépend de la valeur de la variable `perl count`, qui est incrementée pour chaque commande `\item` à l'intérieur de l'environnement. Le tout se passe à l'intérieur d'une liste HTML de type `<DL>`.

```
sub do_env_EnumZW {
  local($_) = @_;
  local($count) = 0;
  s|\item|do {++$count; qq!<DT><IMG ALIGN=TOP ALT=""
  SRC="http://somewhere/icons/circled$count.xbm"><DD>!}|eog;
  "<DL COMPACT>$_</DL>";
}
```

Il doit être clair maintenant qu'avant d'essayer de développer ses propres commandes, on a intérêt à étudier de près la façon dont Nikos Drakos a écrit le code, non seulement pour être compatible avec son travail mais aussi pour s'en inspirer.

2.4. Extensions hypertextes de LaTeX2HTML

LaTeX2HTML introduit plusieurs commandes pour mieux gérer l'aspect hypertexte d'un document dans le contexte HTML. Ces commandes se trouvent dans

l'extension `html`, qui doit être spécifiée quand on veut les utiliser. Pendant la traduction ces commandes seront utilisées pour introduire l'information spécifique HTML dans le document final.

```
\htmladding{<URL>}
\htmladdnormallink{texte}{<URL>}
\htmladdnormallinkfoot{texte}{<URL>}
```

Ces trois commandes permettent l'introduction d'hyperliens de type URL dans un document. `\htmladding` renvoie à une image, alors que `\htmladdnormallink` associe la suite de caractères *texte* avec un autre document. Dans le document L^AT_EX formaté, la partie `<URL>` est ignorée, mais dans le HTML ces liens seront activés. La commande `\htmladdnormallinkfoot` est identique à la précédente, mais la valeur de `<URL>` est imprimée comme note de bas de page dans la version L^AT_EX du document.

Pour faciliter les renvois par hyperliens à d'autres documents déjà traités par L^AT_EX2HTML il existe deux commandes.

```
\externallabels{<URL répertoire externe>}{<fichier des étiquettes>}
\externalref{<étiquette dans fichier externe>}
```

La commande `\externallabels` spécifie comment accéder au fichier `perl` qui contient les identificateurs des étiquettes du fichier externe auquel on veut faire référence. Si le fichier des étiquettes réside sur une machine déportée, il doit être copié localement. La commande `\externalref` pour sa part spécifie l'identificateur dans ce fichier.

Pour inclure du code HTML pur dans le texte on dispose de l'environnement `\begin{rawhtml}... \end{rawhtml}`; tout ce qui se trouve à l'intérieur de cet environnement est ignoré par L^AT_EX.

Si l'on veut que certaines parties du document figurent seulement dans la version électronique (HTML) ou « papier » (PostScript) on peut utiliser les environnements `htmlonly` ou `latexonly` respectivement.

Pour mieux contrôler l'apparence des renvois dans la version imprimée et électronique d'un document la commande `\hyperref` a été créée.

```
\hyperref{texte_h}{texte_d1}{texte_d2}{étiquette}
```

Le premier argument *texte_h* est le texte qui sera mis en évidence (p. ex. souligné) dans la version électronique (seulement), alors que *texte_{d1}* et *texte_{d2}* entoureront le numéro produit par la commande `\ref{étiquette}`, ou *étiquette*, le quatrième argument doit être défini avec une commande `\label`.

Une commande un peu plus simple, mais qui a le même effet pour la version HTML est `\htmlref`, avec la syntaxe :

```
\htmlref{texte}{étiquette}
```

Le premier argument *texte* sera présent dans les deux versions mais dans la version électronique HTML le texte sera activé en devenant un hyperlien renvoyant à *étiquette*.

On a déjà vu dans la section 2.3 la façon dont on peut influencer globalement les dimensions des images produites par `LaTeX2HTML` (les variables `perl MATH_SCALE_FACTOR` et `FIGURE_SCALE_FACTOR`). Un contrôle plus individuel au niveau de chaque image est parfois désirable, ce que réalise la commande `\htmlimage`.

```
\htmlimage{options}
```

Cette commande peut être utilisée à l'intérieur de chaque environnement qui est normalement traduit en image (`tabular`, `equation`, `figure`, etc.). Les options spécifiées en argument déterminent la façon dont l'image créée sera traitée. L'argument *options* peut contenir une combinaison des valeurs suivantes, séparées par des virgules :

scale=facteur_d'échelle

le facteur d'échelle de l'image finale ;

external

l'image ne doit pas être incluse dans le document mais un hyperlien URL lui sera substitué pour y accéder ;

thumbnail=facteur_de_réduction

une version réduite (icônisée) de l'image sera incluse dans le document, et l'image à grandeur réelle sera disponible en externe (c.-à-d. **thumbnail** implique **external**).

Par exemple la commande

```
\begin{figure}
  \htmlimage{thumbnail=0.24}
  \includegraphics{mondessin.eps}
  \caption{Description de mon dessin}
  \label{fig-mondessin}
\end{figure}
```

créera une image icônisée réduite quatre fois par rapport à l'image originale dans le document hypertexte. En plus cette icône servira de bouton hyperlien pour

accéder à l'image en grandeur réelle comme document externe. La commande `\htmlimage`, il va sans dire, sera ignorée dans la version imprimée du document.

Pour être complet notons qu'il existe aussi une commande `\htmladdtonavigation` qui permet la création de boutons de navigation personnalisés. Plus de détails sont donnés dans le manuel d'utilisation `LaTeX2HTML`.

2.5. Exemple de document composé

Nous allons maintenant essayer de montrer à l'aide d'un exemple de document composé comment utiliser les commandes présentées dans la section précédente. Nous montrerons également comment personnaliser la sortie `LaTeX2HTML` en spécifiant des options sur la ligne de commande et en redéfinissant des variables dans un fichier qui contient des commandes `perl`.

Nous partons du document source \LaTeX de la figure 1 et le divisons en quatre parties, un fichier principal (`ex20.tex`) et trois fichiers secondaires (`ex21.tex`, `ex22.tex` et `ex2bib.tex`), qui sont tous montrés dans la figure 4. D'abord nous passons ces fichiers à la moulinette \LaTeX et puis *dans le bon ordre* par `LaTeX2HTML`. Cette dernière remarque est importante, comme nous utilisons des références croisées pour désigner des entités dans des documents externes (les commandes `\externalref` et `\externallabels`). Pour cela il faut d'abord traiter les fichiers `ex21.tex`, `ex22.tex` et `ex2bib.tex`, avant le fichier principal `ex20.tex`.

Par défaut la commande `latex2html` dépose les fichiers qu'il crée dans un répertoire qui porte le nom du fichier original, p. ex. un répertoire `ex20` sera créé si on exécute la commande `latex2html ex20.tex`. Les différents fichiers créés par le programme `LaTeX2HTML` à partir des quatre fichiers de notre document composé sont montrés ci-dessous :

répertoire principal

=====

```
569  ex20.tex
721  ex21.tex
627  ex22.tex
322  ex2bib.tex
```

sous-répertoire ex20

```
1187  ex20.html
109   images.pl
93    labels.pl
```

sous-répertoire ex2bib

```
844  ex2bib.html
109  images.pl
141  labels.pl
```

sous-répertoire ex21

```
1755  T_18854_figure15.gif
12118 _18854_figure15.gif
122   _18854_tex2html_wrap57.gif
```

sous-répertoire ex22

```
624  _15561_table12.gif
1047  ex22.html
512  images.pl
```

```

\documentclass{article}
\usepackage{html,t1enc}
\usepackage[dvips]{graphics}
\usepackage[french]{babel}
\begin{document}
\begin{center}
\Large Exemple d'un document composé
\end{center}

\htmladdnormallink{Les Images}%
{../ex21/ex21.html}

\externallabels{../ex21}%
{../ex21/labels.pl}
Référence à une figure
externe\externalref{Fpsfig}.

\htmladdnormallink{Les tableaux}%
{../ex22/ex22.html}

\externallabels{../ex22}%
{../ex22/labels.pl}
Référence à un tableau
externe\externalref{tab-exa}.

\htmladdnormallink{La bibliographie}%
{../ex2bib/ex2bib.html}
\end{document}

```

Fichier principal ((ex2.tex)

```

\documentclass{article}
\usepackage{html,t1enc}
\usepackage[dvips]{graphics}
\usepackage[french]{babel}
\makeindex
\begin{document}
\begin{section}{Une figure EPS}\label{sec-figure}
Cette section montre comment inclure une
\externallabels{../ex2bib}%
{../ex2bib/labels.pl}%
figure PostScript\externalref{bibPS}
dans un document \LaTeX. La
\hyperref{figure}{figure }{\Fpsfig}
est insérée dans le texte à l'aide
de la commande
\verb!\includegraphics{colorcir.eps}!.
\begin{figure}
\htmlimage{thumbnail=0.2}
\centering
\begin{tabular}{c@{\quad}c}
\resizebox{6cm}{!}{%
\includegraphics{colorcir.eps}} &
\resizebox{6cm}{!}{%
\includegraphics{tac2dim.eps}}
\end{tabular}
\caption{Deux images EPS}\label{Fpsfig}
\end{figure}
\end{document}

```

Fichier avec images (ex21.tex)

```

\documentclass{article}
\usepackage{html,t1enc}
\usepackage[dvips]{graphics}
\usepackage[french]{babel}
\newcommand{\Lcs}[1]{%
\texttt{\symbol{'134}#1}\enspace}
\begin{document}
\section{Exemple d'un tableau}
\label{sec-tableau}
Le \hyperref{tableau}{tableau }{\tab-exa}
montre l'utilisation de l'environnement
\texttt{table}.
\begin{table}
\centering
\begin{tabular}{cccccc}
\Lcs{primo} \primo & & & & & \\
\Lcs{secundo} \secundo & & & & & \\
\Lcs{tertio} \tertio & & & & & \\
\Lcs{quatro} \quatro & & & & & \\
2\Lcs{ieme} \ 2\ieme & & & & & \\
\end{tabular}
\caption{Quelques commandes de l'option
\texttt{french} de
\texttt{babel}}\label{tab-exa}
\end{table}
\end{document}

```

Fichier avec tableau (ex22.tex)

```

\documentclass{article}
\usepackage{html,t1enc}
\usepackage[dvips]{graphics}
\usepackage[french]{babel}
\makeindex
\begin{document}
\begin{thebibliography}{99}
\biitem{bib-PS}\label{bibPS}
Adobe Inc. \emph{PostScript, manuel de
référence (2ième édition)}
InterÉditions (France), 1992
\end{thebibliography}
\end{document}

```

Fichier de bibliographie (ex2bib.tex)

FIGURE 4 - Exemple d'un document composé (fichiers L^AT_EX)

1345	<code>ex21.html</code>	687	<code>images.tex</code>
539	<code>images.pl</code>	191	<code>labels.pl</code>
589	<code>images.tex</code>		
190	<code>labels.pl</code>		

Notons la présence des fichiers `labels.pl` qui associent pour chaque document les étiquettes symboliques des commandes `\label` aux fichiers physiques. Les autres fichiers sont le (ou les) source(s) HTML correspondant à la traduction du fichier L^AT_EX. Les images de type Gif sont produites quand L^AT_EX2HTML ne dispose pas d'une procédure explicite pour traiter une commande à l'entrée, qui est alors traitée par L^AT_EX et transformée en image PostScript, puis finalement en Gif (ou un autre format).

Pour montrer comment on peut personnaliser la sortie HTML nous avons préparé un petit fichier contenant quelques redéfinitions de variables `perl`.

```
# Fichier moninit.pl
# Personnalisation de latex2html
$ADDRESS = "<I>Michel Goossens<BR>Division CN<BR>Tél. 3363</I>".
           "<BR>\n$address_data[1]</I>";
$MAX_SPLIT_DEPTH = 0; # pas séparer en sous-documents
$MAX_LINK_DEPTH = 0; # pas montrer liens descendants
$NO_NAVIGATION = 1; # pas créer un tableau de navigation

1; # Ceci doit être la dernière ligne
```

Maintenant nous sommes prêts à traiter les documents L^AT_EX. Nous utiliserons l'option `-init_file` pour charger notre fichier de personnalisation `mininit.pl` montré ci-dessus, nous donnons un titre au document, puis nous spécifions le texte explicatif à utiliser en bas de la page, ce qui veut dire que, par exemple pour traiter le fichier `ex21.tex` qui contient des images, on utilise la commande :

```
>latex2html -init_file moninit.pl -t "Image" -info "Test du 2/12/94" ex21.tex
```

Ci-dessous nous montrons les messages informatifs générés pendant l'exécution de L^AT_EX2HTML. Notons d'abord le chargement du fichier `html.perl` qui est associé au fichier `html.sty` spécifié dans la commande `\usepackage` dans les documents de la figure 4. Il est obligatoire de charger cette extension si l'on veut utiliser les commandes hypertexte décrites à la section 2.4. Le fichier auxiliaire `ex21.aux` est également lu, ce qui veut dire qu'une exécution L^AT_EX d'un même document doit précéder l'exécution L^AT_EX2HTML. Après lecture du document complet, L^AT_EX2HTML génère un fichier `image.tex` qui contient le source de toutes les commandes et environnements qui ne peuvent être traités directement. À l'aide de L^AT_EX, ils seront transformés en images PostScript avec le programme `dvips`. Puis un autre programme `ghostview` interprète ce PostScript et le transforme en format Gif (en passant par un stade intermédiaire: le format `ppm`). Ce sont ces images

Gif qui sont utilisées par les programmes de visualisation pour montrer l'image à l'écran. Finalement **LaTeX2HTML** lit le(s) fichier(s) contenant les étiquettes correspondant aux autres documents pour introduire les références croisées à l'aide d'adresses de type `<URL>`.

This is LaTeX2HTML Version 0.6.4 (Tues Aug 30 1994) by Nikos Drakos,
Computer Based Learning Unit, University of Leeds.

OPENING /afs/cern.ch/user/g/goossens/tex/gut/html/ex21.tex

Loading /usr/local/lib/latex2html/styles/html.perl...

Reading ...

Reading ex21.aux

Translating ...0/2.....1/2.....2/2.....

Generating images using latex ...

This is TeX, Version 3.1415 (C version 6.1)

(18854_images.tex

LaTeX2e <1994/06/01> patch level 3

Hyphenation patterns for english, loaded.

Generating postscript images using dvips ...

This is dvipsk 5.58c Copyright 1986, 1994 Radical Eye Software

'TeX output 1994.12.02:1830' -> 18854_image

(-> 18854_image001) <tex.pro><special.pro>[1]

(-> 18854_image002) <tex.pro>

<special.pro>[2<colorcir.eps><tac2dim.eps>]

GS>GS>Writing 18854_image001.ppm

GS>Writing _18854_tex2html_wrap57.gif

GS>GS>Writing 18854_image002.ppm

GS>Writing _18854_figure15.gif

GS>GS>Writing 18854_image002.ppm

GS>Writing T_18854_figure15.gif

Doing section links

Done.

La figure 5 montre le résultat de tous nos efforts. Comme demandé il n'y a pas de tableau de navigation, des titres différents ont été précisés pour chaque document et la partie *About this document ...* et l'adresse contiennent le texte et la forme choisis. Les flèches qui portent les numéros ❶, ❷ et ❸ correspondent à des hyperliens pointant vers des document HTML à l'aide de la commande `\htmladdnormallink` dans le source **L^AT_EX**. Les flèches numérotées ❶ et ❷ correspondent à des références croisées introduites par les commandes `\externalref`, qui utilisent des étiquettes définies avec des commandes `\label` dans les documents cibles. La flèche portant le numéro ❸ correspond à l'hyperlien qui connecte l'image icônisée dans le texte avec sa forme grandeur réelle dans un fichier **Gif** externe. Finalement les points de début et d'arrivée du renvoi bibliographique sont signalés par le signe ➤.

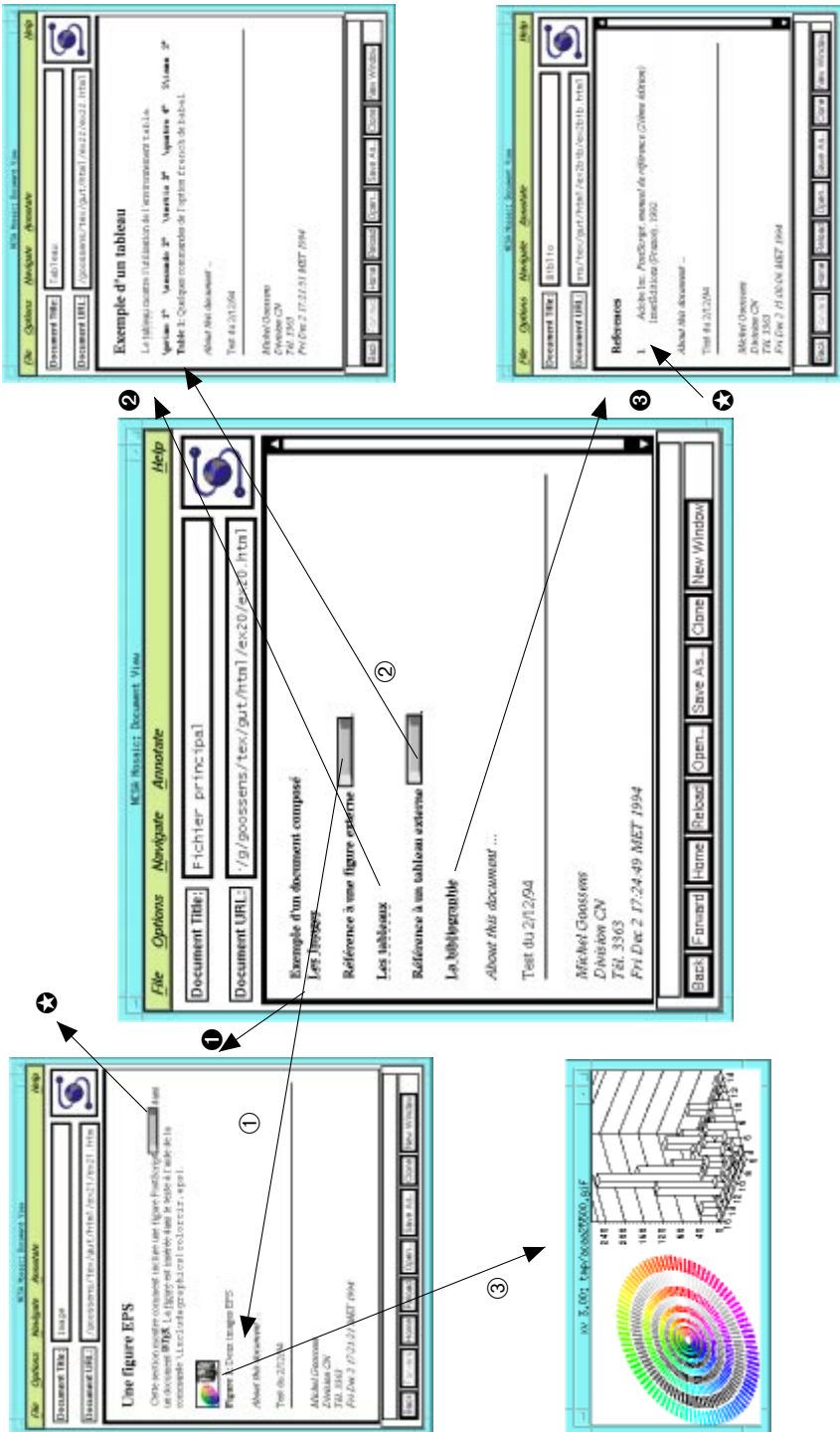


FIGURE 5 - La structure HTML produite à partir du document composé et ses sous-documents (figure 4) telle qu'elle est visualisée par Mosaic

3. Conversion de HTML vers L^AT_EX

Même s'il existe des utilitaires pour obtenir des sorties PostScript à partir des programmes de visualisation (p. ex. Mosaic) ou directement (p. ex. `https`⁵) il est souvent souhaitable de pouvoir disposer du texte sous une forme plus structurée, par exemple comme source L^AT_EX.

Un premier programme `HTML2LaTeX` fournit une traduction d'une grande partie de HTML vers L^AT_EX, alors que `SGML2TeX` est plus général et permet la transformation d'un source SGML arbitraire en L^AT_EX.

3.1. HTML2LaTeX, convertisseur de html vers L^AT_EX

`HTML2LaTeX` est un programme écrit en C par Nathan Torkington (Nouvelle Zélande). La partie traduction est basée sur l'analyseur (*parser*) HTML du programme de visualisation HTML Mosaic du NCSA. La syntaxe du programme est :

```
html2latex [options] [nom_de_fichier ...]
```

Pour chaque nom de fichier `HTML2LaTeX` transforme le balisage HTML à l'entrée en balisage L^AT_EX équivalent. Si aucun nom de fichier n'est spécifié, un message décrivant l'utilisation du programme est affiché. Si *nom_de_fichier* est égal à `-`, alors le texte est lu sur l'unité standard d'entrée `stdin`. Chaque fichier à la sortie portera le même nom qu'à l'entrée, mais avec une extension `.tex` plutôt que `.html`.

3.1.1. Options

`HTML2LaTeX` possède certaines options qui modifient sa façon de travailler. Ces options sont :

- `-n` numéroté les sections ;
- `-p` commencer une nouvelle page après la page de titre (si présente) ou la table des matières (si présente) ;
- `-c` produire une table des matières ;
- `-s` écrire la sortie sur l'unité standard de sortie `stdout` ;
- `-t Titre` produire une page de titre avec le titre *Titre* ;
- `-a Auteur` produire une page de titre avec un auteur *Auteur* ;
- `-h Texte_déb` introduire le texte *Texte_déb* juste après l'instruction `\begin{document}` ;
- `-f Texte_fin` introduire le texte *Texte_fin* juste avant l'instruction `\end{document}` ;
- `-o options` introduire les options *options* dans la commande `\documentclass`.

5. <http://info.cern.ch/hypertext/WWW/Tools/https.html>

3.1.2. Exemples

Quand l'utilisateur entre la commande suivante :

```
html2latex -n - < file.html | more
```

le fichier `file.html` est transformé en L^AT_EX et le résultat est affiché à l'écran. Comme l'option `-n` a été spécifiée, les sections seront numérotées.

Un exemple un peu plus complexe est le suivant :

```
html2latex -t 'Mes premiers pas en HTML' \
-a 'Prénom Nom' -p \
-c -o '[12pt,twoside]{article}' mon-article
```

Dans ce cas le fichier `mon-article` est lu et la sortie sera écrite dans le fichier `mon-article.tex`. Une page de titre (en utilisant le texte « Mes premiers pas en HTML » comme titre et « Prénom Nom » comme auteur) sera ajoutée sur une page séparée (option `-p`). Une table des matières, suivie d'une nouvelle page, est également demandée (option `-c`). Les sections ne seront pas numérotées (action par défaut). Le document sera composé avec le texte à 12 pt, et avec l'option `twoside`, qui prévoit une impression recto-verso.

3.1.3. Limitations

La version actuelle de HTML2LaTeX reconnaît les balises HTML suivantes : `<TITLE>`, `<H1>` à `<H6>`, pour les listes ``, ``, `<DT>`, `<DD>` et ``, puis les différents styles de présentation ``, `<I>`, `<U>`, ``, ``, `<CODE>`, `<SAMP>`, `<KBD>`, `<VAR>`, `<DFN>`, `<CITE>` et `<LISTING>`. En ce qui concerne les entités seulement `&`, `<` et `>` sont reconnues à présent. Le champ des balises `<ADDRESS>`, `<DIR>` et `<MENU>` n'est pas traité correctement. En plus l'attribut `COMPACT` de la balise `<DL>` n'est pas reconnue et le texte de la balise `<TITLE>` est ignoré. Plus grave, les balises `<PRE>` sont ignorées complètement.

Notons que tout le fichier HTML est lu en mémoire, ce qui pourrait créer des problèmes pour les grands fichiers sur des machines limitées en mémoire.

3.2. SGML2TeX, convertisseur de sgml vers L^AT_EX

SGML2TeX⁶ est un programme écrit par Peter Flynn (Cork, Irlande) qui convertit des balises SGML en instructions T_EX. Actuellement le système est écrit en PCL et ne tourne que sous MS-DOS sur PC mais son auteur prévoit de le réécrire dans un langage plus portable.

6. <http://info.cern.ch/hypertext/WWW/Tools/SGML2TeX.html>

SGML2TeX ne vérifie pas si le source SGML est correct, mais accepte tout document SGML en syntaxe concrète de référence. C'est l'utilisateur qui doit définir l'équivalent \LaTeX de tous les éléments, attributs ou entités trouvés dans le source. Un fichier de configuration peut contenir des correspondances prédéfinies pour certains des éléments, attributs ou entités, ce qui allégera sensiblement la tâche de l'utilisateur, qui n'aura qu'à ajouter les définitions manquantes. Par défaut les éléments SGML sont traduits dans une forme acceptable par \LaTeX en utilisant les conventions suivantes :

- les balises de début sont préfixées par `\start` et les balises de fin par `\finish` suivi du nom de la balise en majuscules qui est suivi d'une paire d'accollades (`{}`) ; une définition nulle est entrée dans le fichier d'extension pour chaque balise en question ;
- les entités SGML de la forme `&ent` sont traduites en `\ent{}` et entrées dans le fichier d'extension ;
- les attributs sont traités de la même façon, mais leur valeur est spécifiée entre accolades comme un argument \LaTeX .

Remerciements

Je tiens à remercier Mme Lysiane Besson et M. Arnaud Taddei pour leurs commentaires et suggestions sur une version préliminaire de ce document.

Références bibliographiques

- [1] JOHN K. OUSTERHOUT: *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, 1994.