Tools

TeX innovations at the Louis-Jean printing house

Maurice Laugier and Yannis Haralambous

Abstract

In this paper we will present several TeX innovations, conceived, or currently under development, at ILJ (the Louis-Jean printing house). ILJ was founded in 1804, at Gap (Southern French Alps) and employs 130 people. Due to its specialization in typesetting and printing of scientific books, ILJ has been using TeX since the late eighties. Currently about 30% of ILJ's book production is done with TeX. New developments in the TeX area sponsored or financed by ILJ are described in this paper.

— — * — —

In exactly ten years ILJ (Imprimerie Louis-Jean) will celebrate its bicentennial. Needless to say, this printing house has followed closely all developments of the printing industry: leaden types from the early XIXth century until 20 years ago, photocomposition in the seventies, and since the early eighties, the computer. Almost everything has changed: authors have changed, publishers have changed, even the product a printing house produces is not the same anymore: some years ago one was making books, now they are more and more often accompanied (and perhaps will be replaced in a few years) by those small silver disks, called CD-ROMs.

In the old days, the author would most probably supply a manuscript. Often one had to rival Champollion's skills to decipher these manuscripts, in order to be able to compose them. Later on one used to receive manuscripts in typed form; no deciphering was necessary any more, but this implied the work of an intermediate person, usually the author's secretary, or the author him/herself.

In the last decenny authors have bought personal computers; together with these engines they bought programs that make them believe they can typeset a book. More and more authors send "ready-to-print" books on floppies, which most of the time are "ready-to-throw-away". Since it is not possible to teach authors the rudiments of typography, one has to invest time and energy in *getting the most* out of these files, and finally be able to print the book the author had in mind. For this reason, one has to be able to offer the whole spectrum of services, starting with text input, and finishing with industrial printing.

Typesetting a book in TeX is an even bigger challenge, since the printing process requires also a strong know-how in programming: one has to know TeX sufficiently well to either write the TeX code for a book, or modify the code supplied by the author; one has to know SGML if the text is received marked up in that language and has to be converted into TeX, or inversely, if the author and/or publishier wishes to have the book in SGML form; one has to have some knowledge of PostScript in case something goes wrong at the color separation or flashing stage, and so on.

It follows that a minimum number of services ILJ has to supply are:

1. *Processing of TeX and LaTeX files, at the input,* DVI or PostScript level, that is:
   - writing TeX or LaTeX code, or converting Word, WordPerfect, Mathor, etc., documents into decent TeX/LaTeX code;
   - correcting it;
   - checking the page setup;
   - incorporating illustrations;
   - coloring it;
   - providing TeXnical assistance on the development of the LaTeX style file;
   - producing an SGML representation of it;
   - selecting or creating if necessary the fonts required for it.
2. Making high resolution films (1200 to 2400 dpi).
3. Industrial printing, binding, routing.

To be able to solve reasonably quickly the problems arising in a process as complicated as the one just described, ILJ had to develop a certain number of tools. The fact that TeX is an open system with no commercial maintenance was a risk to take; it also gave ILJ the opportunity of developing auxiliary tools which it would be impossible to make in conjunction with closed "ready-to-use" systems such as PageMaker or Quark XPress.

1 Oriental scripts

The second author, while working at the Institute of Oriental Languages and Civilizations in Paris, wrote a typesetting system for Oriental languages, based on TeX. ILJ has contributed to these projects both technically (by providing the necessary back end for Oriental typesetting and printing), and financially.

Together with John Plaice (Université de Laval, Québec), the second author is also developing Ω an extension of TeX internally based on ISO/IEC 10646/UNICODE. ILJ is ready to adopt ISO/IEC 10646/UNICODE as the fundamental encoding for text processing, in order to solve once and for all the problem of

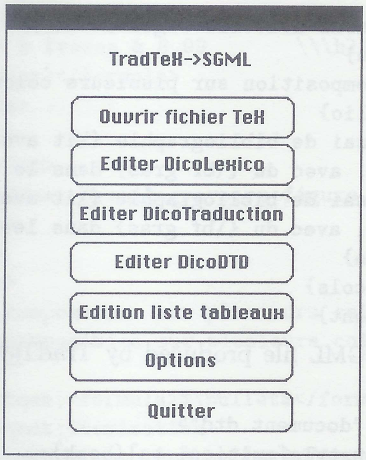

Figure 1: The generic menu of TradTEX→SGML

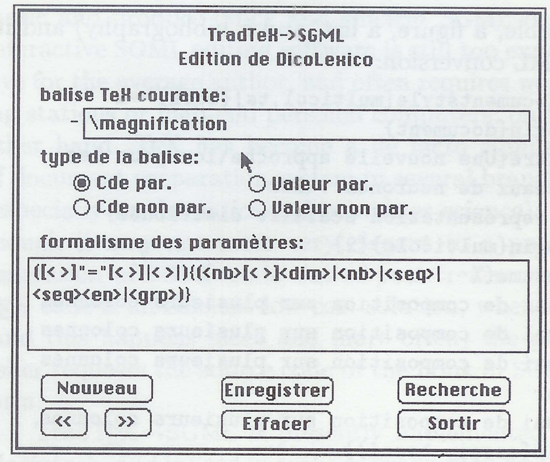

Figure 2: Editing the TradTEX→SGML database

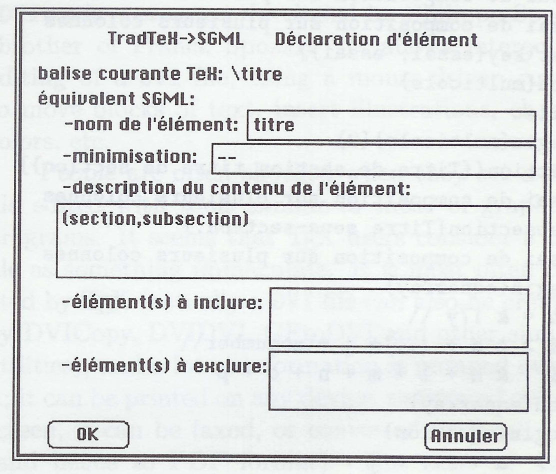

Figure 3: Declaring a TEX command in SGML

encoding ambiguities, a problem which can be very painful for texts with special needs (texts using symbols and/or non-Latin characters).

2 The TradTEX→SGML program

2.1 The principle

The TradTEX→SGML program, developed by Franck Spozio, is an assistance to conversion of TEX or LATEX files into SGML. The user of TradTEX→SGML must have a fairly good knowledge of both TEX and SGML, since the process of conversion involves a stage of analysis.

TradTEX→SGML creates a database containing all TEX codes it has encountered as well as their structure; this database file is accumulating information on TEX commands, from several runs: primitives, standard macros, or user-defined ones. Furthermore it checks the syntactic validity of these commands, in case the same command is defined with different structures in different files or contexts.

The file "DicoLexico" contains the TEX commands and their structures. As the reader can see in fig. 2, this database file can be edited and modified on the fly. The file "DicoTraduction" contains the SGML entities corresponding to the TEX commands, as well as their structure. The TEX codes which have to remain unchanged in the SGML file, in a <NOTATION> entity (for example, those describing math formulas) are stored in a special file, called "liste tableaux".

When reading the TEX code, TradTEX→SGML analyzes all TEX tokens; whenever a token has not been defined or contradicts the description contained in "DicoLexico", the user is prompted to specify the structure of the command (by a dialog as in fig. 3); the data requested are (a) the number of arguments, (b) the nature of delimiters, (c) if it is acting on one or more tokens coming after or before, (d) special characters, etc. Whenever the TEX command affects the formatting of the text, the corresponding informations are stored into the file "DicoDTD"; they will be used to create the DTD file which will mimic the original formatting specifications of the file.

Once this informations is stored, the user is prompted for the translation of the token in SGML (names of elements, attributes, processing instructions, entities, etc.). This information is then stored in file "DicoTraduction".

Once all tokens have been read and verified, the translation of the TEX file into SGML begins, using information from all four database files mentioned above; at the same time a DTD file is created. An example follows of a LATEX file (with two abstracts, a section and a subsection, an array of equations, a table, a figure, a list, and a bibliography) and its SGML conversion:

```
%\documentstyle[multicol,ts]{book}
\begin{document}
\titre{Une nouvelle approche\lcr pour les
réseaux de neurones :\lcr
la représentation scalaire distribuée}
\begin{multicols}{2}
\resume{%
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
\par
Essai de composition sur plusieurs colonnes
\cle{essai, essai}}
\abstract{%
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
\par\key{essai, essai}}
\end{multicols}
\filet
\begin{multicols}{2}
\section{{Titre de section titre de section}}
Essai de composition sur plusieurs colonnes
\subsection{Titre sous-section.}
Essai de composition sur plusieurs colonnes
\begin{eqnarray}
x & = & 17y \\
y & > & a + ... + j +\nonumber\\
 &   & K + l + m + n + o + p
\end{eqnarray}
\begin{equation}
y  >  a + ... + j +
\end{equation}
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
\begin{tabular}{||l|lr||}\hline
\haut gnats & gram & \$13.65 \bas\\\cline{2-3}
          & each &  .01        \\\hline
gnu & stuffed & 92.50 \\\cline{1-1}\cline{3-3}
emur&         & 33.33 \\ \hline
armandillo & frozen & 8.99      \\\hline
\end{tabular}
\end{multicols}
\begin{figure}
\vglue 3cm
\caption{Essai premi\'ere figure}
\end{figure}
\begin{multicols}{2}
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
\begin{itemize}
\item[$\bullet$] Essai d'itemisation
\item[$\bullet$] Essai d'itemisation
\end{itemize}
\begin{figure}
\vglue 8cm
\caption{Essai nouvelle figure}
\end{figure}
Essai de composition sur plusieurs colonnes
\begin{biblio}
\bib{1}{Essai de bibliographie {\it avec} de
l'italique, avec du {\bf gras} dans le texte}
\bib{2}{Essai de bibliographie {\it avec} de
l'italique, avec du {\bf gras} dans le texte}
\end{biblio}
\end{multicols}
\end{document}
```

The SGML file produced by TradTEX→SGML is:

```
<!DOCTYPE "document.dtd">
--\documentstyle[multicol,ts]{book}--
<doc>

<titre id=???>Une nouvelle approche<?\lcr>
 pour les r&eacute;seaux de
 neurones&espace;&colon;<?\lcr> la
 repr&eacute;sentation scalaire
 distribu&eacute;e
</titre>
<deuxcols>
<resume>
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
<par>
Essai de composition sur plusieurs colonnes
<cle>essai&comma; essai</cle></resume>
<abstract>
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
<par><key>essai&comma; essai</key></abstract>
</deuxcols>
<?\filet>
<deuxcols>
<section>Titre de section titre de section
</section>
Essai de composition sur plusieurs colonnes
<subsect>Titre sous-section</subsect>
Essai de composition sur plusieurs colonnes
<formula>\begin{eqnarray}
x & = & 17y \\
y & > & a + ... + j +\nonumber\\
 &   & K + l + m + n + o + p
\end{eqnarray}</formula>
<formula>\begin{equation}
y  >  a + ... + j +
\end{equation}</formula>
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
<formula>\begin{tabular}{||l|lr||}\hline
\haut gnats & gram & \$13.65 \bas\\\cline{2-3}
          & each &  .01       \\\hline
gnu & stuffed & 92.50 \\\cline{1-1}\cline{3-3}
```

```
emur&         & 33.33 \\ \hline
armadillo & frozen & 8.99      \\\hline
\end{tabular}</formula>
</deuxcols>
<figure>
<?\vglue 3cm>
<legende>Essai premi&egrave;re figure
</legende>
</figure>
<deuxcols>
Essai de composition sur plusieurs colonnes
Essai de composition sur plusieurs colonnes
<itemize>
<?\item>&lqsb;<formula>$\bullet$</formula>]
 Essai d&quot;itemisation
<?\item>&lqsb;<formula>$\bullet$</formula>]
 Essai d&quot;itemisation
</itemize>
<figure>
<?\vglue 8cm>
<legende>Essai nouvelle figure</legende>
</figure>
Essai de composition sur plusieurs colonnes
<biblio>
<bib>1</bib>Essai de bibliographie
<it> avec</it> de l&quot;italique&comma;
avec du <bf> gras</bf> dans le texte
<bib>2</bib>Essai de bibliographie
<it> avec</it> de l&quot;italique&comma;
avec du <bf> gras</bf> dans le texte
</deuxcols>
<par>
</doc>
```

And finally the DTD file:

```
<!NOTATION TeX SYSTEM "NotationTeX">
<!ELEMENT formula #NOTATION TeX #CURRENT>
<!ELEMENT par -O >
<!ELEMENT doc -- >
<!ELEMENT titre -- (section,subsection)>
<!ATTLIST titre id ID #IMPLIED>
<!ELEMENT deuxcols -- >
<!ELEMENT resume -- >
<!ELEMENT abstract -- >
<!ELEMENT cle -- >
<!ELEMENT subsect -- >
<!ELEMENT figure -- >
<!ELEMENT legende -- >
<!ELEMENT itemize -- >
<!ELEMENT biblio -- >
<!ELEMENT bib -- >
<!ELEMENT it -- >
<!ELEMENT bf -- >
<!ELEMENT section -- >
<!ELEMENT key -- >
<!ELEMENT eitemize -- >
```

When using a consistent and fairly stable set of TEX/LATEX macros throughout several documents, TradTEX→SGML can become more and more automatic and produce quick and efficient SGML code. Interactive SGML editing software is still too expensive for the average author, and often requires working stations or high-end personal computers; on the other hand, TEX has become a de facto standard of document preparation system in several branches (especially mathematics and computer science): although it may not seem very elegant to a purist, generation of SGML code out of post-treatment of TEX code is an efficient low-cost solution, whenever (and this happens more and more often) the publisher requests the source code of the book in SGML form.

TradTEX→SGML is presently implemented on Macintosh, and is used in real-life production by ILJ.

3 eDVItor

eDVItor is a program developed by Philippe Spozio (brother of Franck Spozio). It allows interactive editing of a DVI file, using a mouse-driven cursor to move blocks of text, insert illustrations, change colors, etc.

People are often shocked when they see a DVI file edited with tools similar to those of graphical programs. It seems that TEX users consider a DVI file as something immaculate. It is most often created by TEX (actually a DVI file can also be created by DVICopy, DVIDVI, GFtoDVI and other similar utilities), and a lot of information is pumped out of it; it can be printed on any device, previewed on any screen, it can be faxed, or converted to PostScript (and hence to PDF format). But none of these drivers and utilities change the text formatting in a DVI file; DVIDVI will perhaps change the order of pages, DVICopy will replace characters by other characters with the same metrics, and drivers do not modify a DVI file.

There is a reason behind this: according to TEX ideology, TEX does the ultimate text formatting, it would be vain to modify it manually. This of course is true, if we consider TEX's line breaking algorithm, or the typesetting of math formulas.

But what about titles, figures, or horizontal lines? We are forced to admit that these depend on the taste of the... human typesetter, rather than on TEX's skill. After all, to place a horizontal line or an illustration, we give millimetric instructions to TEX, concerning both the size of the object and the size of the surrounding white space. And these instructions can very well be wrong, or *slightly* wrong.

In the best of all worlds, one would run TEX on a file as many times as necessary, until the file is perfect from all points of view. In a real-life production world this is unfortunately not possible: a 600-page book can be run only a limited number of times. eDVItor allows us to make "those small last-minute changes", directly on the (otherwise perfect) DVI file.

Of course, the same changes have to repeated every time a DVI file is created anew (unless the corrector is smart enough to report the changes also to the TEX code file). In fig. 4, the reader can see a DVI file progressively modified by dragging blocks of text around.

eDVItor has been implemented both on DOS and on Macintosh. It allows also insertion of illustrations, by simple copy and paste operations, coloring of text and color separation of the whole document. A second version of this program, currently under α-testing, executes commands placed into the DVI file by the means of `\special` commands. Here are some commands which can be executed by eDVItor v2:

- Color processing:

```
\special{Color color1}
text in color 1
\special{Color color2}
text in coulor 2
\special{Color color1+color2}
texte in coulor 1+2
```

- Vertical column adjustment:

```
\special{Post Equalize
    Column1,Column2,column3
    Height=25cm Pages 17-27}
```

This command will spread paragraphs so that columns 1, 2 and 3 of pages 17–27 will have the same height, namely 25cm. To use this functions one has to specify first the parameters of stretching and shrinking of interparagraph blank space.

```
\special{Post Expand sup=1pc inf=-2mm}
```

This command will modify interline space.

- Positioning and moving around figures:

```
\special{Post InsertAt x=10cm y=2,4cm
Pages 1,[76] illustration figure.eps}}
```

EPSF figure `figure.eps` is placed at coordinates x =10cm, y =2.4cm on page 1 (folio 76).

- Inclusion of graphical commands: These commands allow framing of a block of text, or positioning of simple geometrical figures (rectangles, circles, ellipses), eventually filled with a certain color or gray density.

```
\special{Post SetFillDensity 30}
\special{Post FilledZoneFrame}
```

- Inclusion of external DVI files:

```
\special{Post Import fichier.dvi}
```

The DVI file `fichier.dvi` will be merged into the current DVI file. Suppose you are writing a book on TEX and want to include an example of TEX file output: for example a beginning of chapter page. Up to now there were two solutions: either simulate the beginning of chapter by writing the corresponding TEX code, or take the real TEX file you want to show, produce a DVI file, run dvips with the -E option and obtain an EPSF file, and include the latter in the original DVI file as an illustration. Of course the latter solution has the disadvantage that it is not device independent anymore: the EPSF file unavoidably contains bitmap fonts in a fixed resolution. *eDVItor allows you to include a* DVI file into another DVI file.[1]

It should be noted that modifications not involving PostScript are applied to the DVI file when processed by eDVItor: in that way, (a) one obtains a new DVI file, modified according to the `\special` commands, and processable by any DVI driver, (b) since the `\special` commands are in the TEX code, these modifications are automatically applied whenever eDVItor processes the DVI file (so one can produce new DVI files without fear of losing precious information added manually, as in the case of version 1 of eDVItor).

It is the hope of the authors that eDVItor will be the first step to a "WYSIWYG" TEX, in the sense of more effective DVI file manipulation and possibility of last-minute changes.

◇ Maurice Laugier
General Director of ILJ,
Imprimerie Louis-Jean, B.P. 87,
05003 Gap Cédex, France
Email: louijean@cicg.grenet.fr

◇ Yannis Haralambous
187, rue Nationale, 59800 Lille,
France.
Email: haralambous@univ-lille1.fr

[1] The second author always wondered how D.E. Knuth did volume E of Computers & Typesetting, where GFtoDVI printouts are mixed with TEX code.

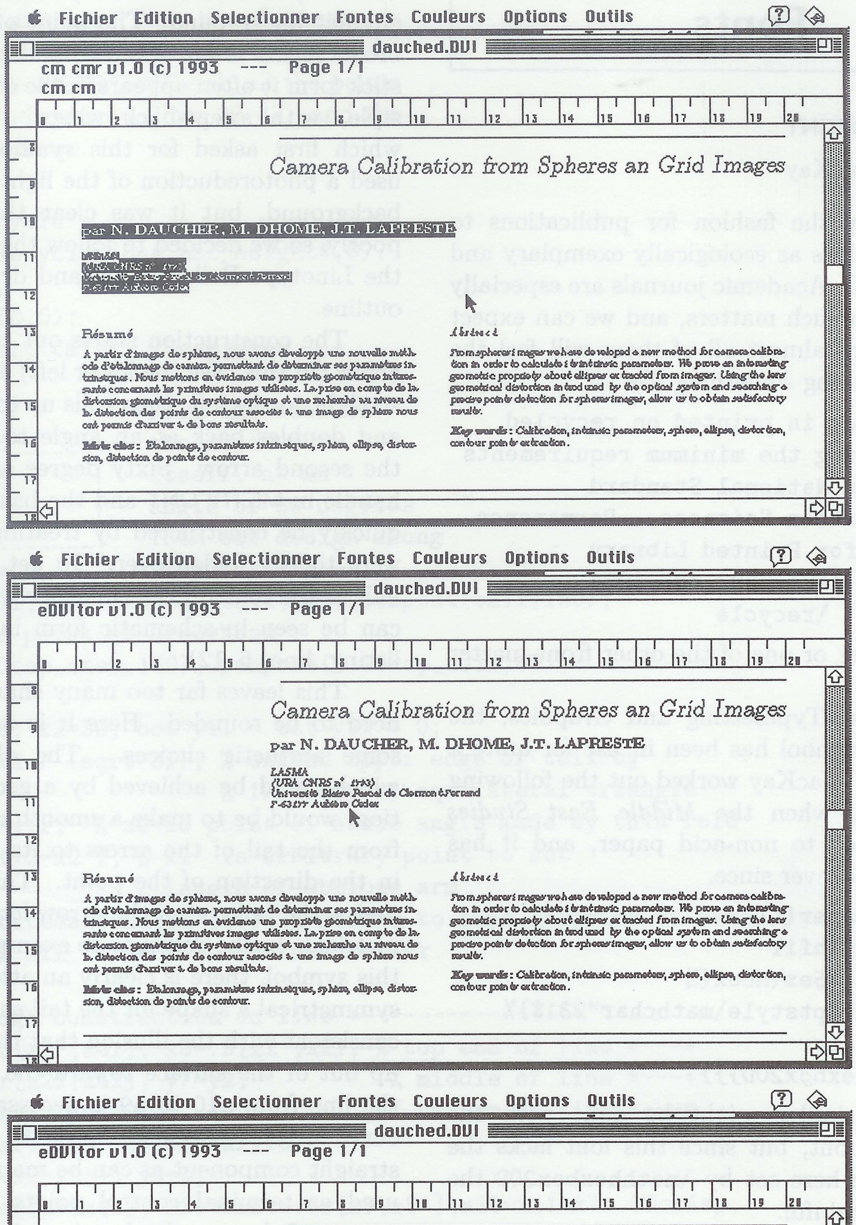

Figure 4: Editing a DVI file with eDVItor