BLUe's Bibliography—a generic approach

Kees van der Laan

Hunzeweg 57,
9893 PB Garnwerd, The Netherlands, 05941–1525
`cgl@risc1.rug.nl`

Abstract

A new and flexible approach to maintaining and using a bibliography database within (All)TEX is provided.

Keywords: Literature database, bibliography, plain TEX, macro writing, education.

Introduction

For typesetting a bibliography or a list of references many TEX-oriented tools are available[1]

- LATEX's bibliography environment, with its database facility BIBTEX (also used in AMS-LATEX)
- AMS-TEX's `\ref ...\endref`
- (L)TUGboat's, casu quo (L)TUGproc's conventions
- Alexander's TiB package
- Durst's Old Shell Game.

From the above it is concluded that there is no generally agreed upon (All)TEX tool to handle effectively and context-independently a list of references.

The incorporation of bibliographic information into a publication is complex because of the how-to-s

- set up and maintain a database of references
- incorporate the needed entries in a publication
- cite to the references from within the paper.

The static approach of selecting with an editor, and format these selected entries for the occasion, means a repetition of the clerical work for each publication. As long as one publishes occasionally that is no problem. Just go ahead, as simple as that.

A regular publishing author must adapt the mark-up of the references each time he submits an article. This is labour-intensive and error-prone. The more so when the local bulletin takes a different format from TUGboat, when we concentrate on publishing within the TEX-community.

This work emerged from the wish to maintain a database independently from any TEX publication series, parameterized such that it is easy to select entries from the database and to format these by customizing parameter macros.

In this paper an independent way to maintain a bibliography database, to select entries from the database, and to the format these, *all within* TEX, is worked out. For each publication the total database is used where the selection of the required entries is driven by a list of names pointing to the entries. To format the list appropriately a few formatting macros have been provided, which can be tailored to the conventions required by the publication series.

> For each publication the list of names `lit.sel` has to be extracted from the (total) list `lit.lab`, and `\ls` customized.

The problem of symbolic referencing to the list of references is solved en-passant, without the need for a multi-pass job.

BLUe stands for Mr BlUe—my innocent user and relative of Ben Lee User of the TEXbook fame.

Generic means that the proposed method can be used with (All)TEX, provided one adheres to a little discipline, and adjusts `\ls` for that particular situation.

Why yet another tool? Basically because BLUe's Bib

- is simpler
- is generic and cooperates with (All)TEX
- needs no (external) sorting
- needs no multi-pass job
- does not create new auxiliary files
- does not need external tools, except for the editor to handle the `lit.dat`, `lit.lab` files, in altering, or extending, the data(base).

Disclaimer.
I don't claim to have solved once and for all the formatting of the references, and the extracting of the database entries within TEX. All those user groups which publish TEX-oriented bulletins, have their peculiar formatting wishes. It is near to impossible to account for all of

[1] Wondering about how Knuth handled his references makes you realize that some persons are a class apart, in doing basic research. I would not dream of referencing via the index.

those conventions. Hopefully, BLUe's Bib will find its niche.

1 The basic approach

One has to decide upon the database tool, the tagging conventions, and the syntax of the entries.

The tool. There are many database tools available, for instance dBASE, next to more advanced relational database packages. What I need is a flexible, and general available tool

> which serves a lifetime.

TeX meets these requirements. So, I decided to use (All)TeX.[2]

Tagging what? The next problem is to decide about the amount of tagging. Of course one can tag all the items, and that is the best and most general approach.[3] However, I don't need that complete generality and certainly don't like its overhead, nor its hindering effects induced by the full generality. For me names and years are important issues related to any bibliographic entry. I know by heart the kind of publication once I know the author's name and the year of publication, in the area I'm working. When I can handle these fields, and also have some parameter macros to typeset the title and the name of the journal that would be enough. Moreover, I require that the entries will be supplied in a systematic and (near) natural way. Because of this modest approach I can always incorporate more tagging, casu quo formatting, when needed.[4] The hardest thing is to *maintain consistency*. I hope I will succeed by this simple and natural approach, although consistency is not imposed and I have no check for adhering to it.

Basically, I like to supply each entry in a (near) natural way, as suggested by the following example

```
Knuth, D.E (1984):
The \TeX book.
Addison-Wesley.
ISBN 0-201-13447-0 (hard cover)
ISBN 0-201-13448-9 (soft cover).
(For the right printing look for
 \cs{language}, or \cs{emergystretch}
 in the index.)
```

which obeys the syntax

```
<name part> (<date part>):
   <title part>
   <rest>
```

This syntax leaves freedom and flexibility, although the essentials—name, date, title[5]—have prescribed positions. For example the number of authors does not matter, nor is it prescribed how the authors should be specified. That is up to you. The representation of the date is free too, leaving room for things like 'in progress' or 'priv. comm.' Nor do I like to prescribe structurally the title in more detail. The rest part is not further prescribed either, because of the great variety of sources—publishers, books, journal series, proceedings, theses, technical reports, and the like.

I like to call the above the ASCII entry of the database.[6]

Database entries. In view of using the entries with TeX as selector and formatter, I decided to supply the entries as replacement text of a def, that is the entries have been supplied with a *name*.[7] For the name I chose to use lowercase letters only: the (first) name followed by the initials and extra letter(s), the latter in case of more than one publication per author. For example

```
\def\knuthded{<ASCII entry>}
```

I allow modest TeX mark-up in the ASCII entry. For example diacritical marks needed for the names, in the title part and so on. In the rest part I don't mind that TUGboat is indicated by TUGboat's convention `\tubissue` with its parameter specs. Furthermore, it is just handy to allow `\TeX`—TeX—and `\PS`—PostScript—and the like for TeX-related names, as supplied in `tugboat.cmn`. I'm not worried much about the amount of TeX mark-up because I expect TeX to be the formatter for the rest of my life!

1.1 Conventions.

My conventions for tagging each entry are as follows, mostly to enhance consistency.
Name part.
Start the name part with the family name of the (first) author followed by his initials with a period separating each next initial and followed eventually by interjections like `van` and the like. This followed by the next author names, if any, separated by a comma from each preceding author. Each name, except for the first, starts with its initials, separated by periods, and each last initial separated by a space from the family name.[8]

---

[2] Note that the file of entries can be maintained by an editor.

[3] Practised by BIBTeX, AMS, and Tib.

[4] In fact I doubted about to include a macro for tagging the titles. For the moment I just required that the title follows the colon and ends by its end-of-line. I guess that will be sufficient.

[5] I consider it important to delineate the main fields, however. That is the separation of the various parts. For example the title part is enclosed between a colon and its end-of-line.

[6] There is usually too much formatting in a bibliography. Why discriminate between a title of a book and an article? Given the context that is clear enough. Especially when for a book the ISBN (or ISSN) number is provided. And why superfluous punctuation? For me there should be as little interpunction as is functionally needed to separate the elements of the entry.

[7] In database jargon these names are the primary keys.

[8] Quite a mouthful already!

Date part.
I use the year, and when it concerns proceedings `editor, <year>`. I also adhere to `in progress` and `priv. comm.`
Title part.
Just provide the title, as straight as possible.
Rest part.
My usual hierarchy is to start with the source followed by secondary sources, the publisher data, ISBN etc., and annotations to end up with. That is: Name of journal, issue number, page numbers and a terminating period. Then some `(Also ...)` parts within parentheses. Publisher again terminated by a period.[9] ISBN or similar number again terminated by a period. And at the end annotations or comments within parentheses.

Selecting entries. For selecting I made use of the active list separator `\ls`, with context dependent meaning. When the database file consists of the entries

```
\def\knuthded
   {Knuth, D.E (1984):
   The \TeX book. \aw.
   ISBN 0-201-13447-0 (hard cover),
        0-201-13448-9 (soft cover).
   (For the right printing look for
    \cs{language}, or \cs{emergystretch}
    in the index.)}
\def\knuthdeg
   {Knuth, D.E (1986):
   The \mf book. \aw.
   ISBN 0-201-13445-4 (hard cover).}
\def\salomond
   {Salomon, D (1992):
    NTG's advanced \TeX\ course:
    Insights and Hindsights.
    MAPS Special, $\approx$500p.}
```

then for selecting them all

```
\ls\knuthded \ls\knuthdeg \ls\salomond
```

I like to call this list the file `lit.sel`.[10] Note that the list is alphabetically ordered.[11] This way of selecting can also handle the formatting, by an appropriate definition of `\ls`. For example

```
\def\ls#1{\advance\bcnt1
          \item{[\the\bcnt]} #1}
```

which with `\newcount\bcnt` will yield the entries as an 'item list,' similar to Knuth's example in TEXbook p.341.

In summary. While marking up a publication the following scheme must be obeyed to handle the formatting of a list of references

```
%Front matter
%Copy proper
%Back matter
\head*References* %Or similar
\input tugboat.cmn%Abbreviations
\input lit.dat    %the data(base)
\input lit.tex    %\def\ls{...} etc.
\input lit.sel    %list of selected names
\bye
```

2 Cross-referencing

The last problem to deal with is the handling of symbolic cross-referencing.

We know already the 'names' and the order of occurrences of the references via `lit.sel`. Because of this we can at the beginning of the publication associate numbers—or whatever one wishes to typeset as citations—to the entries in the list of publications.

The idea is to redefine temporarily the 'names' with as replacement texts what has to be typeset.[12] Let us keep it simple and work out the example of citation by numbers.

With the above structure of the `lit.sel` file one can simply provide the definition

```
\def\ls#1{\advance\bcnt1\edef#1{\the\bcnt}}
```

Let us call the file with the above definitions `lit.ass`.[13] By this approach the control sequence `\knuthded` will yield in the copy proper 1,[14] and so on.

In summary. Handle references, with the possibility to cite them by their names, and format them according to the definition of `\ls`, as follows in publication

```
%Front matter (title, abstract etc.)
\begingroup
\input lit.ass%association macros
\input lit.sel%names are linked
              %to citations
%begin copy proper
\head*Introduction*%Or similar
...%\knuthded etc. yields its citation
   %number or whatever you chose
\head*Conclusions* %Or similar
...
%end copy proper
\endgroup\noindent
%Back matter
\head*References*  %Or similar
\input tugboat.cmn %a.o. the abbreviations
\input lit.dat     %the data
\input lit.tex     %the formatting macros
\input lit.sel     %the list of names
\bye
```

---

9 City is superfluous in the context of an ISBN number.
10 Mnemonics: literature selection.
11 It can be ordered if needed by my Sorting in BLUe macros within TEX.
12 Earlier I considered typesetting the bibliography at the beginning of a document with appropriate page numbers, and associate citation texts. When printed the pages can be palced wher we want, or `dvitodvi` can be used.
13 Mnemonics: literature association.
14 Brackets can be added if wanted. I refrained from introducing them as part of `\ls` to allow for a sequence of citations.

Note the data integrity: the file `lit.dat` is only *used* at the end, and the file `lit.sel` is used at the beginning and at the end. No rewriting! The names have been redefined within a scope, however.

Examples

As examples I have formatted

- the list of references as printed at the end of this paper (LaTeX context), and
- my total database of TeX-related references (TeX context).

For cross-referencing input `lit.ass` and `lit.sel` at the beginning, and don't forget to provide for the grouping.

Within a LaTeX context. The list of references at the end of this article has been obtained, within the context of `ltugproc.sty`, via

```
\documentstyle{ltugproc}
%Front matter, with lis.ass, lis.selbb
%Copy proper of this article
%Back matter
\section*{References}
\input{lit.dat} %data
%\input{lit.tex}%just the next 2 lines!
 \frenchspacing
 \def\ls#1{\ea\bibentry#1\endgraf}
%\input{lit.selbb}%selected for BLUe's
                 % Bib the following list
 \ls\alexanderjc
 \ls\amsd
 \ls\amsf
 \ls\amsj
 \ls\arsenaud
 \ls\beebenhfc
 \ls\durstlka
 \ls\knuthded
 \ls\laancgk
 \ls\laancgm
 \ls\laancgo
 \ls\lamportlb
 \ls\rahtzspq
 \ls\whitneyrf
\end{document}
```

Within a (plain) TeX context. The printing of my complete (All)TeX-related database, can be done within plain as follows

```
\beginsection Bibliography\par%TB 340
\input tugboat.cmn%abbreviations
\input lit.dat    %complete database
\input lit.tex    %formatting macros
\input lit.lab    %list of all names
\bye
```

with, when context free, `\ls` as supplied earlier

```
\def\ls{\advance\bcnt1
            \item{[\the\bcnt]} #1}
```

and, within the AMS spirit[15]

```
%The file: lit.tex
%macros to format lit.dat,
%driven by lit.lab
\frenchspacing
\newcount\bcnt \newcount\suffixcnt
\let\lstnme\relax \let\lstyear\relax
%
\def\ls#1{\ea\bibitem#1}
%
\def\bibitem#1(#2){\global\advance\bcnt1
 \def\authornme{#1}\def\authoryear{#2}%
 \ifx\lstnme\authornme
  \def\authornme{------}%
  \ifx\lstyear\authoryear
   \global\advance\suffixcnt1
   \def\authoryear{--}%
  \else\let\lstyear\authoryear\suffixcnt0
  \fi
 \else\let\lstnme\authornme
      \let\lstyear\authoryear\suffixcnt0
 \fi
 \item{[\the\bcnt]}\authornme\,(\authoryear
  \suffix)}%end \bibitem
%
\def\suffix{\ifcase\suffixcnt\or a\or b\or
 c\or d\or e\or f\or g\or h\or i\or j\or
 k\or l\or m\or n\or o\or p\or q\or r\or
 s\or t\or u\or v\or w\or x\or y\or z\fi}
\endinput
```

and with data

```
%The file: lit.dat
\def\alexanderjc
  {Alexander, J.C (1986):
   Tib, a reference setting package.
   \tubissue{7}(3), 138\dash139.
   (An update note is in
                     \tubissue{8}(2), 102)}
%...and so on
```

3 Maintenance?

The *maintenance of the data* comes down to extending and maintaining the file `lit.dat`. This has nothing to do with the formatting proper of a publication at hand. Because of the systematics used, especially that `\def\<name>` is on one line, it is easy with a programmable editor, an AWK script or a similar tool, to extract these lines and replace `\def` by `\ls`. In doing so the file `lit.lab` has been obtained without retyping, and therefore without retyping errors.

For each publication an author has to create the list `lit.sel` as a subset of the list `lit.lab`, again by an editor or a similar tool. A discipline is needed when references have to be inserted, while proofing. Update `.dat`, `.lab`, `.sel`, by hand I guess, or start first by preparing the contents and do finally the bibliography. It's all a matter of discipline.

[15] It is always cumbersome to obey the wishes of others.

4 Looking back

When developing a tool the most difficult thing is to know when to stop. A work is never perfect nor finished, there is always the polishing phase. My experience is that when I think I'm finished it turns out to be true for only 80% or so! The energy needed for the last 20% is as much as for the first 80%, if not more! In the spirit of Rahtz' UKTUG presentation of 1990 on the issue, I pondered about the advantages, disadvantages and what else?[16]

Advantages.
- simple (approach, TEX encodings, and use)
- generic (can be used with any TEX flavour)
- one tool—TEX—for selecting (via TEX's hashing of accessing def-s) and for formatting
- near natural way of specifying the entries, limited tagging
- one-pass job
- open-ended, extensible.

Disadvantages.
- (unique) names have to be looked up, no pattern-matching search
- limited tagging of the elements of the entries
- no support for adhering to consistency, otherwise than using control sequences.[17]

What else? With respect to data-integrity and alleviating the clerical work—or avoiding the intelligent AWK-like scripts—it would be worthwhile to generate `lit.lab` automatically from the natural specified data, via pattern matching techniques. Ipso facto for `lit.sel` driven by specification of free keywords. For the moment I stopped, however, and will look over BLUe's shoulder how things go in practice.

Acknowledgements

The idea of suppressing repeated names in the example of printing the complete database, is borrowed from AMS-TEX.

Erik Frambach is kindly acknowledged for proofing the article, and for stressing to add the reason why, as well as for suggesting that it would be helpful to include summaries of earlier works on the issue.

Conclusions

A simple, flexible and generic approach for handling a database of bibliographic entries is proposed, within the context of TEX, independent from a special TEX flavour.

The approach allows for selecting, formatting and cross-referencing.

I use this method for a database with hundreds of entries, with a few percent to be selected each time.

As examples it is shown how to process the included list of references within a LATEX context, and how to process the complete database within a (plain) TEX context as such and adhering to AMS' style.

Sorting on the fly, see [5, 11]—whoops this citation has been done via `[\arsenausd, \laancgo]`—is no longer needed, nor does one need a multi-pass job.

As suggested in my AMS BLUes paper, the above can be worthwhile for a publisher, relieving authors from the details of formatting a bibliography by letting them just supply the names to the database of (pre)formatted entries, available already at the publisher's computing environment.

References

Alexander, J.C (1986): Tib, a reference setting package. *TUGboat* 7, no. (3), 138–139. (An update note is in *TUGboat* 8, no. (2), 102. A C program inspired upon troff's refer (and its successor bib). It is a preprocessor tailored for use with (All)TEX. It allows for a few citation styles, and a publisher can extend these. It provides also for a powerful word-definition citation search. Flexible in general. Tagging of the entries a la refer has to be obeyed. From the UKTuG meeting of 1990 the following characteristics. It consists of: tibdex, a hashing program to speed up access to large collections of citations, by creating an index to the bibliography; tiblist, to format and print the database; looktib, a program to query the database; tib, main program goes through the source document and looks for keywords between [and]. Advantages: 1-pass, pattern-matching citation no need to invent unique key, compatible with UNIX refer databases, good toolbox of style elements, single index of bibliographies, formatting more-closely related to TEX. Disadvantages: pre-processor system, possible wrong matches, possible errors in repeated citations, non-algorithmic style language, left up to Tib to work out citation type, lack of extensibility in field, not very robust (crashes).)

AMS (1993): AMS-TEX User's Guide 2.1.

AMS (1993): AMS-LATEX User's Guide 1.1.

AMS (1993): Guidelines for preparing electronic manuscripts. AMS-TEX (booklet, 52p), and the mirrored one AMS-LATEX (booklet, 58p). (The first is very well-done. I have not seen a guideline of similar quality of yet! Simply the best available. Much experience is embodied to learn from. The second is verbose, incomplete, and deals at length with issues an author should not be bothered with.)

[16] Keep in mind however, that a disadvantage can also be interpreted as an advantage and vice versa! For example simpleness can be seen as an advantage, but is experienced as a disadvantage when bells-and-whistles are needed.

[17] An author can be relieved alltogether of the concistency aspects by allowing him to specify the lables, *not the contents!*

Arsenau, D (1992): overcite.sty, drftcite.sty, citesty. (From the file server. No longer needed when BLUe's Bib is used.)

Beebe, N.H.F (1991): The TUGlib server. MAPS 91.2, 117–123. (Also TEXline 11.)

Durst, L.K (1989): Bibliographic citations, or variations on the old shell game. *TUGboat* 10, no. (3), 390–394. (It discusses how to cope with TEXing a bibliography. BLUe's Bib has undoubtedly been influenced by this work, because I read it at least a year before creating BLUe's Bib. Durst does not use the list separator TEXnique, and therefore the encoding is a little more complex. Also the writing to a file and the external sorting makes it more cumbersome. In the paper it is not worked out how to use it within one of TEX's flavours. The basic approach is very similar to mine, however.)

Knuth, D.E (1984): Computers and Typesetting. The TEXbook. Addison-Wesley. ISBN 0-201-13447-0 (hard cover) ISBN 0-201-13448-9 (soft cover). (For the correct printing look in the index for `\language` or `\emergystretch`.)

Laan, C.G van der (1993): Manmac BLUes—or how to typeset a book via TEX. MAPS 93.1, 171-191.

Laan, C.G van der (1993): Sorting in BLUe. MAPS 93.1, 149-170. (Abridged TUG '93. *TUGboat* 14, no. (3), ?—?)

Laan, C.G van der (1993): Typesetting number sequences. MAPS 93.1, 145–148. (Submitted TUGboat. No longer relevant when BLUe's Bib is used.)

Lamport, L (1985): LATEX User's Guide & Reference Manual. Addison-Wesley. ISBN-0-201-15790-X. (With respect to BIBTEX the following characteristics from UKTUG meeting of 1990. Advantages: clear layout of database, unique identifyer for elements, compatibility with Scribe databases, explicit statement of citation type, extensible style language, uses LATEX's cross-referencing, easy to edit output, can be mixed with non-automatic generated bibliographies, cross-referencing and abbreviations. Disadvantages: multiple passes (LATEX, BIBTEX, LATEX, LATEX); have to remember unique references, style language is opaque, database is very wordy and boring to enter.)

Rahtz, S.P.Q (1987): Bibliographic tools. Literary and linguistic computing, 2, 4, 231–241.

Whitney, R.F, B.N Beeton (1989): TUGboat authors' guide. *TUGboat* 10, no. (3), 378–385. (Updated versions via the file server.)

Matrix icons via LaTeX

Kees van der Laan

Hunzeweg 57,
9893PB Garnwerd, The Netherlands
`cgl@risc1.rug.nl`

Abstract

Some macros for typesetting matrix icons are provided, based upon LaTeX's picture environment, which don't need coordinate awareness: just the (dimensionless) sizes of the 'boxes.'

Keywords: Matrix icons, LaTeX, picture environment, macro writing, education.

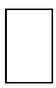
Introduction

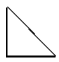
In my Math into BLUes I used matrix icons for formatting the schematic flow of matrix decompositions, to mimick Wilkinson's diagrams as supplied in his The Algebraic Eigenvalue Problem. The crucial element is the creation of diagonal lines. I used Amy Hendrickson's `\diagline`, which is flexible and powerful, but suffers from non-uniformness over the line orientation, and because of that it is difficult to control the line thickness for diagonal lines.

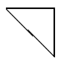
In this paper I used the line fonts which come with LaTeX via the `\line` macro, as part of the picture environment.

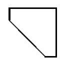
Given the context of matrix icons the limited availability of line orientations does not hinder.

From literature I distilled that the following special matrix icons are useful

rectangular matrix 

lower left triangular matrix 

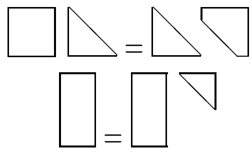
upper right triangular matrix 

upper Hessenberg form 

The above forms can be drawn easily in the picture environment. The macros in this paper hide that environment, and provide the pictures in a `\vbox`, for use at any place where a `\vbox` can be used, without to worry about 'coordinates.'

Examples

- Some matrix icons, Wilkinson(1965)



$AL = LH$

$A = QR$

via

```
\setlenght{\unitlength}{1ex}
$$\vcenter{\icmat44\kern\unitlength
  \iclLt44=\iclLt44\icuh413
  \qquad AL=LH}$$
$$\vcenter{\icmat63=\icmat63\kern
  \unitlength\icurt63\qquad A=QR}$$
```

The problems

For (portable) line diagrams within LaTeX the picture environment is generally used. The drawback is that one has to be aware of coordinates, to put everything in the right position.

Diagonal lines have to be built up, either from line elements provided by some fonts, see Lamport and Spivak, or from 'dots,' see Hendrickson.

An intrinsic problem is that in contrast with horizontal and vertical rules the line thickness of the diagonal lines can't be changed easily.

Design

Given the context of the limited number of matrix icons and their restricted shapes, I decided to

- build upon the line elements which come with the LaTeX fonts
- use the LaTeX picture environment
- parameterize the size via LaTeX's `\unitlength`
- box all the icons in a `\vbox`
- free the use from the knowledge of the picture environment, and the associated coordinates.

The macros

```
\def\icmat#1#2{%ICon MATrix(rectangular)
%#1 is ht of icon matrix, e.g. 4
%#2 is wd of icon matrix, e.g. 2
\vbox to#1\unitlength{\hrule
   \hbox to#2\unitlength{\vrule
     height#1\unitlength\hfil\vrule}%
           \hrule}%
}%end icmat
%
\def\icurt#1#2{%IConUpperRightTriangle
%#1 is ht of icon matrix, with UT
%the upper triangular part, e.g. 4
%#2 is wd of icon (upper triangular)
%matrix, e.g. 2
\vbox to #1\unitlength{\hrule
   \hbox{\picture(#2,#2)%
    \put(0,#2){\line(1,-1){#2}}%
    \endpicture\vrule}%
   \vfil}%
}%end icurt
%
\def\iclt#1#2{%IConLowerLeftTriangle
%#1 is ht of icon matrix, with LT
%the lower triangular part, e.g. 4
%#2 is wd of icon (lower triangular)
%matrix, e.g. 2
\vbox to #1\unitlength{\vfil
   \hbox{\vrule\picture(#2,#1)%
     \put(0,#2){\line(1,-1){#2}}%
     \endpicture}%
   \hrule}%
}%end iclt
%
\def\icuh#1#2#3{%IConUpperHessenberg
%#1 is size of icon matrix, with UH
% the upper Hessenberg part, e.g. 4
%#2 is wd of icon (upper Hesenberg)
% matrix, e.g. 1
%#3 is size Lower Left triangular part,
% #1-#2 (for simplicity the latter is
% added, could have been calculated,
% perhaps some inconsistency test could
% be incorporated)
\vbox to #1\unitlength{\offinterlineskip
   \hrule
   \hbox to#1\unitlength{\vrule height%
       #2\unitlength depth0pt\relax
       \hfil\vrule}%
   \hbox to#1\unitlength{\picture(#3,#3)%
    \put(0,#3){\line(1,-1){#3}}\endpicture
    \hfil\vrule}%
   \hbox to#1\unitlength{\hfil\vrule
     width#2\unitlength height.2pt\relax}%
   }%
}%end icuh
```

Conclusions

The macros for drawing matrix icons hide the use of LaTeX's picture environment and make the use easier and more general. A drawback is still that the line thickness can't be varied, especially for the diagonal lines. Be my guest and extend the given ones with those you need. I welcome your results.

References

Hendrickson, A (1985): Some diagonal line hacks. *TUGboat* 6, no. (2), 83–86.

Knuth, D.E (1984): The TeXbook. Addison-Wesley, 1984.

Laan, C.G van der (1991): Math into BLUes.
Part I: Mourning. Proceedings TUG '91, *TUGboat* 12, no. (4), 485–501.
Part II: Sing your song. Proceedings EuroTeX '91, *GUTenberg Cahiers*, 10&11, 147–170.
(Earlier version of I&II in MAPS 91.1)

Lamport, L (1986): LaTeX, user's guide & reference manual. Addison-Wesley.

Spivak, M.D (1991): LAMS-TeX Wizard's manual. TeXplorators.

Wilkinson, J.H (1965): The Algebraic Eigenvalue Problem. Clarendon Press.

An application of literate programming: creating a format for the bulletin of the Polish TUG*

Włodek Bzyl † and Tomasz Przechlewski ‡

Introduction

On establishing the Polish TeX Users Group the authors were appointed to create its bulletin. One of the first tasks was a creation of a format for the bulletin. We wanted it to be easily maintained and fully documented. The format had to be sufficient to understand, appreciate and later to modify the code by ourselves and/or others.

It was decided that the format should be coded in `WEB`. Knuth's `WEB` System of Structured Documentation was thought to fulfill our expectations. We chose `FWEB`, a multilingual implementation of `WEB` by John A. Krommes (Princeton University) based on Silvio Levy's `CWEB`.

`WEB` programs are easily modified to different environments. Ideally, we should have started with an existing `WEB` file and then modified it via a change file. However, there were not any formats written in `WEB` at that moment. To that end we had to translate to `WEB` the format of our choice—TUGboat style. Oddly enough, we became pioneers.

Problem

There are only few sets of macros for typesetting bulletins in the public domain (TTN, TUGboat). The TUGboat format is the best known, widely used one and can be obtained from almost all archives. TUGboat is designed for only one language—English. As our format is bilingual we had to modify *The TeXbook* style. Apart from that the modification was necessary because:

- CM fonts had to be replaced with Polish fonts which contain Polish diacritical characters;
- some parts of the code had to be changed, some adjusted to get a design that was slightly different from TUGboat design;
- the parts of the code unnecessary for our purposes at that moment had to be removed.

Moreover, taking into account the technology of printing the bulletin we decided to use Computer Concrete family of fonts instead of Computer Modern. It yielded decent results.

Template of WEB program

The `@` symbol in the `WEB` language is an 'escape character'. It introduces commands. A `WEB` file has the form of a sequence of two elements: comments and code. They are separated by `WEB` commands started with `@`. The skeleton of a `WEB` file is shown below:

```
@Lx

@* Title.   ...  Some documentation

@A
TeX macros

@* Title.   ...  more documentation

@A
more TeX macros
etc...

@* Index.
```

The parts of the code introduced by the `@*` sequence (or `@ `, `@*1`, `@*2`) are sections (unnamed sections, subsections etc.).

`@Lx`	switch to TeX language
`@@`	`@` symbol
`@`	start an unnamed section
`@*`	start a major section
`@*1, @*2`	start a subsection (subsubsection)
`@A`	begin TeX code
`@O`	open new output file

Figure 1: *List of frequently used WEB commands*

Converting to WEB

The TUGboat format originally comes in three files:

1. `tugboat.cmn`,
2. `tugboat.sty` and
3. `ltugboat.sty`.

All files contain a lot of comments explaining the code. `tubguide.tex`, which is separately delivered, is a kind of a 'user guide' for authors. It is possible using `WEB` to combine all the files into one.

*Published in TUGboat **14.3**, 296–299 (1993).

†Uniwersytet Gdanski, Instytut Matematyki, Wita Stwosza 57, 80-957 Gdansk, Poland, `matwb@halina.univ.gda.pl`
‡Uniwersytet Gdanski, Katedra OPD, Armii Krajowej 119/121, 81-824 Sopot, Poland, `ekotp@halina.univ.gda.pl`

Unfortunately, TUGboat format had not been converted into WEB. So the first thing to do was the conversion. The process of making a *.web file is quite simple. One can do this in the following way:

- Change explanatory comments to sections by removing per cent signs and preceding the whole text with @ or @* symbols (short comments may be left untouched);
- precede macros with @A;
- double @ in macros.

The first step is the most important one as the structure of a document is decided at that moment (sections, subsections, etc.).

What is WEB

The web file is a structured document. It consists of documentation and macros simultaneously. A web file is processed with two preprocessors: TANGLE and WEAVE. TANGLE strips off documentation and reorganizes the code. WEAVE produces documentation in TeX format.

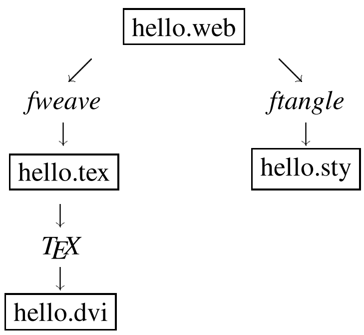


Figure 2: *WEB data flow*

Modifying WEB programs

Both processors, WEAVE and TANGLE, can work with two input files: *web* file and *change* file. A change file contains data which override selected portions of *web* file. The structure of a change file has the following form:

```
@x
 ... old lines ...
@y
 ... new lines ...
@z
```

Any text up to the first @x, and between @z and @x, will be bypassed (some additional comments are put there usually).

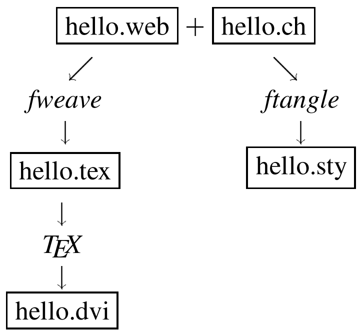


Figure 3: *WEB data flow using a change file*

The whole process is illustrated with the following toy example. Let the file hello.web contain the following six lines:

```
@Lx

@* First example.
@A
\def\greetings{Hello!}
@* Index.
```

and the change file hello.ch adapts it to the Polish language.

```
change to Polish language
@x
\def\greetings{Hello!}
@y
\def\greetings{Cze\'s\'c!}
@z
```

Conclusions

We have found this approach useful in spite of the fact that we did not use all features of WEB. Named modules are supported by FWEB, but we did not use them or the conditional exclusion/inclusion or macro definitions.

FWEB is available via anonymous ftp from ftp.pppl.gov:/pub/fweb. It runs on IBM-PC's, UNIX machines, and many other systems that provide an ANSI C compiler.

Bibliography

[1] Knuth Donald E., *TeX the program*, Addison-Wesley 1988.
[2] Knuth, Donald E., *Literate Programming*, Center for the Study of Language and Information, Leland Standard Junior University, 1992.
[3] Krommes John A., *The* WEB *System of Structured Software Design and Documentation for C, C++, Fortran, Ratfor, and TeX*. User Manual, 1993.
[4] Sewell Wayne, *Weaving a Program. Literate Programming in* WEB, Van Nostrand Reinhold, NY 1989.

Appendix

Excerpt from the `tugboat.web`. This will generate two files: `tugboat.cmn` and `tugboat.sty`.

```
@z=======================================
%%
%% Version 1.0
%%
%% W{\l}odek Bzyl, Tomek Przechlewski
%%
%%=======================================
%% original filename="tugboatc.web",
%% version="1.0",
%% date="8-July-1993",
%% filetype="TeX macros for TUGboat",
%% email="Internet:
%%     matwb@halina.univ.gda.pl,
%%     ekotp@halina.univ.gda.pl",
%% keywords="TUG, tugboat, plain tex",
%% abstract="This composite file
%%     contains the plain-based macros
%%     for preparation
%%     of TUGboat converted to WEB".
@x=======================================

%% limbo

...

\def\LaMeX{{\rm L\kern-.345em
  \raise.3ex\hbox{\sc a}\kern-.16em
  M\kern-.111em\lower.6ex\hbox{E}
  \kern-.075emX}}

\def\Wtitle{GUST.WEB}

...

%---------------------------------------

@Lx   @% set the global language to \TeX

@* Identify the version.

@A
@O tugboat.cmn

\def\fileversion{v1.0}
\def\filedate{8 July 1993}

\message{File 'TUGBOAT.CMN'
  \fileversion\space\space <\filedate>}

@* Put in the index commands
   with 'at' inside.

@f '\@@ 11

@* Helpful shorthand.

@*1 Changes of category.
The following allow for easier changes
of category. These require that the
character be addressed as a
control-sequence:
e.g. \verb|\makeescape\/| will
make the \verb|/| an escape character.

@A
\def\makeescape#1{\catcode`#1=0 }
\def\makebgroup#1{\catcode`#1=1 }

...

@O tugboat.sty

...

@* Stop reading this file if
   it's been loaded already.

@A
\ifx\tugstyloaded@@\thistubstyle
  \makeatother\initializearticle
  \endinput
\else
  \let\tugstyloaded@@\thistubstyle
\fi

\message{File 'TUGBOAT.STY'
  \fileversion \space\space <\filedate>}

@* Load macros common to \TeX\ and \LaTeX.

@A
\input tugboat.cmn

@* Some things with the same names
   as in, or reiterated from, \AmSTeX.

@A
@% override an \AmSTeX\ convention
\def\document{}

\output{\output@@}

...
```

Teaching Typography—The Didot Project*

Mary Dyson

Department of Typography & Graphic Communication
University of Reading
2 Earley Gate
Whiteknights, Reading
RG6 2AU UK
ltsdyson@rdg.ac.uk

Abstract

This paper briefly outlines the Didot project on teaching digital typography. A key issue, namely the nature of the interaction between computer and design specialists, is discussed and related to the type of material that could be included in a curriculum for digital typography. Teaching methods and material that have been developed in this area are outlined and an evaluation of one of these methods is described.

1 The Didot Project

1.1 Introduction.

The acronym Didot stands for 'DIgitising and Designing of Type' and this project has been funded as part of the European COMETT II programme. The project started in 1990 and is due to finish in September 1993. The partners come from research centres, academic institutions, commercial organisations and studios in France, Switzerland, Germany, UK, Spain, Greece and Italy.

1.2 Aims.

The aims of the project are to:

- design, implement and evaluate a curriculum for digital typography, designed for both computer-oriented specialists and graphic artists and typographers;
- organise seminars and workshops for both groups; and
- publish and distribute information.

2 Key Issues

2.1 Promoting discussion.

One of the main interests within the project is encouraging discussion between computer specialists and design specialists. Seminars and workshops have encouraged both groups to attend, but there has tended to be a predominance of one or other group at the meetings. Whilst designers have received information about digital techniques, and computer specialists have heard about design issues, we need to question how far we should go in these directions. One of the more obvious problems is interpreting the 'language' of the other discipline. Design concepts are not expressed in the 'normal language' of computer scientists.

2.2 Roles.

Graphic artists and typographers are undoubtedly users of tools of digital typography, and may have something to say about the development of such tools, but they are not normally the developers, as they lack the technical skills. Computer specialists, as developers of the tools of digital typography, tend also to be users. However, they may not have the appropriate design knowledge to make best use of the tools. At a Reading Didot seminar, we discussed some of the options: computer scientists using the tools (with direction from designers) as opposed to designers using the technology themselves.

2.3 Contributions.

The contribution that designers can make to the development of tools was discussed at a summer school in Lausanne. The nature of the tools that are being produced should be influenced by the working procedures adopted by designers. For example, designers may have clear ideas as to what type of work they wish to do on paper, and what can best be carried out on screen.

3 Questions Arising

The above issues can be discussed in terms of three interrelated questions. We need to decide who we should teach, what we should teach and how we should teach.

3.1 Relevant material.

The first two questions need to be considered together in deciding what material is relevant to each of the groups engaged in digital typography. The Didot project focuses on type design, looking at methods for creating

*Published in TUGboat **14.3**, 329–332 (1993).

and drawing characters. However, some of the work at Reading has extended the scope of the project to include how we use typefaces in designing documents. The study of digital typography is therefore relevant to, not only those involved in creating and manipulating fonts, but also users of document preparation systems. However, it may be necessary to distinguish between what is taught to users and what is taught to developers and implementers (cf. Brown and Utting, 1992).

3.2 Curriculum development.

One way to approach this diversity is to design a curriculum for digital typography which can cover a wide range of requirements. The nature of a generic curriculum is discussed by Dyson (1992), who argues that the same topics may be relevant to different disciplines, but the subject matter may need to be treated differently depending on the background of the students.

The subjects that could be included in a curriculum for digital typography have been explored by André and Hersch (1992) who concentrate on the computer science aspect of the subject. They put forward the argument, which is fundamental to the Didot project, that digital typography should not be taught without teaching classical typography. Within the Didot project, this is dealt with in terms of historical and cultural aspects of letterforms and the fundamentals of letterforms and the design of type.

4 Teaching Methods and Material

The Didot project has explored a range of teaching methods and materials in relation to digital typography which include:

- seminars and workshops,
- tools, and
- vacation courses.

4.1 Programme of seminars.

The nature of seminars has varied, depending on the specialisms of the seminar organisers, and also the country where they have taken place. The seminar/workshop in Reading explored ways of introducing people to some of the issues surrounding letterform design and studied digital techniques alongside traditional methods of design and manufacture. The summer school in Lausanne provided a more thorough grounding in technical matters, combined with the cultural, historical and aesthetic aspects of the subject. The Basel seminar built upon the previous seminars and developed and evaluated educational concepts. The French seminars were aimed at graphic designers and provided a means of demonstrating and working with the new technologies. The workshops of the Didot works seminar in Hamburg again focused on digital tools, with the lectures providing a rich design context.

The seminars in Italy and Greece were somewhat different in nature as they highlighted the important role of education within their respective countries. In particular, they raised awareness of the problems of using the tools of digital typography without the necessary background knowledge.

4.2 Local workshops.

In addition to these international seminars, a series of local workshops in Reading have introduced typography to beginners through the three areas of lettering, traditional handsetting and computers. The main objective of the workshops was to explore the relationship between major typographical variables through practical experience of different techniques and tools. Lettering introduces students to the influence of the tool on letterforms. Handsetting allows students to directly manipulate type and space, an experience which can then be translated into the less tangible medium of computer typesetting.

Basic issues of legibility, dealt with in theory classes, were re-examined. The relationships between choice of typeface, type size, interlinear spacing, line length, setting, hyphenation and format were explored in a series of exercises using the computer to set type. The students then evaluated the results of their exercises through conducting empirical tests. These activities were aimed at establishing effective design procedures for digital typography.

4.3 Tools.

As part of the Didot project, specific tools have been developed as teaching material. An interactive program comprised of exercises in character-hinting techniques has been developed at EPFL. At Reading, a hypertext on the subject of document preparation systems has been written to support a series of lectures and is currently being evaluated. As a student project, a video has been made based on material from the Reading seminar. The video explains the process of punchcutting and type manufacture to people with no knowledge of the subject.

4.4 Vacation courses.

Some of the teaching methods used at Reading in relation to the teaching of historical and cultural aspects of letterforms have also been evaluated. Students attend two vacation courses as part of the four year BA(Hons) in Typography & Graphic Communication. One of these is in Northern Europe and the other is in Italy. These courses abroad provide direct experience of the material they are learning about through lectures and seminars in the Department and aim to stimulate interest in the subjects they are studying. The evaluations have looked in particular at working methods, and how useful the students perceive these methods to be. Comparisons have also been made with other forms of teaching, such as lectures, seminars and practical work. The questions have evaluated:

- the type of activities engaged in on the vacation courses;

- forms of preparation;
- sources of information;
- methods of learning;
- methods of recording information; and
- methods of analysis and synthesis.

The results support the use of first hand experience as a means of learning. The courses help with student's understanding of specific issues in theory and history, as well as providing inspiration for practical work. There does however, need to be sufficient preparation before the course and a means of consolidating what has been taught afterwards, to make best use of the time spent in observation and analysis whilst on the course.

5 Conclusions

Digital typography encompasses a diverse range of specialisms and we must consider the balance between teaching the core of the subject and developing and distributing specialist material. The requirements of specific types of users need to be clearly defined and mapped onto a range of appropriate teaching methods and materials.

References

[1] André, J. and Hersch, R. 'Teaching digital typography', EP-ODD, **5(2)**, 79–89, 1992.

[2] Brown, H. and Utting, I.A. 'Teaching EP to computer scientists', EP-ODD, **5(2)**, 91–96, 1992.

[3] Dyson, M.C. 'The curriculum as a hypertext', EP-ODD, **5(2)**, 63–72, 1992.

ANNOUNCEMENT
LaTeX2ε — A New Version of LaTeX

Chris Rowley and Frank Mittelbach
LaTeX3 Project

September 28, 1993

What is it?

LaTeX2ε is a new, standard version of LaTeX—prepared and supported by the LaTeX3 Project Team.

Why is it needed?

Over the years many extensions of LaTeX have been developed. This is, of course, a welcome development, since it shows that the LaTeX system is in a healthy state. It has, however, had one unfortunate consequence: there are now several incompatible systems, in the sense of format (`.fmt`) files, all claiming to be LaTeX. Therefore, in order to process documents coming from various places, a site maintainer needs to provide several format files: LaTeX (with and without NFSS), SLiTeX, AMS-LaTeX, and so on. In addition, when looking at a source file it is not always clear for which format the document was written.

What will it offer?

LaTeX2ε will put an end to this unsatisfactory situation—it will give access to all such extensions based on a single format and thus end the proliferation of mutually incompatible dialects of LaTeX 2.09.

It will use an enhanced version (NFSS2) of the New Font Selection Scheme. Files such as `amstex.sty` (formerly the AMS-LaTeX format) or `slides.sty` (formerly the SLiTeX format) will become extension packages, all working with this single format.

The introduction of this new version will also make it possible to add a small number of often-requested features (such as extended versions of `\newcommand`).

To summarize:

- **Standardisation:** a single format incorporating NFSS2, to replace the present multiplicity of incompatible formats (NFSS, lfonts, pslfonts, etc.)
- **Maintenance:** a standardised system supported by a reliable maintenance policy.

Compatibilty

LaTeX2ε will, as far as possible, adhere to the following principles:

1. Unmodified version 2.09 document files will produce the same output with LaTeX2ε as with version 2.09.
2. All new features of LaTeX2ε will conform to the conventions of version 2.09, making it as easy as possible for current users to learn to use them.

Documentation

The new version will be described in a new edition of Leslie Lamport's *LaTeX: A Document Preparation System*, and in a new book, *The LaTeX Companion* by Michael Goossens, Frank Mittelbach and Alexander Samarin (both to be published by Addison-Wesley)[1]. The *Companion* will also contain a complete description of NFSS2.

Distribution Policy

Maintenance of the new system will be undertaken by the LaTeX3 Project Team.

A complete distribution of all files, incorporating corrections of errors, will be made available *twice a year on fixed dates*. This will happen even if there have been no changes to the files, and hence only the release dates have had to be updated. It will therefore be easy to determine whether you have all the latest bug fixes—information which is essential to system maintainers.

Error Reports

Error reports can be made using a report generating program `latexbug.tex`. This will be part of the main distribution.

Error reports will be accepted only if the version of LaTeX2ε that produced the error is not older than one year. Error reports can be sent to this e-mail address: `latex-bugs@rus.uni-stuttgart.de`.

[1] More information about this book can be found in *Annex $\mathcal{F}$ of this MAPS.*

Availability

The α-test version of LaTeX2$_\varepsilon$ is scheduled for late fall and the first regular distribution will be in early 1994 (says Frank). This will also contain information and advice to help in the process of updating an installation for use with the new version (says Chris).

Note added by TUG

LaTeX3 Update: Coming Soon!

There will be a status report on the LaTeX3 Project coming soon. Depending on which publication gets out first, it will either appear in *TUGboat* 14, #4, or TTN 3,1 (January 1994). Further prototype code for LaTeX3 is taking shape at this very moment (I am informed by reliable sources!). More information is coming!

Review of recent LaTeX books*

Nico Poppelier

Elsevier Science Publishers
Amsterdam
The Netherlands
`n.poppelier@elsevier.nl`

A Guide to LaTeX,
Document Preparation for Beginners and Advanced Users,
Helmut Kopka and Patrick Daly.
Addison Wesley 1993, 436 pages (including indexes),
ISBN 0-201-56889-6.

LaTeX Line By Line,
Tips and Techniques for Document Processing,
Antoni Diller.
John Wiley & Sons 1993, 291 pages (including index),
ISBN 0-471-93471-2.

— — * — —

Whenever I read a book that describes a computer-related topic I ask myself these questions: does this book contain information I haven't seen elsewhere, does it explain things in a way that gives new insights? And I do this especially with books on TeX or LaTeX.

In April 1992 I gave a favourable review of two books about LaTeX, two *good* books, written by Helmut Kopka. However, they were written in German, which made them inaccessible to a large portion of the TeX user community. Fortunately the first of these two books is now available in an English version: *A Guide to LaTeX, Document Preparation for Beginners and Advanced Users*. This version was written by Helmut Kopka and his colleague Patrick Daly at the German Max-Planck-Institut für Aeronomie, and is based on the fourth edition of *LaTeX, eine Einführung*. The English version is not a mere translation, but an internationalized version, where parts specific to the German language have been replaced by descriptions of, e.g., the new font-selection scheme (NFSS) and the Babel system.

As in the German original, Kopka and Daly follow Lamport's basic notion that with LaTeX the user is freed from worrying about the layout while writing a piece of text. *A Guide to LaTeX* has chapters on document and page styles, displayed text, mathematical formulas, pictures, user-defined structures and a few advanced features of LaTeX. In the appendices the authors treat the `letter` document style, including possibilities for customization, BIBTeX, SLITeX, LaTeX extensions, and the CM and DC fonts.

Their book provides a wealth of information, and if the updated English version of the companion volume *LaTeX, Possibilities for Extensions* will be published by Addison-Wesley soon enough, there is practically no need for further books about LaTeX, since almost everything beginning or advanced users need to know is in one of these two books.

In contrast with this, Antoni Diller does not present LaTeX as a system for the production of structured documents. LaTeX is a system that emphasizes structure over presentation, and that is ideally suited for the production of many instances of a certain type (class) of document, for example office memoranda or scientific articles. Instead, in *LaTeX Line By Line* LaTeX is presented as a collection of TeX macros, with which you can achieve all sorts of effects. Therefore, all examples and explanations in the book use a mix of LaTeX commands, plain TeX commands and TeX primitives.

There is of course nothing wrong with this approach, if the book is intended as a book on tips and tricks in LaTeX and plain TeX. However, the preface of *LaTeX Line By Line* clearly shows that the book is intended as a book for novice users, and attempts to explain *all* about LaTeX. In other words: it is intended as 'your first and only book on LaTeX'.

There are no glaring errors or omissions in the book, but it lacks structure, and the mix of LaTeX, plain TeX and TeX primitives will really confuse any novice user. Because of this, *LaTeX Line by Line* is not a book for beginners, even though the author writes in the preface 'This book can be read by someone who has no previous knowledge of either LaTeX or TeX.'

In the introduction of this review column I explained what the things are that I am looking for in a new computer book: what makes this book special or unique? What does it explain that I haven't seen before, or in a way I haven't read before? For what special group of people is it written? Antoni Diller's book on LaTeX is-

*Published in TUGboat **14.2**, 127–128 (1993).

n't special in any sense; it is just another poorly written book about LaTeX, of which there are unfortunately a few too many already. *LaTeX Line by Line* contains a lot of useful tricks, especially in the area of mathematics, but they are presented in an unstructured and confusing way. Also, the title is not appropriate: it is not a book about LaTeX, but a book about how to combine LaTeX, plain TeX and TeX primitives to achieve certain special effects in layout.

A detail: both books give the old address of the TeX Users Group (that is a problem with putting addresses and similar factual information in a book).

Another detail: the book by Helmut Kopka and Patrick Daly has the nicest LaTeX logo I have ever seen – including the one on Lamport's book! No wonder, since the designer at Addison-Wesley called Barbara Beeton and asked her what it should look like. And I must say: she did a very nice job!

The logo on Antoni Diller's book looks horrible, since the 'A' in LaTeX is *much* too far to the right. Not only that, but it is also reproduced like that many times on the front cover. Add to this the poor design of the book and the fact that it was reproduced from low-resolution output, I am afraid that there is another book about LaTeX on the market that I cannot recommend.

The Technical Council and Special Interest Working Groups*

Michael Ferguson

Chair, Technical Council

The primary purpose of the TEX User Group's Technical Council is to provide a forum and mechanism for studying technical issues concerning TEX, Metafont, and their auxilliary support systems. Although this includes such important service, standardisation, future tools projects, as Archives (TWG-Archives: George Greenwade, TWG-Chair), Multiple Language Coordination (TWG-MLC: Yannis Haralambous, TWG-Chair), and Independent Research Projects (IRP-TWG: Alan Hoenig, TWG-Chair) which includes the LATEX3 project (Frank Mittlebach, Project Chair), it is the 'Special Interest Working Groups (SI-TWG)' that will have the most interest for the ordinary TEX user. SI-TWGs are dedicated to the exploitation of TEX systems for the specialised needs of particular user groups. These vertically integrated interest groups are destined to be an incredibly exciting part of the TEX community.

Our first SI-TWG was 'TEX for the Disabled' with T.V. Raman (raman@cs.cornell.edu) as the chair. Raman, who is blind, has created a very impressive system for interpreting the control codes and structures of TEX by means of audio tones. His TWG is not only concerned with making TEX more accessible for the blind, but also for persons with other disabilities. This group has been able to connect people who need and people who produce such things as braille fonts for mathematics, Computerized Books for the Blind, and appropriate online versions of the LATEX manual. There has also been discussion and offers within the group to write some specialized prorams, such as adapting DOS for the vision impaired. This SI-TWG is an exciting beginning to, what I believe, will be a very important part of the TEX environment.

At this moment, we are in the process of organizing two new SI-TWGs. The first is to address the needs of those that are using, need to use, or merely interested in, the coupling of TEX and SGML (Ken Dreyhaupt: dreyhaupt@spint.compuserve.com). This is a very important area for the enhancement of cooperation and understanding between two communities, and will also serve as means of discussion and research for those that must straddle both areas. The other new SI-TWG being organized concerns the relation of TEX and Adobe's new file format Acrobat (Doug Henderson, drh@bluesky.com).

SI-TWG may reflect the specialized technical interests of a particular community such as publishers, scientific organizations, chemists, astronomers, physicists, historians, logicians, theoretical computer scientists, or musicians, or perhaps satsify a need for the, possibly specialized, use of a tool such as LATEX. One particular area might be Human/Computer interfaces for LATEX.

What does it take to start an SI-TWG? First, and foremost, it takes an enthusiastic person with a burning interest. There also should be a user community with similar interests that will come together, usually electronically, for discussions and actions. Each TWG has a chair and a member of the Technical Council, who is sometimes the chair, to act as a liason between the TWG and the TUG Board. The TWG chair is expected to report, usually through the Technical Council Liason. its activities to the Technical Council and hence the board. The chair is also encouraged to report the TWG's activities in any forum that he/she feels beneficial and to seek cooperation with individuals and institutions as appropriate.If there is any action that a TWG would like to report to the board for formal (TUG) recognition, this action or recommendation will be evaluated by the Technical Council, and then brought to the board. It is anticipated that there will be little need for such action by a SI-TWG but the possibility is there.

The TWG chair is completely responsible for the membership and activities of the TWG. The Technical Council and its liason is charged only with reporting the activities of the TWG and carrying recommendations to the TUG Board. If you would like to participate in a particular TWG, please contact the Chair. If you would like to start a TWG, please contact any member of the Technical Council.

Michael Ferguson (Chair),
mike@inrs-telecom.uquebec.ca
Yannis Haralambous,
yannis@gat.citilille.fr
Alan Hoenig,
ajhjj@cunyvm.cuny.edu

*Reprint from the Daily Telegraphs TUG'93 conference, Aston, July 26–30, 1993.

Call for papers: TEP '94

Mary Dyson

TEP94 is a workshop on the teaching of electronic publishing and digital typography. This will be held in Darmstadt (Germany) on 12–13 April 1994 together with the conference on electronic publishing, document manipulation and typography (EP94), the conference on raster imaging and digital typography (RIDT94) and the workshop on principles of document processing (PODP94).

This workshop will be the second of its kind. The first was TEP92 held in Lausanne, Switzerland in conjunction with EP92 where there was some broad ranging discussion of terms such as *Electronic Publishing ng* and *Digital Typography*, consideration of the skills and people involved, and description of tools for teaching the subject. The proceedings have been published as a special issue of the journal Electronic Publishing: Origination, Dissemination and Design.

TEP94 will again be concerned with the issues surrounding teaching in this area. Papers are invited on the following general topics:

- the relationship between theory and practice in teaching electronic publishing
- training versus teaching
- coping with change; what are the constant elements in teaching electronic publishing and what do we now mean by publishing?
- the typography component of digital typography; how do we teach design aspects of the subject, as opposed to software training; how are they combined?

In addition to these general themes, the workshop will focus on the design of multimedia systems and electronic books. Papers here should be concerned with how we teach multimedia systems, rather than how to use multimedia systems for teaching. Some of the issues are:

- whether we should be developing individual specialisms or general skills
- whether multimedia or electronic books are different to other types of electronic publishing
- whether multimedia falls within the domain of the publisher
- what tools are available
- how we teach people to write and design for the media
- the specific use of electronic books within the field of typography or electronic publishing
- can we teach or encourage good design through these systems?

TEP94 programme committee

Mary Dyson	University of Reading, UK (chair)
Jacques Andre	INRIA-Rennes, France
David Brailsford	Univ. of Nottingham, UK
Heather Brown	University of Kent at Canterbury, UK
Henri-Paul Bronsard	College du Vieux Montreal, Canada
Hans-Jurgen Ehlers	Friedrich Alexander Universitat, Germany
Richard Furuta	Univ. of Maryland, USA
Peter King	University of Manitoba, Canada
Roberto Minio	PIRA, UK

Instructions for submission

Papers should be written in English and limited to 3000 words and will be published as separate proceedings in advance of the workshop. The final paper must be accompanied by a floppy disk with the text in either an ASCII file or a Microsoft Word file. Figures should also be provided on disk in TIFF, PICT or EPS format. Authors should contact the TEP94 editor to discuss formats. Special fonts should be used only in figures. All material must be prepared for printing at a high definition (>1200 dpi).

Deadlines

- 30 September 1993
  Full paper received by TEP94 secretariat (in paper form)
- 12 November 1993
  Notification of acceptance or rejection
- 4 January 1994
  Final version received by editor

TEP94 secretariat

Mary Dyson, TEP94 workshop
Department of Typography & Graphic Communication
University of Reading
2 Earley Gate
Whiteknights
Reading, RG6 2AU, UK

Telephone	0734 318084
Fax	0734 351680
e-mail	`ltsdyson@uk.ac.rdg`

Call for papers: TUG '94
Santa Barbara, California

31 July – 4 August

The *15th Anniversary Meeting* of the TEX Users Group will be held in Santa Barbara, California from 31 July through 4 August 1994. Those wishing to present papers must have their titles and outlines submitted to the program committee at `tug94@tug.org` by *February 1, 1994*, completed papers by *May 20, 1994*.

Since TEX and Metafont applications and interests are as varied as our users, we are encouraging papers over the entire range of related topics, with a particular *focus on innovation*. Let us take a fresh look at what we have and envision new areas of use.

TEP94 reply form

```
Last name...........................First name......................
Affiliation.........................................................
Address.............................................................
....................................................................
....................................................................
Tel................................
Fax................................
Telex..............................
E-mail.............................


1.  I wish to attend the TEP94 workshop and receive further information
2.  I wish to submit a paper to TEP94
    Provisional title:..............................................
    ................................................................
    ................................................................
3.  I wish to receive further information on
    RIDT94............PODP94.............EP94............
```

Table of Contents TUGboat

Volume 14.1, 14.2 and 14.3

April 1993 / July 1993 / October 1993

Abstract

TUGboat table of contents files are on `math.utah.edu` in `pub/tex/pub/tugboat`, also accessible via `tuglib@math.utah.edu` server by 'send index from tex/pub/tugboat'.

TUGboat 14.1 (April 1993)

TUGboat 14.2 (July 1993)

TUGboat 14.3 (October 1993)[1]

[1]Proceedings of the 14th Annual Meeting, Aston University, Birmingham UK July 25 – 29, 1993.