Baskerville

pdfmark=/DOCVIEW,PageMode=/UseBookmarks,Page=1,View=/FitH

The Annals of the UK TeX Users' Group **Guest Editor: Carol Hewlett** **Vol. 6 No. 3**

ISSN 1354–5930 **July 1996**

Articles may be submitted via electronic mail to `baskerville@tex.ac.uk`, or on MSDOS-compatible discs, to Sebastian Rahtz, Elsevier Science Ltd, The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, to whom any correspondence concerning *Baskerville* should also be addressed.

This reprint of *Baskerville* is set in Times Roman, with Computer Modern Typewriter for literal text; the source is archived on CTAN in `usergrps/uktug`.

Back issues from the previous 12 months may be ordered from UKTUG for £2 each; earlier issues are archived on CTAN in `usergrps/uktug`.

Please send UKTUG subscriptions, and book or software orders, to Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Fax/telephone: 0121 476 2159. Email enquiries about UKTUG to `uktug-enquiries@tex.ac.uk`.

Contents

I Editorial 2
II Beautiful tables, the easy way with the mdwtab package 3
1 A touch of class 3
2 New column types 4
3 Unboxed tables 5
4 Life's little luxuries 5
4a Footnotes 5
4b Vertical alignment of ruled tables 5
4c Extra row separation 5
III The frequent supplement 6
5 **69a Multilingual typesetting in LaTeX** 6
6 **75a Footnotes in tables** 6
IV An introduction to PSTricks, part 3 8
A Tree diagrams 8
V Java, Java 16
VI The Euromath System 19
VII How does HTML handle mathematics? 20
A Introduction 20
7 SGML Notations 21
B HTML3 (expired draft) 21
8 Summary 25
C Future of maths in HTML 25
D Conclusion 26
VIII Hyper-G: Information—the Next Generation 28
A Introduction 28
B Using Hyper-G for Electronic Publishing 28
C Providing Quick Access 30
D New Publishing Paradigms 30
E Turnaround Time and Cost Effectiveness 31
F Current Electronic Publications With Hyper-G Technology 31

IX Portable Documents: What Next? 33
9 Introduction 33
A What Next? 33
10 Les Carr, University of Southampton 33
B Converting from LaTeX to SGML 33
11 Sebastian Rahtz, Elsevier Science 33
C SGML is here 34
12 Andrew Dorward and Neil Bradley, Pindar 34
D Java – The Krakatoa of the Web 34
13 Henry Rzepa, Imperial College, London 34
X An aged archivist remembers. 36
XI Malcolm’s gleanings 37
A In consistency 37
B MINSE 37
C Times change 37
D *TUGboat* 37
XII Announcement of the Annual General Meeting 38
XIII The UK TeX Users’ Group 40
XIV Obtaining TeX 41

I Editorial

The recent headline in the *Financial Times*, “Hewlett adds to worries in high-techs” referred, of course, to Hewlett Packard (no relation, unfortunately). Some articles in this *Baskerville* are high-TEX but some are no-TEX-at-all, covering the wider areas of electronic publishing and other ways of conveying maths notation.

This *Baskerville* contains a complete report of the *Portable Documents: What Next?* meeting. The summary of the meeting is complemented by the papers on ‘Euromath’ and ‘Hyper-G’ written by the people who gave those papers. The paper on maths and HTML was presented at the UKTUG meeting on 20 March.

On a technical note, I used emTEX, in its ‘huge’ version running on an IBM (Pentium) PC to process the LATEX files, and DVIPS together with GSview (the Windows Ghostscript viewer) for previewing.

This issue of *Baskerville* would not have been possible without the generous help that I received from the past editors and guest editors, Sebastian Rahtz, Robin Fairbairns, David Carlisle and Malcolm Clark: they have my sincere thanks. As there is still no permanent editor for *Baskerville*, you can work out who is likely to produce the next issue by comparing the list of committee members with the names of those who have already taken their turn. I’ll just welcome him with the old cliché, come on in, the water’s lovely! mdwtab

II Beautiful tables, the easy way with the mdwtab package

Mark Wooding

The first thing you'll probably notice when you load the mdwtab package is that TeX runs out of memory that much quicker. The author is well aware that the package is far too large, but can't decide which bits of its functionality can be dropped. The second thing you'll notice is that some table-related packages stop working properly, because they don't understand how mdwtab handles tables, and still think that LaTeX is in control. The author has attempted to retain compatibility with the 'Tools' collection of packages, and in particular with David Carlisle's excellent set of table handling things. The third thing you'll probably notice is that your tables still look (almost) exactly the same as they did before. In common with the array package, rules in tables now contribute to the table's width and height, which fixes a problem with the sides of tables looking uneven.

This is an example	of a ruled
table, using LaTeX's	standard table
handling. Can you see	the slight nicks
in the vertical rules	on the sides?

It's odd starting an article about a package with a list of its drawbacks: the author is of the firm opinion that you should have no illusions concerning the drawbacks of the package. (There are a load of others which I haven't described here; most of them aren't very interesting.)

On the other hand the package *does* provide all the column types and other new features of the array package, including the `\newcolumntype` command.

1 A touch of class

Tables which have horizontal rules tend to look rather better if you insert a little extra space above and below the rules. Compare with to see the difference. Which do you think is nicer?

Package	**Advantage**	**Disadvantage**
Standard LaTeX	Built-in	Ugly tables
array	New column types	It isn't perfect
mdwtab	Beautiful tables	It's very big

hv **Package**	**Advantage**	**Disadvantage**
vhv Standard LaTeX	Built-in	Ugly tables
array	New column types	It isn't perfect
mdwtab	Beautiful tables	It's very big
vh		

The first example above was created using the standard LaTeX `\hline` command. The second example has some extra space inserted around the horizontal rules. This is achieved by using the `\vgap` command. If you say '`\vgap{`*length*`}`', TeX will insert a 'short' row whose height is the *length* given, ensuring that the vertical rules in your table are extended appropriately. For example, I just said `\vgap{2pt}` at appropriate points in the table above.

However, this isn't always what you want. If you use `\multicolumn` commands for headings, you'll get odd-looking 'stubs' as in the following example: Please bear in mind that this *could* be the effect you want (for instance, the `MIX` word-layout diagrams in *The Art of Computer Programming* do this sort of thing).

hv Item one	Item two	
vhv Item three	Item four	Item five
vhv	Item six	Item seven
vh		

In the event that you don't like this effect, and I can't blame you if you don't, you can suppress the rule stubs in certain columns by saying '`\vgap[`*columns*`]{`*length*`}`' – the *columns* argument contains a list of column numbers whose rules are to be omitted. The rules on the very left hand side of the table are numbered 0, while the rules on the right hand side of column *n* are numbered *n*. Column numbers are separated by commas, and column ranges can be given. For example, to suppress rules in columns 1, 2, 3, 5, 7, 8 and 9 you'd say '`1-3, 5, 7-9`'.

In the example above, if I end my table rows like this:

```
...                  \hline \vgap[2]{2pt}
... \\ \vgap[2]{2pt} \hline \vgap   {2pt}
... \\ \vgap   {2pt} \hline \vgap[1]{2pt}
... \\ \vgap[1]{2pt} \hline
```

then it ends up looking much nicer:

hv[2] Item one	Item two	
v[2]hv Item three	Item four	Item five
vhv[1]	Item six	Item seven
v[1]h		

Isn't that neat?

All of this gets terribly cumbersome to type. The `\hlx{`*hlx-commands*`}` command provides a neat abbreviation. The *hlx-commands* argument is a list of single letter commands to perform. There's a load of commands provided, for doing various little jobs:

`h` is equivalent to `\hline`. If you type two `h` commands in a row, a space is left between them, as usual.

`v[`*columns*`][`*length*`]` means exactly the same thing as `\vgap[`*columns*`]{`*length*`}`, except that the *length* argument is optional. If you omit it, the default value of `\doublerulesep` is used, which is usually set to 2 pt.

`c{`*columns*`}` is equivalent to `\cline{`*columns*`}`. You can specify the columns by giving comma separated column numbers and ranges, just as for `\vgap`: the `\cline` command has been upgraded to understand these more complex descriptions.

`s[`*length*`]` leaves a vertical gap of height *length*. If you omit the *length*, the value of `\doublerulesep` is used.

I ought to come clean now. Since I'm a lazy typist, I didn't actually use the `\vgap` command in the tables above. What I actually said was `\hlx{hv}` for the first row, `\hlx{vhv}` for the middle rows, and `\hlx{vh}` after the last one.

2 New column types

It seems to be traditional to add new column types when LaTeX's table handling gets upgraded, and the author saw no reason not to follow the trend. The complete list of column types, and other funny characters that can be used in the argument of the tabular environment, is given below. The new ones added by this package are marked with a little '*'.

`l` Left aligned text (in tabular) or equation (in array).

`c` Centred text (in tabular) or equation (in array).

`r` Right aligned text (in tabular) or equation (in array).

`Ml`, `Mc` and `Mr`* Left, centre and right aligned equations.

`Tl`, `Tc` and `Tr`* Left, centre and right aligned text.

`p{`*width*`}` Top aligned paragraphs, with the given width.

`m{`*width*`}` Vertically centred paragraphs, with the given width.

`b{`*width*`}` Bottom aligned paragraphs, with the given width.

`#{`*pre*`}{`*post*`}`* User defined column type: *pre* is inserted before the table cell's text, and *post* is inserted afterwards.

`|` Inserts a vertical rule between columns.

`!{`*text*`}` Inserts *text* between columns, treating it just like a vertical rule.

`@{`*text*`}` Inserts *text* in place of the usual intercolumn space.

`>{`*text*`}` Inserts *text* just before the cell's text.

`<{`*text*`}` Inserts *text* just after the cell's text.

`*{`*count*`}{`*preamble*`}` Inserts *count* copies of the *preamble* into the table preamble.

You can define your own new column types by saying '`\newcolumntype{`*type*`}[`*narg*`][`*opt*`]{`*text*`}`', which defines a new column type *type*, which means exactly the same thing as the preamble characters *text*. The column type can take arguments (even optional ones) – this works in exactly the same way as `\newcommand`.

3 Unboxed tables

Normally LaTeX will wrap tables up in a box. This makes things convenient sometimes, but horizontal positioning can be a bit of a pain. As well as the usual `[t]`, `[c]` and `[b]` position arguments, the package adds `[L]`, `[C]` and `[R]`, which position the table left-aligned, centred and right-aligned respectively. Such tables are called *unboxed* tables, because they're not wrapped up in a box. As well as allowing you to control horizontal position more easily, such tables have some other advantages.

You can pause an unboxed table for a bit and insert some normal paragraph text. When you say `\tabpause{`*text*`}` in the middle of a table, LaTeX inserts the *text*, typeset in paragraph mode, in the middle of the table. The text can be split across pages and all the normal things like that.

4 Life's little luxuries

As well as perhaps not looking as glorious as they might do, LaTeX's tables have a few other rough edges. The mdwtab package tries (and by and large succeeds) in smoothing these off and tucking all the nastiness under the carpet.

4a Footnotes

LaTeX doesn't allow footnotes in tables. They just don't work: the footnote text mysteriously vanishes. This package will carefully handle footnotes in both boxed and unboxed tables, ensuring that they appear in the right place. (This is done by using a trimmed down version of the author's footnote package, which tries to provide a general solution to the problems of footnote handling.) You can therefore use footnotes in your tables with abandon, and expect everything to work beautifully.

4b Vertical alignment of ruled tables

It's been pointed out numerous times that having rules in tables can make top- and bottom-aligned tables look rather odd: the baseline of the text tends to be lined up with the rules in the tables, rather than with the actual first or last rows. In other words, you get strange results like and . Clearly this isn't terribly desirable.

An	odd
looking	table

An	odd
looking	table

Various solutions have been proposed for this problem. *The LaTeX Companion* describes a pair of commands `\firsthline` and `\lasthline` which provide a workaround. The mdwtab will calculate the height of the rules and other material at the top or bottom of the table and shift it into the right position. Hence you can have and with no extra effort at all.

A	nice
looking	table

A	nice
looking	table

4c Extra row separation

In maths, it's conventional to insert a little extra space between the rows of an array. To save you having to end every line of an array with something like '`\\[\jot]`', extra space of the amount `\arrayextrasep` is inserted automatically. (This parameter is set to 1 jot by default.) There's an analogous `\tabextrasep` parameter, although this is initially 0 pt, and isn't likely to be changed.

If you're worried about matrices looking rather odd as a result of this extra space, don't be. In the author's opinion, they actually end up looking slightly nicer as a result. However, you can almost certainly get better results by using a dedicated matrix environment which takes extra-special care over the spacing.

faqa

III The frequent supplement

Robin Fairbairns

A couple of questions answered this issue; this arises because the column failed to make its appearance in *Baskerville* 6.2 for lack of space.

The first (on multilingual typesetting) was posed by Allan Reese; what we have here is merely a start towards solving the problem that concerns him. A comprehensive review of what makes up the Babel system is due in a future part of David Carlisle's LaTeX tour, but I suspect that even so, questions remain to be answered. Do feel free to send them in!

5 69a Multilingual typesetting in LaTeX

TeX itself defines a mechanism to define a 'language', which provides a hook for hyphenation patterns: each 'language' (in TeX's sense) has its own set of hyphenation patterns.

Babel, Johannes Braams' multilingual package for LaTeX, uses the primitive TeX concept of 'language' as part of its own concept of language; Babel's 'language' also encompasses culturally appropriate typesetting rules for the language; for further details, see Babel's documentation.

Assuming that you have a current LaTeX installation, creating a multiple-language version from it involves three steps:

1. Install babel (`macros/latex/packages/babel`); move the `.sty` and `.ldf` files somewhere appropriate. The installation also generates `hyphen.cfg` and `language.dat`; these must be around when you generate the new LaTeX in step 3.
2. Edit `language.dat` to record the languages whose hyphenation patterns you want to support. For example, my installation's version says:

   ```
   english   hyphen.tex
   UKenglish ukhyph.tex
   french    f8hyph.tex
   ```

   (the upper-case 'UK' is one of Babel's little quirks: one can also call the language 'british'; whatever, it's a different language as far as TeX is concerned).
3. Generate a new version of LaTeX, from the same `latex.ltx`, etc., that came with your current installation. Since the `hyphen.cfg` is 'around', it will be used in preference to the `hyphen.ltx` which is generated when LaTeX is unpacked. The LaTeX format you've built will have the hyphenation patterns installed.

Of course, your TeX has to be big enough to hold all these hyphenation patterns. I've not encountered problems with the *Web2c*-based Unix installation I use at work, but the emTeX I have on my PC at home isn't, by default, large enough. Using *htex386*, I increase the pattern memory by adding the line:

```
set emtexopt=/mt25000
```

to my `autoexec.bat`

6 75a Footnotes in tables

The standard LaTeX \footnote `command doesn’t work in tables; the table`traps the footnotes and they can't escape to the bottom of the page.

If your table is floating, your best bet is (unfortunately) to put the table in a minipage environment and to put the notes underneath the table, or to use Donald Arseneau's package `macros/latex209/contrib/misc/threeparttable.sty`

Otherwise, if your table is not floating (it's just a 'tabular' in the middle of some text), there are several things you can do to fix this.

1. Use `mdwtab.sty` from the directory `macros/latex/contrib/supported/mdwtools`, which is described in Mark Wooding's article above.

   It handles footnotes properly, quite apart from its ways of "increasing the beauty" of your tables.

2. Use \footnotemark `to position the little marker`appropriately, and then put in \footnotetext `commands to fill in`the text once you've closed the tabular environment. This is described in Lamport's book, but it gets messy if there's more than one footnote.
3. Stick the table in a minipage anyway. This provides all the ugliness of footnotes in a minipage with no extra effort.
4. Use threeparttable (`macros/latex209/contrib/misc/threeparttable.sty`) anyway; the package is intended for floating tables, and the result might look odd if the table is not floating, but it will be reasonable.
5. Use `tabularx` or `longtable` from the LaTeX tools distribution (`macros/latex/packages/tools`); they're noticeably more inefficient than the standard tabular environment, but they do allow footnotes.
6. Grab hold of `footnote.sty` from CTAN, also lurking in `macros/latex/contrib/supported/mdwtools`.

   Then put your tabular environment inside a savenotes environment. Alternatively, say \makesavenoteenvtabular `in`the preamble of your document, and tables will all handle footnotes correctly.

IV An introduction to PSTricks, part 3

Sebastian Rahtz
Elsevier Science Ltd
Email: s.rahtz@elsevier.co.uk

A Tree diagrams

PSTricks has an extremely rich environment for drawing trees, which allows for very complex structures and presentation. The available commands are listed in Table 1 and the graphical parameters which apply especially to these are listed in Table 2. As one might expect, most other commands and parameters are also available, from both the generalized drawing, and the node connectors and labels. Each of the node types described earlier is turned into a 'tree' node, and named by prefixing it with a 'T' and removing the 'node' suffix.

Table 1: PSTricks tree drawing commands

\pstree{node}{subtrees}	draws a node and subtrees connected to it
\psTree{rootnode} subtrees	an 'environment' form of \psTree
\Tn	null tree node
\tspace{dim}	leave gap of *dim* before next level
\TC*[settings]	tree node like \Cnode node
\TR*[settings]{something}	tree node like \Rnode node
\Tcircle*[settings]{something}	tree node like \circlenode node
\Tc*[settings]{dim}	tree node like \cnode node
\Tdia*[settings]{something}	tree node like \dianode node
\Tdot*[settings]	tree node like \dotnode node
\Tf*[settings]	tree node like \fnode node
\Tfan*[settings]	draws a triangle with a top corner of the predecessor node
\Toval*[settings]{something}	tree node like \ovalnode node
\Tp*[settings]	tree node like \pnode node
\Tr*[settings]{something}	tree node like \rnode node
\Ttri*[settings]{something}	tree node like \trinode node
\skiplevel*[settings]{nodes or subtrees}	miss out entire levels in a particular subtree
\skiplevels*[settings]{n} {nodes or subtrees}	skip *n* levels

Table 2: PSTricks Graphical parameters for trees

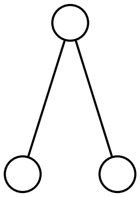
Parameter	*Default*	*Explanation*
bbd=dim		set lower bounding box to *dim*
bbh=dim		set upper bounding box to *dim*
bbl=dim		set left bounding box to *dim*
bbr=dim		set right bounding box to *dim*
edge=command		the node connector used to join tree nodes
fansize=dim	1cm	size of base for \Tfan tree node
levelsep=*dim	2cm	the distance between successive levels in a tree; the * makes the dimension be *in addition* to the size of the nodes (levels are normally a fixed distance apart)
showbbox=true/false	false	draw a dotted frame showing the enclosing rectangle of trees
thislevelsep=*dim		like *levelsep* but applies only to the current tree

PSTricks Graphical parameters for trees *cont.*

Parameter	*Default*	*Explanation*
thistreefit=tight/loose		like *treefit* but applies only to the current tree
thistreenodesize=dim		like *treenodesize* but applies only to the current tree
thistreesep=dim		like *treesep* but applies only to the current tree
tndepth=dim		the minimum depth of tree node labels
tnheight=dim		the minimum height of tree node labels
tnpos=l/r/a/b	b	the position of tree node labels relative to the node (left, right, above, below)
tnsep=dim		the gap between tree node labels and the node (by default the same as *labelsep*
treefit=tight/loose	tight	if tight, *treesep* is the minimum distance between nodes on any level; i f loose, *treesep* is the distance between the enclosing bounding boxes of subtrees
treeflip=true/false	false	does a mirror image of the free, flipping the nodes
treemode=R/L/U/D	D	the direction of tree growth (right, left, up and down)
treenodes=dim	−1pt	if positive, this sets a fixed size for tree nodes, regardless of content
treesep=dim	0.75cm	the distance between successive nodes in a tree
xbbd=dim		increase lower bounding box by *dim*
xbbh=dim		increase upper bounding box by *dim*
xbbl=dim		increase left bounding box by *dim*
xbbr=dim		increase right bounding box by *dim*

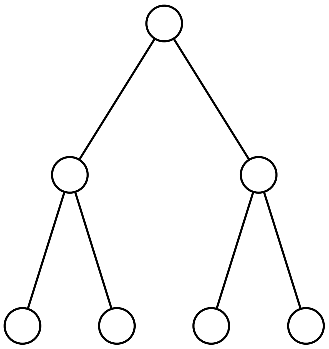
The fundamental concept in PSTricks trees is the *nesting* of trees; a simple tree consists of a root, and one or more nodes:

```
\pstree{\TC}{\TC\TC}
```



but each node can itself be a tree:

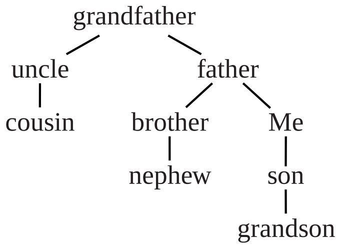
```
\pstree{\TC}{\pstree{\TC}{\TC \TC}
 \pstree{\TC}{\TC \TC}}
```



This simple construct allows very complicated structures to be erected, as the examples below show.

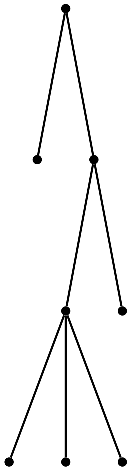
The following tree is a version of that shown in the *LaTeX Companion*, section 10.5.2, which was drawn using the *ecltree* package; the ease of notation is roughly similar. As often in LaTeX, the readability depends a great deal on how the code is laid out. The only change to the defaults is to lessen the vertical space between trees, and add some extra space around nodes.

```
\pstree[nodesep=2pt,
  levelsep=20pt]{\TR{grandfather}}
{
 \pstree{\TR{uncle}}{\TR{cousin}}
 \pstree{\TR{father}}
   {
   \pstree{\TR{brother}}{\TR{nephew}}
   \pstree{\TR{Me}}
     {
      \pstree{\TR{son}}{\TR{grandson}}
     }
   }
}
```



If we now consider another tree drawing package described in the *LaTeX Companion*, Vanroose's *trees*, the example in section 10.2.3 is a little harder to reproduce. The skeleton is trivial:

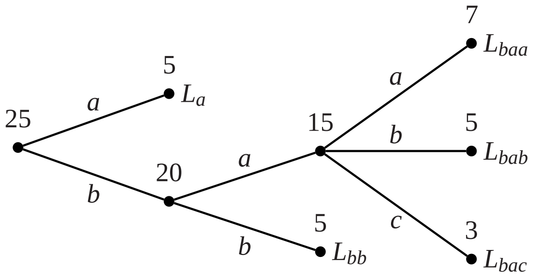
```
\pstree{\Tdot}
{
\Tdot
\pstree{\Tdot}
 {
 \pstree{\Tdot}
  {
   \Tdot
   \Tdot
   \Tdot
  }
  \Tdot
 }
}
```



but when we come to add in all the node and connector labels, and change a few parameters to make the result nicer, the markup becomes a little complex, though the quantity is roughly similar to that of Vanroose:

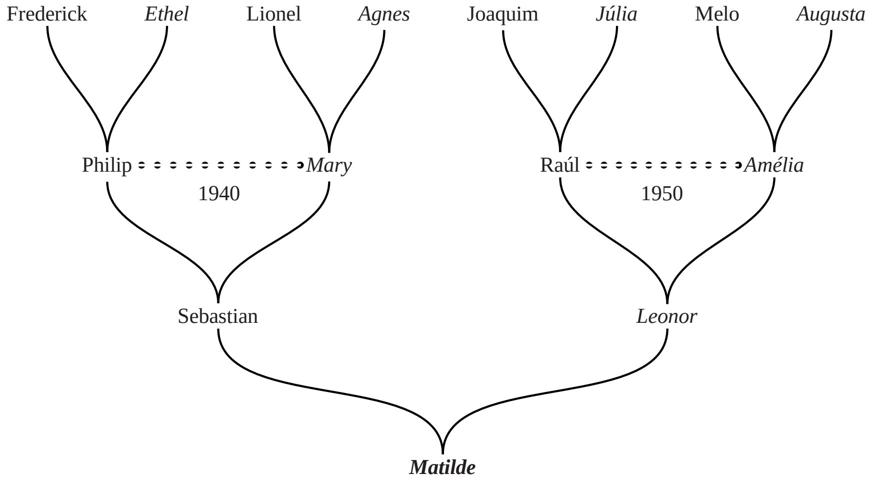
```
\psset{labelsep=2pt,tnpos=a,radius=2pt}
\pstree[treemode=R]{\TC*~{25}}
{
\TC*~{5}~[tnpos=r]{$L_a$}
\taput{$a$}
\pstree{\TC*~{20}\tbput{$b$}}
```

```
 {
 \pstree{\TC*~{15}\taput{$a$}}
  {
   \TC*~{7}~[tnpos=r]{$L_{baa}$}\taput{$a$}
   \TC*~{5}~[tnpos=r]{$L_{bab}$}\taput{$b$}
   \TC*~{3}~[tnpos=r]{$L_{bac}$}\tbput{$c$}
  }
  \TC*~{5}~[tnpos=r]{$L_{bb}$}
  \tbput{$b$}
 }
}
```



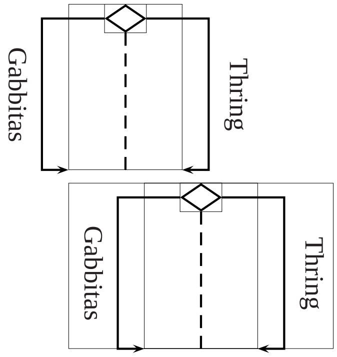
The node connectors in a tree are created by running the macro `\psedge` with the two nodes; the definition of `\psedge` can be overridden explicitly by a redefinition, or by using the *edge* parameter. Here we redefine `\psedge` to be a curve, arrange the angles (bearing in mind that the tree is to grow upwards), and obtain a pleasing result. Note also the explicit links between named nodes, as well as the regular connections.

```
\footnotesize
\def\psedge{\nccurve}
\newcommand{\Female}[2][]{\TR[#1]{\emph{#2}}}
\newcommand{\Male}[2][]{\TR[#1]{#2}}
\psset{nodesep=2pt,angleA=90,angleB=-90,unit=.6cm}
\pstree[treemode=U]{\Female{{\bfseries Matilde}}}{
 \pstree{\Male{Sebastian}}{
  \pstree{\Male[name=P]{Philip}}{\Male{Frederick}\Female{Ethel}}
  \pstree{\Female[name=W]{Mary}}{\Male{Lionel}\Female{Agnes}}}
 \pstree{\Female{Leonor}}{
  \pstree{\Male[name=R]{Ra\’ul}}{\Male{Joaquim}\Female{J\’ulia}}
  \pstree{\Female[name=A]{Am\’elia}}{\Male{Melo}\Female{Augusta}}}
}
\psset{doubleline=true,linestyle=dotted}
\ncline{P}{W}\nbput{1940}
\ncline{R}{A}\nbput{1950}
```



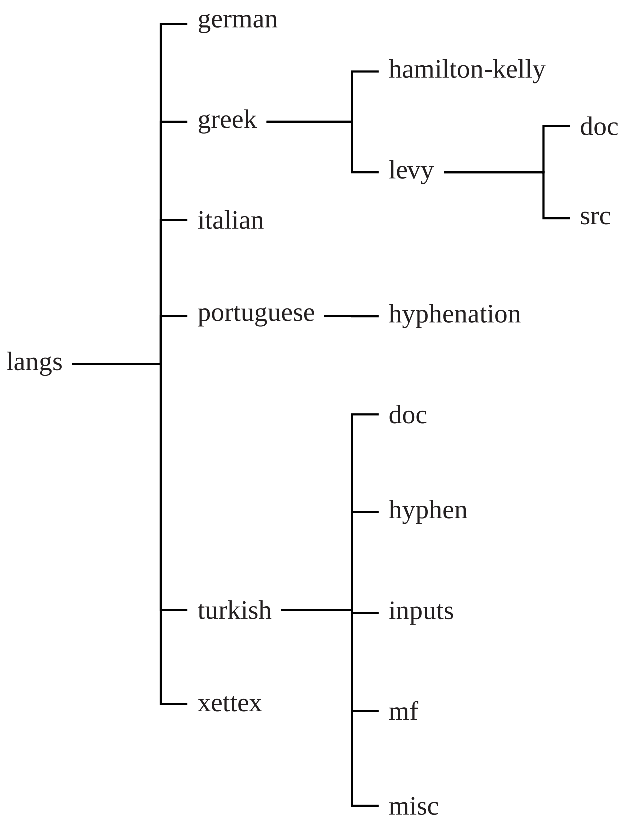
We said earlier that PSTricks does not always work out the extent of objects correctly, and this is illustrated by the connectors in the following example. Turning on *showbbox*, we can see that without the use of *xbbl* etc in the second incarnation, the bounding box is not correct:

```
\psset{angleB=-90,arrows=->,nrot=:U}
\def\molesworth#1{%
 \pstree[#1]{\Tdia{ }}
 {
  \Tp[arrows=->,edge={\ncbar[angleA=180]}]
  \nbput{Gabbitas}
  {\psset{linestyle=dashed,arrows=-} \Tp }
  \Tp[arrows=->,edge={\ncbar}]
  \naput{Thring}
 }
}
\psset{showbbox=true}
\begin{tabular}{l}
\molesworth{}\\[10pt]
\molesworth{xbbl=1cm,xbbr=1cm}
\end{tabular}
```



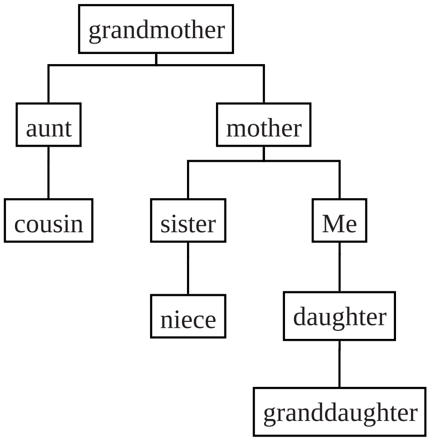
The technique of redefining edges is also necessary if we want right-angled joins, rather than straight lines. Again, we need to take care of the *angleA* and *angleB*, and ensure that in this left-right tree the nodes all line up on their left edges, using the *ref* parameter. If this is not done, the `\ncangle` edges produce strange results.

```
\def\Item#1{\Tr[ref=l]{%
 \psframebox[linestyle=none]{#1}}}
\def\psedge{\ncangle}
\psset{xbbd=1.5cm,treemode=R,
  angleB=-180,angleA=0,levelsep=72pt}
\pstree{\Item{langs}}{%
  \Item{german}
  \pstree{\Item{greek}}{%
    \Item{hamilton-kelly}
    \pstree{\Item{levy}}{%
      \Item{doc}
      \Item{src}
       }
    }
  \Item{italian}
  \pstree{\Item{portuguese}}{
    \Item{hyphenation}
   }
  \pstree{\Item{turkish}}{%
    \Item{doc}
    \Item{hyphen}
    \Item{inputs}
    \Item{mf}
    \Item{misc}
   }
\Item{xettex}
}
```



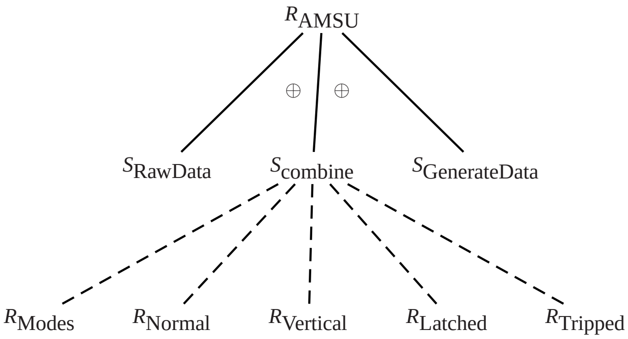
This allows us to remake the small family again, but this time with nicer connectors:

```
\def\XX#1{%
 \Tr{\psframebox{\rule{0pt}{9pt}#1}}%
}
\def\psedge{\ncangle}
\psset{angleB=90,angleA=-90,
   levelsep=36pt,armB=14pt}
\pstree{\XX{grandmother}}
{%
\pstree{\XX{aunt}}{\XX{cousin}}
\pstree{\XX{mother}}
   {
 \pstree{\XX{sister}}{\XX{niece}}
 \pstree{\XX{Me}}
    {
    \pstree{\XX{daughter}}
    {\XX{granddaughter}}
  }
 }
}
```



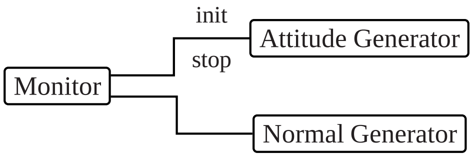
The node connectors are, of course, standard PSTricks objects, so they obey all the normal parameters; here we draw the lower part of the tree with dashed lines, and each node content is set in math mode.

```
\footnotesize
\psset{nodesep=2pt}
\def\XX#1#2{%
 \TR{\ensuremath{#1_{\mbox{#2}}}}%
}
\pstree[xbbr=1.5cm]{\XX{R}{AMSU}}
 {
  \XX{S}{RawData}
  \pstree{\XX{S}{combine}
          \trput{\ensuremath{\oplus}}
          \tlput{\ensuremath{\oplus}}}
   {
   \psset{linestyle=dashed}
   \XX{R}{Modes}
   \XX{R}{Normal}
   \XX{R}{Vertical}
   \XX{R}{Latched}
   \XX{R}{Tripped}
   }
  \XX{S}{GenerateData}
 }
```



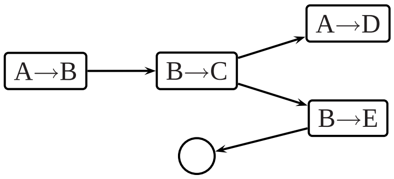
If we want to hang distinct-looking trees off one node, the *ncangle* connector, with some offsets, produces the right result.

```
\psset{framearc=.2,levelsep=4cm,
  armB=1cm,angleB=-180}
\def\psedge{\ncangle}
\def\TreeBox#1{\Tr{\psframebox{#1}}}
\pstree[treemode=R]{\TreeBox{Monitor}}
{
 \psset{offsetA=4pt}
 \TreeBox{Attitude Generator}
 \naput[npos=2.5]{{\small init}}
 \nbput[npos=2.5]{{\small stop}}
 \psset{offsetA=-4pt}
 \TreeBox{Normal Generator}
}
```



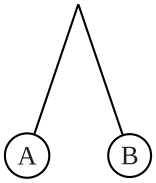
With simple connectors, we do not need to worry about the offsets or angles; what this example shows is how nested trees can change direction:

```
\psset{arrows=->,framearc=.2}
\def\Treebox#1{%
 \Tr{\psframebox{#1}}
}
\pstree[treemode=R]
{\Treebox{A$\rightarrow$B}}{
\pstree{
   \Treebox{B$\rightarrow$C}
         }{
 \Treebox{A$\rightarrow$D}
\pstree[treemode=L]
 {\Treebox{B$\rightarrow$E}}
 {\Tn\TC[arrows=<-]}
      }
}
```



Finally, let us not forget the simple empty node:

```
\pstree{\Tp}{
 \Tcircle{A}
 \Tcircle{B}
}
```



V Java, Java

Malcolm Clark

January 30th, Queen Elizabeth II conference centre, just off Parliament Square, high security at the door, thousands of suits (so that's what happened to all those delayered IBM guys), few women, even fewer pony-tails. We start 20 minutes late (surely not technical difficulties? no, it appears we're letting the late comers get through security and up to this floor). Dry ice in the hall, synchronised flashing lights (is this a disco, or an awards presentation?), sub-Star Wars music. The audience of a thousand are reverentially hushed except for the bleeping of personal telephones. The lights go down, the music up; Robert Young-Johns (one of Sun's VPs) bounds onto the stage, wearing the Java uniform – a cute black shirt with all the right logos. Good move: they are clearly not suits, but still projecting an 'acceptable' corporate identity. He apologises for the hype: 'Java will not cure all known ills', but appeals to our imagination, citing an examples from home banking (yawn), Dolly Parton and Jupiter.

Sun now score a small point by managing to convince the Minister for Science and Technology (did you know we had one?), one Ian Taylor MP, to start the conference off. Apart from an interesting political incorrectness (perhaps understandable, he is, after all a Tory minister), he was adequate. At least he seemed aware of the Internet, and he was also aware that the digital revolution needs to get out to rural areas too, citing the freeing up of radio spectra which could offer ISDN-like data transfer rates. He was also looking forward to a future of smart cards and kiosks (and even interworking between government departments). He stayed for the next talk then strode off with his apparatchik hot on his heels.

Bob Sproull next: this is more like it. This man is seriously worth listening to (Newman & Sproull, Principles of Interactive Computer Graphics). We knock through a wave diagram (the waves are mainframes, minis, pc plus proper user interface, and the network: they are as much focus shifts as waves). The man says 'paradigm'.[1] I like him. He tackles some of the worries: does the Internet have stable funding? In his view it has more stable funding than some of the governments who once funded it. They could disappear tomorrow, but the Internet would still be there. What's wrong with the Web? He cites its lack of interactivity (sic), extensibility, and security, but lauds its 'publish once, read anywhere' democratisation. What's the answer? Java of course. Java applets (little applications) can be downloaded and can provide the interactivity we long for. The interaction is at the client end, so we aren't shipping information backwards and forwards from client and server. Ah you say, I don't like the idea of downloading a program over the Internet to run on my machine; that's just asking for trouble. Obviously the Java team thought about this. What happens is that the applet runs on a Java virtual machine. You get your bit of code (platform independent) and the Java virtual machine interprets it. That way you can insist that the applet has access to only a very few facilities: it should 'do no harm'. For example, applets cannot write to files on the client, and cannot access arbitrary bits of memory or files (your credit card details are safe). There's a bunch of other stuff to ensure that this is going to be secure, some bound in the structure of the programming language, and some in its operation. The Java language specification is, in normal Sun style, open: the specifications are published. They intend to offer 'compatibility' tests and performance guidelines (claiming along the way that interpreted Java can be as fast as compiled C++). In this way they hope to avoid the Balkanisation that is Unix (his phrase). Later, I asked Sproull how confident he was that a similar 'Balkanisation' would not dismember Java, wondering what would happen when Netscape or Microsoft decided to add their own extensions (think of HTML, for example). He said he was optimistic rather than confident, but that there was clearly an advantage to a level of *de facto* standardisation brought about by a high level of interoperability, really hoping that the momentum was sufficient to prevent too much deviation.

The history of Java is interesting: it was not envisaged as an Internet killer app, but as a way of controlling domestic appliances from toasters to VCRs. Or maybe even your doorbell (if I go back to a talk I heard recently by Andy Hopper of Olivetti Research Laboratories). In passing you can begin to see why those potential 4,294,967,296 IP addresses start to look limiting. We don't need one for each individual on the planet: we need one for each connected device on the planet (or off the planet: no doubt Galileo could have had an IP address, just as the Shuttles do). It was designed to be a language for embedded systems, imposing the constraint that it had to be pretty compact. Sun have been maintaining

[1] You can easily identify those who flowered in the 60s and 70s: they know how to say 'paradigm', and aren't afraid to.

that the network *is* the computer for some time. The network really is essential to the workings of this computing. Given its platform independence (or 'neutrality', as the current term seems to be), Sproull could claim that we were moving to a 'write once, run anywhere' range of apps.

After coffee (I would have prefered to say 'a steaming cup of Java', but no, imagination did not stretch to the caterers), Oliver Morton of *Wired* took the stand.

This was a surprisingly erudite, literate and interesting talk. I hadn't really expected a journalist (even one at *Wired*) to be like this Pony-tail. He talked about something I understood – the Cambrian period. His suggestion was that there is a parallel between evolution of lifeforms and evolution of programs. He suggested that it was back in the Cambrian when diversity exploded, and that we date biological variation from then. He suggested that the parallel 'event' now is networking, which will lead to the same sort of explosion of riches and exploitation of 'niches'. I don't really buy argument by analogy (a viewpoint which goes back a long way), but the talk was emminently enjoyable, plausibly relevant, and the AV work was excellent! Instead of fixed 'slides' prepared in some PowerPoint-clone, where the excitement lies in the dissolves and fades, this was a slick (non-perjorative) presentation, where the text on the screen moved around in a way which contributed and enhanced what he had to say. It wasn't really multimedia, but it showed a real quantum leap from 'ordinary' presentations. Of course, this stuff is a blow to me. Usually I scribble down what's on the slides and then go back to listening to the talk. Not possible.

Matt Reid, another Sun manager took us on 'The Java experience', bringing to the podium specimen Java developers. This array of talent included Charles Ashley (Matrix Publishing), Alan Slater (Orbital Technologies), Per Gunnar Osteby (Applet UK), Alan ?? (Knowledge Media Institute – the Open University's newest venture) and Gary Bullock (Parallax). A good range of things in there, from Web enhancement through to good interactive applications, telepresence and the word 'Javatise'. Then we wheeled on a star: Miko Matsumura from *Hot Wired*. He told us he had already addressed audiences on Madison Avenue and Silicon Valley, so we knew he thinks he's good: he probably is too. There is something about these assured 20-something year olds, so completely in control, and at the leading edge which makes me want to retch. Envy I think it's called. The example he showed included an 'avatar'. That's interesting, since it suggested he'd read Neal Stephenson's *Snow Crash*, which I predict is an excellent predictor for much cyberspace development. He ended by telling us he is the first Java evangelist. I reached for my bell, book and candle.

The afternoon split into three streams. I took the stream which contained my first jolt of Java and more Bob Sproull. Simon Roberts, from Sun's Educational Services taught us how to write an application in the Java language in just 45 minutes. No, that doesn't mean it takes only 45 minutes to learn Java. It just means he led us through it, assuming that we were happy with C++ already (I wish!). Java borrows its syntax mainly from C++, but just to confuse there are some syntactic differences: it is largely object oriented, and where it differs most from C++ is in any part of that language which lets you effect strange and unsocial things – like pointers and overloading of operators. It also simplifies by permitting only single inheritance. I ended up more confident that Java was quite coherent and had a lot of thought hours behind it. It also seems to be the programmers' revenge on the Internet. Let me explain. In the good old days, the Internet was the preserve of the hacker (again, in a non-perjorative sense); you had to know a bit about networks, be able to incant in Unix, have a smattering of some obscure programming language (C, C-shell, grep, Perl or whatever). This was when the Internet was free and easy, when the Acceptable Use Policy said 'no commercial use', when people shared programs, and all was sweetness and light (this is a nostalgic view). Then along came the Web with its trivially simple HTML. Anyone could use the Internet, and anyone could provide information for it. And at the same time, it exploded into commerce, politicians started pontificating, and frankly, it got rather non-elitist! Along comes the next step (apologies to Steve Jobs!) – Java. This will separate the men from the boys. Only the traditional propellor heads will be writing the code, but yet, it's what the Net needs. Control has been restored to the anarchists. Well, maybe. In fact I suspect we'll see application environments where most people can create Java applets without getting their hands dirty. But for a while the programmers will be leading.

Back to Sproull: he's talking about Commerce and Security. To be honest, he isn't really into commerce, but he dutifully covers the areas he thinks are going to develop by breaking commercial uses of the net into four areas: selling and advertising (for example, find the product: this can be exemplified by the use of directories); make the sale (here there is great weight on the security of transactions and the confidence with which payments may be made); fulfillment (where orders might be filled entirely digitally, and where standing orders might also be arranged entirely automatically); customer service (which might expand into relationship management where the supplier advises the customer of other related products or services, or upgrades, enhancements, etc.). In other words we are looking for added value, as we would expect.

Security is one of the major issues on the Internet. There is a lot of loose talk about how insecure it may be, usually based on anecdotal evidence which evaporates when tracked down to its source. But taking it further, to the prospect of downloading an applet to run on your machine, needs careful thought. As I have commented already, considerable thought has gone into Java to make it secure. Sproull identified a number of areas that we think about within the general heading of security. We think of ‘benign’ programs: those that ‘do no harm’. The language restrictions of Java are supplemented by the fact that the Java virtual environment checks the byte code of the downloaded applet to ensure that it has not been tampered with en route. Add to this that an applet can only call another applet from the server which provided the calling applet, and, in fact, may only contact that source server and no other. There is also a security manager on the client which restricts access to the client’s resources. If you are dealing with a ‘trusted’ server, the browser’s/client’s policies may be relaxed. For example, if you were running an Intranet within a department or enterprise, you could conceivably allow the applet to read files and ship the details around. Spreadsheet applications or other databases could be candidate applications. But in an ‘open’ environment like the Internet, you would be less likely to take this option.

Privacy on the net is of course possible, essentially through encryption based on secret keys. This makes key management an issue. If you have various plastic cards, each with their own PIN, you will know how difficult it can be to remember each one. Many people will resort to writing down the PINs or keys, with the result that security is compromised. However, this is a reasonable human response to an unreasonable expectation. With public key cryptography, whose details are well-known, though whose implementation is the subject of US government restrictions, we may encrypt with a public key and decrypt with our private key. Provided the key is big enough this is pretty secure (unless your adversary happens to have access to a supercomputer or a farm of workstations). In any case it is possible to attack these schemes, principally by replay attacks (where, for example, you copy the transaction to buy a Porsche over the net, replay it, and intercept the second delivery), the man in the middle ploy (where you pretend to be the supplier and therefore acquire the essential information which is then used surreptiously), or perhaps by analysing the traffic very carefully to identify useful information which is then sold or used. Much of current communication depends on trust: that does not mean that the transactions have to be completely accurate: provided we have reliable and well-understood remedies for mistakes, we are usually happy. In general terms we have to balance the risks and the security: high levels of security usually involve high levels of inconvenience. At present Java tends towards security.

So there. What’s it got to do with us? It is not inconceivable that TEX itself could be written in Java. But who would want to ship that behemoth around just to run through a document? Had TEX ever been modularised, this might have been viable, but its monolithic structure militates against this. What then of the helper applications – the viewers or the printer drivers? Possibilities. Remember that the ‘applet’ (more like macro-applet) need only be written for a single virtual platform. This may be more difficult in the first instance, but in the longer term it has to be worthwhile. Future proofing is in-built. I go back to the notion my avatar promulgated in *Baskerville* 5.3 – distributing `.dvi` but this time having a Java applet be the viewer or printer. Use once, discard. Not an issue for us, since we all have a good viewer, but useful for the disenfranchised. The drawback lies in the fonts, but as suggested then, Adobe’s Multiple Masters will do nicely.

Did I mention that Java won’t run on Windows 3.x? or on Mac OS6? You need Windows 95, NT or the Mac’s Copland. So throw away those slow old machines. But wasn’t that always the story?

VI The Euromath System

Jill Tardival
Computing Laboratory, University of Kent
Email: gmt@ukc.ac.uk

Mathematicians, scientists and engineers who use mathematical notation have particular typesetting needs which generally have not been met by available software. The principal problem is typesetting mathematical symbols in documents, which in the majority of cases is performed using edit/view cycles as with LaTeX. Using such systems neither equations within mathematical documents, nor emails containing such notations, can be viewed directly from the authoring system. Mathematics is a science which depends on the use of complex symbols in order to describe abstract ideas and the laborious methods used to exchange ideas is perceived as a barrier to mathematical communication.

The European Mathematical Trust (EMT) was established to provide a powerful support system for the core of a mathematical computing environment. Software has been developed through European collaboration involving commercial organisations and academic mathematicians in France, Germany, Portugal, Denmark, Sweden, Eire, Slovakia and the UK. The product is the Euromath system, an SGML based, WYSIWYG, mathematical editor, based on the Grif SGML editor.

The current version of Euromath (Em1) is Unix based and has three Document Type Definitions (DTDs which determine the structure of documents) for creating articles, sheets and slides (OHPs). All include the facility to manipulate WYSIWYG mathematics without an editing/previewing cycle. Further DTDs, including letter and fax, are scheduled for future releases.

The incorporation of LaTeX to SGML conversion software for mathematics allows notation to be created in either of two ways. The appropriate symbols may be selected from symbol palettes, or if the author is already familiar with LaTeX, the notation may be typed in as a LaTeX string and converted to WYSIWYG SGML using a preset key combination. The equation will be seen as it will appear in the document and edited using either method. Similarly certain LaTeX documents containing mathematics can be converted on input to SGML and edited in a WYSIWYG fashion.

Standard features of the editor include automatic updating and numbering, search and replace facilities, European (ISO-Latin 1) language support, graphics inclusion (X11 bitmaps, xwd, Idraw, EPSF, CGM and TIFF) and WYSIWYG generation of commutative diagrams.

The Euromath software package includes a Gopher interface and automated on-line access to remote databases. Software which helps the user to create new DTDs, and to link the editor to other applications is also incorporated in the Euromath package.

Subsequent releases will include a PC platform, an integrated mailing system, an extended help system, extra DTDs, development of the directory services and an interface to computer algebra systems through the related OpenMath project.

The Euromath project has been funded by the European Union Science programme. The European Mathematical Trust is a registered charity, financially based in the UK. The editor is available to academic sites by subscription to Euromath. The current release is the first stage of a unified computer based WYSIWYG mathematical environment.

VII How does HTML handle mathematics?

Malcolm Clark

A Introduction

There is a very short answer to the question posed in the title: not at all. However, as an instantiation of SGML (see, for example, van Herwijnen, 1990, Goossens and Saarela, 1995b and Bryan, 1988), we can look first at how mathematics is handled in other SGML DTDs; and then examine how the expired HTML3 draft proposed to include mathematics.

The DTDs already available which are designed to handle mathematics include ISO 9573 (ISO, 1988) (confusingly, also known as ISO 12083), which is part of CALS, AAP (Association of American Publishers) (ANSI, 1994), and the HTML3 draft (Raggett, 1995). The Euromath DTD might also be relevant (Euromath consortium, 1996a), but since its status is closer to that of proprietary it is too awkward to consider here. Van Herwijnen (van Herwijnen, 1990) comments on the first two, comparing them to TEX and `eqn`. A longer and more detailed examination of the AAP, Euromath and ISO 12083/ISO 9573 is given in Poppelier, van Herwijnen and Rowley (Poppelier et al., 1992).

Van Herwijnen provides an example from physics for the decay of a particle together with representations in TEX, `eqn`, ISO 9573 and AAP. The equation is:

$$\Gamma(J/\psi \rightarrow \eta_c\gamma) = \frac{\alpha Q_c^2}{24}\,|A(J/\psi \rightarrow \eta_c\gamma)|^2\,\frac{m_\psi^2}{m_{\eta_c}^2}\left(1-\frac{m_{\eta_c}^2}{m_\psi^2}\right)^3$$

The entire expression is too extensive to compare here, but the left hand side of the equation in ISO 9573 may be given as

```
<mfn> <fname> &Gamma; <of>J/&psi; &rarr; &eta;
<sub> c</sub> &gamma;
```

while using the AAP dtd, it could be

```
<g>G</g>(<fr sol>J</><g>y</g> &ar; <g>h</g>
<inf>c</inf><g>g</g>)
```

For the sake of completeness, the `eqn` alternative is

```
Gamma(J/psi rarrow eta sub c gamma)
```

Eric's comments are interesting. He comments that the two SGML representations are cumbersome and difficult to read, especially when contrasted to TEX and `eqn`. He also rails against the obsession with representation. For example Γ 'means' decay width, but as far as the DTDs are concerned we have `&Gamma;` and `<g>G</g>`. To be fair, TEX and `eqn` hardly fare better, but at least we do know that we could have provided a more meaningful command. The second representation (AAP) is particularly unfortunate, since instead of treating the symbol as a symbol, it treats it as a Greek letter. Of course, Eric is deeply imbued with the basic notions of SGML, and would be very sensitive to this.

His contention is that someone who already knew TEX or `eqn` would have no motivation for learning or using these rather baroque alternatives.

There is a point to be made about the rather cumbersome nature of the SGML. Writing it by hand will be cumbersome, but surely no-one ever wants to write in this way. Structure editors are available. In the TEX world, Scientific Word gives a structure editor for LATEX. This can be done since it is possible to hold an equation as elements in a tree structure, so that modification or correction to an element can be managed quite simply, and changes can propagate down the tree. The same sort of thing exists within the SGML world. Euromath uses the Grif (Quint and Vatton, 1986 and Grif, 1996) editor for just this, and it would be easy to see other similar editors, like Chamberlin's Quill (Chamberlin, 1988) maintaining the information. There is a question lurking whether mathematicians would actually like to input in this way. Just as experienced keyboarders find GUIs very difficult and slow to use, perhaps the same sort of resistance would be found. However, the real point is that humans should not be expected to write SGML. If they really must write LATEX, then an approach like Scientific Word, which could be coerced into generating a tree structure which could be mapped onto a DTD, is potentially more valuable.

A general issue, which Eric raises implicitly, is that none of the DTDs offer a way of encoding meaning in a flexible way. Either the element is present already, or it is not. There appears to be no straightforward way of extending the range of elements. In the world of high energy physics and mathematics this must be something of a straightjacket. On the other hand, the bane of many editors' lives is the ease with which individual authors can 'extend' TeX or LaTeX by adding a few new definitions.

7 SGML Notations

If we really did have existing equations, then one way to handle them within SGML is through a Notation (see also Bryan, 1988). A Notation permits a document to include data which is not to be parsed. It is therefore possible to include TeX or LaTeX and assume that at that point a convenient processor will be magicked to deal with it. He gives the example of the definition in a DTD:

```
<!NOTATION TeX SYSTEM "">
<!NOTATION LaTeX SYSTEM "">
<!ELEMENT Formula - 0 CDATA>
<!ATTLIST Formula #NOTATION (TeX|LaTeX|eqn)
                   #CURRENT>
```

which may then be used later with the `Formula` element as

```
<FORMULA NOTATION=TeX>
...
</FORMULA>
```

A scheme which already maps SGML to LaTeX (e.g. Flynn, 1995 or Goossens and Saarela, 1995a) would find this a very easy way to absorb maths, provided all the equations were in the same notation. The prospect of a `\newcommand` or `\def` within the Notation could be worrying.

Although this sort of expedient is plausible, it is not entirely successful. One of the arguments behind the use of SGML is that it codes structure or meaning, rather than appearance (to echo one of Eric's points). TeX and LaTeX sometimes code meaning, but not in a consistent and reliable way. How do we extract information? If we have SGML, it is relatively easy to find corresponding structural elements, which may then be extracted. Once we start including a Notation, this chance is all but gone; and if we include alternative Notations (say TeX, LaTeX *and* `eqn`) it becomes even more problematic.

It is not clear to me how these Notations, or even the DTDs differentiate between in-line and displayed equations. I assume that an attribute could be included which specified the style. On the other hand ISO 12083 distinguishes between in line, displayed and 'display formula groups' styles, through the use of different elements.

There is perhaps a deeper question here, which is this, should it be at all relevant? Should the author be able to specify that some equations are in-line and others are to be displayed? It should make no difference at all to the content, although it would make great changes to the appearance. But to deal with maths is to deal with appearance, to a large extent. The display seems to be a key issue. Since many equations are strongly two dimensional (as opposed to the one dimensional nature of most text), it is a key question whether it is reasonable to expect this aspect to be reflected in any linearisation.

To give ISO 12083 its due, it says "Since there is no consensus on how to describe the semantics of formulas, it only describes the presentational or visual structure."

B HTML3 (expired draft)

HTML3 supports a `<math>` element which provides some capability for the inclusion of maths expressions. The draft (Raggett, 1995) does indicate that this capability is limited. It describes the functionality as similar to that found in "common word processing packages". In itself this may be seen as a hint of one of the driving forces within HTML3 – a desire to emulate word processor functionality. For a language which derives initially from a high energy physics community, this is a modest aim. The other design aim of `<math>` in HTML3 is to be "concise and comparatively easy to read". As a rider, it is suggested that this will make formulae longer than LaTeX, but shorter than Euromath or ISO 12083.

It is stated that the maths owes "a lot to LaTeX's math mode". In some cases it uses names for elements which are derived from TeX/LaTeX. Immediately a misconception springs up. The second paragraph of the section on maths gives examples of LaTeX commands: namely `\atop`, `\choose` and `\sqrt`. Only `\sqrt` is a LaTeX command. The `\atop` and `\choose` are unrepentant TeX. Ignoring for a moment the fact that there is a clear confusion between

LaTeX and TeX in the mind of the author of the draft (Dave Raggett),[2] there is actually a deeper problem. The use of an operator like `\over` requires much more work on the part of a processor, since it is often not until the whole expression has been parsed that sense can be made of what is actually `\over` what (see Rahtz, 1995). From years of teaching TeX, I can confirm that the `\over` command (and its buddies, `\above`, `\atop`, `\choose`, `\brack` and `\brace`) can lead to immense frustration and confusion.

HTML3 also has an `<above>` token, but this is quite different from TeX's `\above`. Since the TeX command is quite typographic, specifying the width of the line separating numerator and denominator, this is probably not too ambiguous. The function of the HTML3 tag is to allow something to be drawn above an expression. There is a similar `<below>` tag. `<above>` is a sort of numerator operator:

```
<math>
<above>x+y</above>
</math>
```

gives something like

$$\overline{x+y}$$

with the nuance that the element takes an attribute `sym` which can specify other symbols: `line` (the default), `cub`, `larr`, `rarr`, `hat` and `tilde` (in a sense `hat` and `tilde` correspond to `\widehat` and `\widetilde`). For example,

$$\overbrace{a+b+c}$$

would be

```
<math>
<above sym=cub>a+b+c</above>
</math>
```

the corresponding `<below>` has the same list of possible `sym` attributes, though I don't immediately understand what `hat` and `tilde` would do.

There are also some terminological surprises. `<math>` almost borrows the use of underscore and carat/circumflex for sub-scripts and super-scripts, except that they are referred to as index and exponent. The `^` and `_` are actually `shortref`s for `<sup>` and `<sub>`.[3] Since HTML syntax and LaTeX syntax are rather different, the ease of LaTeX's sub- and super-scripting has to be abandoned. The HTML tag has to be terminated. It is unfortunate that there was not a way of employing an implied end tag. For example

$$a_n^{23}$$

is given from

```
<math>
a^23^_n_
</math>
```

Should you need to subscript a subscript, the `shortref` form cannot be used. Although `<math>` does support a grouping operator,

$$a_{b_c}$$

is obtained from

```
<math>
a<sub>b<sub>c</sub></sub>
</math>
```

Perhaps cleverly, superscripting an expression with a binary operator results in the expression being placed over the operator, like `\stackrel`. For example

$$A \xrightarrow{\alpha'} B \xleftarrow{\beta'} C$$

would be

[2] I suspect this is quite a widespread misunderstanding.

[3] The notion of exponent and index may be welcome, since we would expect HTML to be concerned with the underlying content. Perhaps TeX does tend to overload the idea, since a superscript may signify more things than just an exponent. A semantic separation might be a very good idea, but this is unlikely to be the intention, since they are `shortref` characters for `<sub>` and `<sup>`. Poppelier *et al.* put this very succinctly: "What is the function of the 2 in SU$_2$, $\log_2 x$, x_2, x^2 T_2^2? In SU$_2$ it is the number of dimensions in the Lie group; in $\log_2 x$ it is the base of the logarithm; if x is a vector, the $_2$ in x_2 is an index; the ^{2} in x^2 could be a power, but if T is a tensor, the ^{2} in T_2^2 is a contrainvariant tensor index."

```
<math>
A &rarr;^&alpha;&prime;^
    B &larr;^&beta;&prime;^ C
</math>
```

In passing note that the use of the 'prime' operator is different from TeX/LaTeX use, and the use of space. I am not entirely clear about the use of space in `<math>`. It is certainly not ignored, as in TeX, and the draft does comment on the use of different horizontal white space within the equation. The draft runs

> Spacing between constants, variables and operators is determined automatically. Additional spacing can be inserted with entities such as `&thinsp;`, `&sp;` and `&quadsp;`. White space in the markup is used only to delimit adjacent variables or constants. You don't need spaces before or after binary operators or other special symbols as these are recognised by the HTML math tokeniser. White space can be useful, though, for increased legibility while authoring.

This does imply a rather different use of space. This use of space does have the effect that a 'prescript' can be made quite unambiguously simply by ensuring it is preceded by a space. It would imply a string `sin` is recognised as 'some sort of function'. The string `xyz`, would also presumably imply 'function', while in LaTeX it would imply the three variables *x*, *y* and *z*. It does leave unclear how $\sin^2\theta$ and

$$\max_{i=1}^{n} x_i$$

would be handled with ease. If the $\sin^2\theta$ uses `<sup>` but the

$$\max_{i=1}^{n} x_i$$

requires `<above>` and `<below>` then we have an interesting inconsistency.

`<math>` does adopt TeX/LaTeX's notion of binary operators, and in general claims to reflect the assumptions of TeX/LaTeX. It does not however provide support for multi-line equations, stating that 'this can be effectively handled by combining math with the `TABLE` element'. To me this wanders far from the basic concepts of SGML. However, what it appears to mean is that the `<array>` tag uses the same sort of syntax as `<table>`, not that an array uses the table tags.

From the draft, it is anticipated that chemistry could be set from within the `<math>` tag. I would view this as a mistake. It may be (almost) defensible from within LaTeX to use math structures, although the various chemistry packages at least try to separate the notions. It seems unfortunate that HTML3 should not attempt something similar.[4] An example might be

```
<math class=chem>
Fe_2_^2+^+Cr_2_O_4_
<math>
```

for

$$\mathrm{Fe}_2^{2+}+\mathrm{Cr}_2\mathrm{O}_4$$

where the different notational style of chemistry is tackled, notably its use of an upright font and consistent baselines for subscripts.

Some hints on appearance are provided: it is expected that functions (operators), numbers and other constants are portrayed in an upright font, and variables are italic. Unlike TeX/LaTeX, limits for integrals and summation signs are said to be placed directly above or below, or to the immediate right (depending on the symbol). Unfortunately, the draft does not indicate quite what this ambiguous term means. I suppose it does not mean 'emulate' the TeX/LaTeX mode, though that is obviously plausible, and from the point of view of a browser author could be a reasonable path.

What does it look like?

```
<math>
&int;_a_^b^{f(x)<over>1+x} d x
</math>
```

[4] And of course this emphasises the inadequacy of referring to a subscript as an index and a superscript as an exponent. The terms are pretty meaningless for chemical notation.

for

$$\int_a^b \frac{f(x)}{1+x} dx$$

Note that the sub- and super-scripts, like TEX/LATEX also denote limits.

Some maths accents are available: `<vec>`, `<bar>`, `<dot>`, `<ddot>`, `<hat>` and `<tilde>`. There are no explicit equivalents for `\check`, `\breve`, `\acute` and `\grave`, although they could be created with `<above>`.

Another borrowing from TEX/LATEX is the notion of grouping: HTML3 uses a `<box>` element where TEX/LATEX would use parentheses. `<box>` can be replaced by a `shortref` form of `{` and `}`, which greatly aids brevity and comprehension.[5] Although TEXnically a braced group is a sort of 'box', perhaps `<group>` might have been a better, though less concise term, in the context. It is perhaps an unfortunate choice, since 'box' carries overtones for many TEX users. Still, it does ensure that all the power of grouping is present (fairly essential in view of the `<over>` element). To overload slightly, one of the attributes of the `<math>` element is `box`, which causes a box to be drawn around the formulae. The `<box>` element is used in a number of ways; it is used, for example with the `<left>` and `<right>` commands for delimiters which grow. This leads to a rather strange construction:

```
<math>
f(x)=<box>
     (<left> 1+x <over> sin x <right>)
     <\box>
</math>
```

where `(<left>` gives a left parenthesis of appropriate size and `<right>)` gives the corresponding right parenthesis. As with TEX, it is recognised that sometimes it may be necessary to have a delimiter larger than 'default'. `<box>` therefore has a `size` attribute to enable this to happen. The permitted values are `normal`, `medium`, `large` and `huge`. The `shortref` form cannot take attributes.

Integrals (and other large operands which are stretchy) also need the use of `<left>`, without any corresponding `<right>`. For example

```
<math>
<box>&int;_-&inf;_^&inf;^
    <left> f(x,y) <over> x^2^+y^2^
    </box> d x&thinsp;d y
</math>
```

should give

$$\int_{-\infty}^{\infty} \frac{f(x,y)}{x^2+y^2} dx\, dy$$

Although `<math>` is said to recognise functions as maths entities, there is no list of all the entities given in the draft. However, straightforward elements like `<sqrt>` and `<root>` exist which work the same as their TEX counterparts. The LATEX notation for $\sqrt[n]{x}$ however is `\sqrt` with an optional argument: `\sqrt[n]{x}`, not as HTML3:

```
<math>
<root>n<of>x</root>
</math>
```

which is rather closer to TEX.

What the draft terms 'Greek letters' are available in a similar way to TEX/LATEX as entity references (and includes omicron). Some control over font styles is available through `<b>` and `<t>`. The first emboldens, while the latter makes upright (Times?). You may even combine the two as `<bt>`. Note that these changes apply to variables and constants, and not to numbers, delimiters, operators and other symbols. An interesting attribute is `class`, so that we could identify a vector as

```
<math>
<b class=vector>a</b>=A&prime;
</math>
```

Arrays or matrices are quite verbose, but broadly similar to LATEX. They are introduced by `<array>`, while each row starts with `<row>` and each cell with an `<item>`. This is in line with HTML's `<table>` model. Adapting some

[5] Should you need the symbols themselves, they are obtained by the entities `&lcub;` and `&rcub;`.

aspects of LaTeX, column definitions can be added, for example `coldef="CCCC"`, the default, where columns are centred. The alternatives are `R` and `L`. This is one of the few instances in HTML where case is vital. If the attributes are separated by `+`, `-` or `=` this will propagate down the array as a separator. For example

```
<array coldef="R=C+C+C">
```

Other attributes include `ldelim` and `rdelim` to specify the right and left delimiters of the array or matrix. Unlike other instances where a name is used, the symbol itself appears to be used in this context.

8 Summary

I tend to think that this makes it more difficult for someone with a LaTeX background to interpret `<math>`. When two languages are quite different, there is rarely confusion in flipping from one to the other, but when they share many similarities it can be frustratingly simple to converge at all the wrong times. Note also that `<math>` uses the ISO entity names for symbols (Smith and Stutely, 1988) rather than the TeX names. In a few cases this sows potential confusion. The confusion which exists within the draft between TeX and LaTeX is not of itself a problem, except that people coming to HTML3, being told it is ‘like LaTeX’, would find some key differences.

The main divergences are

1. the interpretation of space
2. need to close most SGML elements (e.g. `^` and `_`)
3. `<over>`
4. interpretation of functions (a consequence of 1.)
5. entity names similar but not identical to corresponding commands
6. use of `&prime;`
7. missing commands
8. poor support for cross-referencing

C Future of maths in HTML

At this point, it may be worth considering the extent to which any maths expression can divorce the semantic component from the form on the page. Often the way equations are portrayed can assist in their interpretation. Both tables and maths seem to be examples where the meaning and the appearance are very closely intertwined. There are instances where an author changes notation in order to pursue an argument. One assumes that the essential meaning does not change between changes in notation, and that perhaps a markup system might not even note the change, except perhaps as an attribute.

The draft document which forms the basis of this discussion expired in September, 1995. Some of the new structures which it introduced, notably tables, form part of most browsers now, but mathematics didn’t make it (although `<sub>` and `<sup>`, without `<math>` did) (see W3C, 1996b). An email from Dave Raggett (to David Carlisle), who wrote the draft, notes that W3C

> has set up a small working party on math, and we expect to publish a detailed proposal by early Summer. The March’95 spec will provide a starting point, but we may end up with something rather different.

He goes on to state:

> The W3C math group has the goal to develop an open specification for HTML math that:
>
> Is suitable for teaching, and scientific publishing.
> Works with symbolic and numerical math applications
> Supports filters to/from other math formats, e.g. TeX
> Is easy to learn and to edit by hand
> Is well suited to template and other math editing techniques
> Can be rendered to:
> graphical displays
> speech synthesisers
> plain text displays e.g. VT100 emulators
> print media, including braille
> Supports lengthy expressions via fold/unfold and line breaking with author control.
>
> This is shaping up as the need for simple macros and declarations that define terms etc. for use across multiple `<math>` elements, and parsing of `PCDATA` using “models” referenced by `<math>` elements. These models

define how to interpret stuff at a level sufficient to support symbolic manipulation without having to make all these distinctions explicit in the markup itself.

This is quite a bold extrapolation from the original specification. Some of the suggestions seem to me to be incompatible. The inclusion of symbolic and numerical applications is interesting and goes far beyond the existing maths DTDs.

The relevant W3C web page (W3C, 1996c) contains a reference to QED, an ambitious programme to build a 'single, distributed, computerized repository that rigorously represents all important, established mathematical knowledge'. If this genuinely represents part of HTML's solution to maths, we will have some time to wait before a system is available.

D Conclusion

It is not clear to me at present whether HTML will ever be rich enough to do the sorts of things which mathematicians and physicists want to do with maths. Of course, it could be that these are not the market at all. I have long argued that one of TEX's (and LATEX's) major problems is that the population of users who benefit by it is small – very much a minority. And the expansion of the use of computers has made them an even smaller minority. At best we are a niche market. Some of the simpler problems are already tackled quite conveniently by word processors, further eroding the niche. Why bother with mathematics at all? Is it really worth the effort, compared with something sexy like `<frame>`s?

Having said that, it appears that Public Entities in ISO 8879 (Smith and Stutely, 1988) are sufficient to encompass most of the symbols I have seen in LATEX and AMSTEX. The potential is there. What makes this especially intriguing is that there is software around like Panorama from SoftQuad which is designed to enable any SGML document whose DTD is known to be rendered on the screen. Therefore for truly 'heavy' applications, this seems a much better way to go. In fact, I would see it as an altogether better way to go. Browsers which could examine the `Doctype`, find it on an appropriate server and then render it would be much more flexible, and enable us to use existing SGML documents easily on the Web. HTML would simply be a lightweight DTD used because it had a lower overhead.

On the other hand, browsers like Netscape Navigator are becoming larger, are starting to include 'plug-ins' and be Acrobat-aware. If HTML3 version 2, as outlined by Raggett does support filters (hopefully to LATEX rather than TEX), as well as symbolic manipulation (Mathematica, Maple, etc.), it could be a very powerful tool. However, the latest information available through the World Wide Web Consortium does not encourage belief that this is much more than a dream.

References

ANSI/NISO, (1994). Information and documentation – Electronic manuscript preparation and markup. Technical Report, ISO 12083:1994 ANSI/NISO.

Bryan, M. (1988). *SGML an author's guide*. Addison-Wesley.

Chamberlin, D. D. (1988). An adaptation of dataflow methods for WYSIWYG document processing. In *ACM Conference on Document Processing Systems*.

Euromath consortium (1996a). Euromath system. `www.math.ethz.ch/~shared/emb/GeneralFiles/Euromath.html`

Flynn, P. (1995). HTML and TEX: making them sweat. Baskerville, 5(2):7–11.

Goossens, M. and Saarela, J. (1995a). TEX to HTML and back. TUGboat, 16(2):174–214.

Goossens, M. and Saarela, J. (1995b). A practical introduction to SGML. TUGboat, 16(2):103–150.

Grif (1996). GRIF S. A. – SGML and HTML solutions. `www.grif.fr`

ISO (1988). Information processing – SGML support facilities – Techniques for using SGML. Technical Report, ISO/IEC TR 9573:1988, ISO.

Poppelier, N. A. F. M., van Herwijnen, E. and Rowley, C. A. (1992). Standard DTDs and scientific publishing. *EPSIG News*, (3):10–19.

Quint, V. and Vatton, I. (1986). Grif: an interactive system for structured document manipulation. In van Vliet, J. C., editor, *Text processing and Document Manipulation*, pages 200–213. Cambridge University Press.

Raggett, D. (1995). `www.w3.org/pub/WWW/MarkUp/html3/` Expired HTML3 draft.

Rahtz, S. P. Q. (1995). Another look at LATEX to SGML conversion. TUGboat, 16(3):315–324.

Smith, J. and Stutely, R. (1988). *SGML the user's guide to ISO 8879*. Ellis Horwood.

van Herwijnen, E. (1990). *Practical SGML*. Kluwer Academic Publishers.

W3C (1996b). HTML 3.2 specification. `www.w3.org/pub/WWW/MarkUp/Wilbur/`

W3C (1996c). Math markup in HTML. `www.w3.org/hypertext/WWW/MarkUp/Math/`

VIII Hyper-G: Information—the Next Generation

Klaus Schmaranz

Summary
The first part of the paper deals with problems and shortcomings of electronic publishing today that have up to now prevented the really big breakthrough. In the following sections the use of Hyper-G, the first second generation Web system for distribution of electronic publications on the Web as well as on CD-ROM is discussed. It is shown that the object oriented database architecture of the Hyper-G server overcomes most of the problems of today's solutions. The last section contains a short description of journals and books that are already utilizing this new technology.

Please note that throughout the paper the name Hyper-G is used describing the technology of the system. The server itself as a product is available under the product name HyperWave.

A Introduction

Speaking about portable documents one will easily realize that portability of documents has several faces. Mainly portability means that documents can be transported on the Web. This implies that the document format chosen has to be compatible with several hardware and operating system platforms. So far one can make sure that interesting information is useful for the majority of Web users by choosing the right format.

Although a well chosen document format makes sure the documents are usable for readers this is only the first step to really portable information. Documents on the Web only make sense if interested readers can find the information they desire. Thus easy location of documents has to be considered to be part of the portability aspect too. Besides easy data retrieval transmission speed is a sensible point to be considered. Readers having to wait annoyingly long for documents or having to deal with broken connections are very likely not to be interested in electronic publications at all.

Considering that at the moment an estimated 300 million users from different cultures and speaking different languages have access to the Internet, multilinguality of documents becomes more and more important. Although with today's methods it is not possible to translate documents to other languages on the fly there is a lot that can be done. One can consider publishing papers in more than one language and mechanisms can be implemented to give the user the possibility to look up unknown terms. Dictionaries as well as glossaries can be made available for specialized areas.

Electronic publishing naturally does not only mean to publish documents on the Web. There are also a lot of potential readers of electronic publications that still have no Internet access. To keep costs low the system chosen for distribution must be able to support both Web and CD-ROM. Otherwise the amount of work doubles because these versions have to be prepared seperately.

A look at electronic documents on the Web shows that at the moment most of the Web sites serving electronic publications are run by universities and only a few are operated by publishing companies on an evaluation basis free of charge. However electronic publishing in the long term is surely driven by publishing companies which means that charging mechanisms and user as well as group access management are highly important topics.

In the following sections you will find a detailed discussion of the features of Hyper-G, the first 'second-generation' Web server [Maurer 96] that make electronic publishing easier and more effective than ever before. This includes navigation issues that help the readers to find their way through the electronic jungle. Also the standpoint of publishing companies concerning billing and access rights is considered making sure that the system will fit their needs. Additionally turnaround time from submission to appearance of a paper and the cost effectiveness of this process will be discussed.

B Using Hyper-G for Electronic Publishing

Before discussing new publishing paradigms let us have a closer look at Hyper-G. Using Hyper-G for electronic publishing solves a lot of the problems mentioned above and opens the way for completely new electronic publishing paradigms.

Hyper-G automatically supports hybrid Web and CD-ROM publication without additional effort as has been successfully proven over the last two years with J.UCS, the Journal of Universal Computer Science by Springer-Verlag [Maurer 94].

The kernel of the Hyper-G server is an object-oriented distributed network database with a separate link database. Information structure as well as document meta information are a basic part of the concept [Kappe 91]. This makes it possible to present the user with a seamless world-wide structured information space across server boundaries.

The document structure gives readers the possibility to locate interesting papers easily since all the information can be structured by topic, by journal or by other criteria that are important. Structure in Hyper-G is achieved by the use of collections that themselves can hold collections or documents. This concept allows it to build structured information trees where the contents of a collection need not even reside on the same server as the collection itself. Collections or documents can be members of arbitrary many collections without physically copying them. The result is that different structures can be applied to the same dataset and users can choose the most logical structure for their purposes.

Document meta information such as author, title, keywords, creation date, modification date as well as expiry date and many more, support the readers in getting as much information as possible. Naturally document meta information is searchable and full text searches may be performed. The scope of searches is user definable and can be one small part of one server or even the whole content of all servers worldwide in one single operation. Even when doing searches on multiple servers it is not necessary to know about the server addresses.

More than that: meta information cannot be applied only to documents but also to hyperlinks! This means that links can have types, such as annotation links, inline links, also version links for documents where multiple versions exist and many more.

Hyper-G servers do not provide read access only, write access is also possible. Read and write access to documents are controlled on a user and group access right basis and billing is integrated in the server.

All links in Hyper-G servers are stable [Andrews 95], which means that dangling links (links pointing to nowhere) are impossible. Whenever a document is moved from one location to another, even across server boundaries, all the links pointing to that document automatically point to the new location. This kind of stability is achieved using URNs (Uniform Resource Names) instead of URLs [Berners-Lee et al. 94]. URNs can naturally be mapped to URLs when accessing a Hyper-G server with a standard WWW or Gopher client. If documents are deleted all the links pointing to this document remain open and are hidden. Whenever the document reappears the links are closed again.

Not only are all links stable, they are also bidirectional. Accessing a document on a Hyper-G server readers do not see only the outgoing links of documents but also all links pointing to the document from the outside. Links can be followed in the reverse direction, which would not be possible using first-generation Web servers. This very special feature is achieved by the separate link database of Hyper-G servers. If links were embedded in the documents themselves as is the case with first generation Web servers this would not be possible.

The separate link database has another advantage: it makes every document hyperlinkable even if the document format does not allow links [Maurer 96].

Amongst other structuring elements Hyper-G supports the concept of clusters. A cluster contains several documents that are related to each other and therefore should be viewed together. As an example out of chemistry 3D molecular models could be clustered together with an explanatory text. In this case the user would get the 3D model in one window together with the explanatory text in another window.

Clusters are also used to serve multilingual documents. Documents in different languages are clustered together and the user then gets the document matching his language preferences. In first-generation Web systems the only possibility to have multilingual documents is to let the user choose the language on the entry page and then follow different paths through the server for different languages. This approach causes a lot of work for server operators and the readers have no chance to change the language while reading. With Hyper-G only one path through the server has to be maintained and the readers can switch between multiple languages on the fly [Andrews 94].

Versioning of documents is also supported using clusters. Different versions of a document are clustered together and the reader can switch back and forth between different versions on the fly. To support the server operator and keep maintenance of multiple document versions easy a special parser is available. When updating documents this parser tries automatically to find the positions of all hyperlinks of the old version in the new one.

Other special features of Hyper-G are glossaries and automated glossary hyperlink creation. A glossary in Hyper-G is defined as an arbitrary collection of explanatory documents that are classified by their titles and keywords. Hyperlinks to glossary items in a document are then automatically created by a special parser that searches for the glossary items in the text of the document. Automated creation of glossary links is at the moment supported for HTML and HTF documents, PDF and PostScript support are under development.

To make creation of referential hyperlinks easier a *Vocative Hyperlink Creation Language* (VHCL) has been implemented. This language supports the description of document context and potential hyperlinks in that context. As an example typical phrases like “see also page *nn*” would be recognized by the program and a link to page *nn* would automatically be created. Journals normally have their well defined citation rules making it easy to write a VHCL program that recognizes citations and automatically creates inter document as well as intra document hyperlinks. As is the case with glossary links this feature is at the moment implemented for HTML and HTF; PDF as well as PostScript support are under development.

C Providing Quick Access

As has been mentioned earlier it is extremely important to provide quick access to information otherwise it would be worthless. Since Internet is neither very reliable nor fast considering long distance data transfer it is necessary to mirror documents to several servers world-wide. Doing this readers are able to choose the server that is geographically most convenient for them in the network sense.

Implemented in Hyper-G are two mechanisms that make long distance transmissions effective: first a cache is implemented that works as all the well-known proxy servers do. Although caching can help a lot it is surely not enough, because lifetime of documents in the cache can be rather short depending on the traffic. For this reason a second mechanism called replication is implemented.

Replication means that documents from one server can be mirrored to other Hyper-G servers and the replicated documents know about the original. This functionality is one of the benefits of using URNs instead of URLs, it would be impossible to implement it for first generation systems using URLs.

Besides caching and replication readers can actively utilize the ability to have write access to their personal home collection. Instead of defining bookmarks on their local computer they can insert references into their home collection on their Hyper-G server. The benefit is that their “bookmarks” are then accessible from wherever they connect to the server, which is especially ideal for people who are travelling a lot.

D New Publishing Paradigms

Having a closer look at the way electronic publishing is done today one will mostly find HTML or PDF documents that are very similar to their paper based counterparts. Often a search engine is provided to make location of interesting papers easier, all other benefits of doing publishing electronically are mostly neglected. For the reader of electronic publications nearly no value is added compared to paper based articles. Worse than that—considering HTML documents the possibility to do high quality printouts for archival purposes is lost. This is surely not enough to make electronic publishing on the Web a success.

In the above section discussion was about the special Hyper-G features. Utilizing them allows completely new electronic publishing paradigms that are no longer driven by technical demands and shortcomings of certain document formats like HTML. Instead authors can concentrate on the content rather than the document format and choose a format convenient for them without loosing important hypernavigation features.

As an example a paper about new chemical structures could consist of 3D models of molecules that are clickable. The hyperlinks could then lead to spectrum images that are then linked to some additional text based explanations in, for example, PDF [Adobe 93]. A video of an experiment, naturally again with hyperlinks to explanations, completes the presentation.

All the documents in the example above carry meta information like keywords and can therefore easily be located in a search.

Acceptance of electronic publications is highly dependent on their quality. For electronic publishing quality does not only mean high quality contents, which can be assured by an appropriate refereeing process. Stability of electronic publications is at least as important. Technically it is easy to change electronic papers after publication but this is unacceptable. Instead Hyper-G’s annotation and versioning mechanisms can be used to alert the reader of new results or errata. In this case the paper is not changed at all, only additional information is added to the paper. Therefore all citations of the paper that existed for the original version are still valid and the reader can choose to browse annotations and newer versions of the paper on demand.

Annotations in Hyper-G are hyperlinks pointing to the document that is annotated. Since Hyper-G’s links are bidirectional the reader simply follows an annotation link backwards to read the annotation. The use of URNs in a link database instead of URLs embedded in documents guarantees that the annotation links are stable. This means that an annotated document can be moved around in the server or even from one server to another without generating annotations that point to nowhere. All links that pointed to the document before are then pointing to the document at its new location.

Being able to examine the neighbourhood of a paper makes it possible to find other interesting papers on the same topic that very likely are difficult if not impossible to locate if only unidirectional links were possible as is the case with first-generation Web servers.

E Turnaround Time and Cost Effectiveness

One of the most time consuming processes of electronic publishing is refereeing. Up to now refereeing normally means that a paper copy of the submitted paper is sent to the referees who send a corrected paper copy back. The annotated version is then sent to the author. If there are misunderstandings between referee and author the document is usually sent back and forth several times.

A much faster turnaround time can be achieved if refereeing, corrections and clarifying misunderstandings can be parallelized. The logical way is to do refereeing electronically. Using Hyper-G's electronic annotations this task can be performed easily: papers are inserted into the Hyper-G server with read access only for the referees and if the referees agree also for the author.

The referees then comment on the papers using the annotation mechanism. If desired annotations can also be made readable for the author, so the author is able to react immediately on the referees' comments. More than that – the author himself could also annotate the referees' comments to clarify misunderstandings. Naturally the author as well as the referees remain anonymous [Maurer 95].

This kind of refereeing shortens the time used for the whole process significantly because the authors are able to do corrections in their papers and clear misunderstandings while refereeing is still in progress. There is no longer a need to send papers back and forth between referees and authors.

Naturally it would be too optimistic to think that refereeing can usually be done within days instead of weeks. There is nothing that can be done about referees that are too busy, but most of the time it helps a lot to do everything in parallel instead of sequential.

An additional benefit of this kind of refereeing is that the whole process from submission over refereeing to publication is automatically documented and can be stored for archival purposes.

In general electronic publications are considered to be cheaper than paper based publications. This is true if both the electronic as well as the paper based version of a paper have the same contents and the electronic version only provides full text search as added value. This is not true if one wants to utilize all the additional features that lie in the electronic nature of the medium. The final step of inserting the paper into the server is in this case not only the insertion itself. Also hyperlinks as well as structure, eventually a version of the paper that is split into single sections or versions of the paper in different formats have to be prepared. Naturally there can be other electronic specialities like Java scripts or 3D navigation rooms and many more. All that is very time consuming if it has to be done by hand. The whole process is critical in terms of cost effectiveness and should be automated to the highest extent possible.

Using the special tools that Hyper-G provides such as the *Table of Content Generator*, the *Glossary Hyperlink Generator* and the *Vocative Hyperlink Creation Language* automates the additional work to the maximum extent possible. The steps to be performed at insertion are limited to running the tools and controlling them. All reference and glossary link creation is done automatically.

F Current Electronic Publications With Hyper-G Technology

The first electronic journal based on Hyper-G was J.UCS – the Journal of Universal Computer Science by Springer-Verlag. It is a monthly journal covering all knowledge areas of computer science and additionally to the Web version a yearly CD-ROM and printed version are provided by Springer. Papers in J.UCS appear in two parallel formats: hypertext and hyperlinked PostScript. PDF is planned for 1997.

Springer also publish Few Body Systems (FBS), one of the most reputable Journals in physics. Started in January 1995.

Academic Press distribute the Journal for Network and Computer Applications (JNCA) (former JMCA and JMA). Started in January 1996.

Datenstrukturen by Ottmann and Widmeyer, the German bible of Data structures. This uses hyperlinked PostScript.

Meyer's Lexikon, one of the most comprehensive German encyclopedias, is electronically available on a Hyper-G server on an n-user license basis.

Addison-Wesley publish some 30 books electronically on the Web using Hyper-G.

References

Adobe Systems Inc.: Portable Document Format Reference Manual, Addison-Wesley (1993).

Andrews, K., Kappe, F., Maurer, H. and Schmaranz, K.: On Second Generation Network Hypermedia Systems, Proc. ED-MEDIA ’95, (1995), 69–74.

Berners-lee, T., Cailliau, R., Luotonen, A., Nielsen, H. and Secred, A.: The World-Wide Web. Communications of the ACM 37, 8 (1994), 76–82.

Andrews, K. and Kappe, F.: Soaring Through Hyperspace: A Snapshot of Hyper-G and its Harmony Client, Proc. of Eurographics Symposium on Multimedia/Hypermedia in Open Distributed Environments, Graz (1994).

Kappe, F., Maurer, H. and Tomek, I.: Hyper-G – Specification of Requirements, Proc. Conference on Intelligent Systems (CIS) ’91, (1991), 257–272

Maurer, H. and Schmaranz, K.: J.UCS – The Next Generation in Electronic Journal Publishing, Computer Networks and ISDN Systems, Computer Networks for Research in Europe, Vol. 26 Suppl. 2, 3, (1994), 63–69.

Maurer, H. and Schmaranz, K.: J.UCS and Extensions as Paradigm for Electronic Publishing, Proceedings DAGS’95, Boston Massachusetts, (1995).

Maurer, H. ed.: HyperWave – The Next Generation Web Solution, Addison-Wesley, (1996).

IX Portable Documents: What Next?

Frank Harwood

9 Introduction

This article is a summary of the conference held on 15 February 1996 by the British Computer Society Electronic Publishing Specialist Group in conjunction with UKTUG continuing the 'portable documents' theme. The two previous articles on Euromath and Hyper-G also cover papers presented at this conference and so are not mentioned in this summary.

A What Next?

10 Les Carr, University of Southampton

"Now we have them, how do we use them/maximise the benefit?" was the theme of Les's talk. We have available a spectrum of portable formats, from `.PDF` with high visual fidelity, to SGML which preserves logic of content without reference to appearance. (The intriguing question was floated and left open, that although we have a large measure of portability between systems – do we have temporal portability? – can I read today's electronic document in 30 years time?)

Documents are now available from around the globe, and to be most useful need links between them. Hypertext-type linking is found in personal systems (e.g., Guide, Hypercard) through to global systems (e.g., WWW), but to a degree, all are 'closed' – a non-universal and therefore 'proprietary' markup is used which does not extend into other people's systems.

The Microcosm Model (at Southampton) separates the document control system (how to produce and display it), from the link control system. Links are any type of relationship between documents, and flexible link definitions are allowed for. The Open Journals Project, funded by JISC, applies Microcosm technology to WWW. It is possible to integrate on-line journals with each other and with various on-line databases and teaching resources. The concept of a document becomes very broad indeed and the databases of links (linkbases) become value added commodities in their own right. Linkbases are configurable for different levels and purposes. This could be seen as opening a new publishing idiom where the various 'closed' technologies mentioned become local and short term solutions.

See it at `http://journals.ecs.soton.ac.uk/`

B Converting from LaTeX to SGML

11 Sebastian Rahtz, Elsevier Science

An in-depth review, heavily illustrated with examples, was given by Sebastian, discussing the problem, various feasible approaches, and demonstrating results so far produced at Elsevier (package not in the public domain).

The publisher faces a large community using LaTeX, a mature notation and free typesetting system, well suited to scientific and multi-lingual work. Unfortunately, it is not what the publisher uses, not an international standard and does not convert for various purposes as does SGML.

Four practical approaches were mentioned:

1. Throw away electronic file and retype.
2. Strip out TeX coding and treat as unknown word processor.
3. Write program to parse LaTeX and output SGML.
4. Re-implement TeX to output SGML codes.

of which the first two were not discussed.

The parser approach has been implemented a number of ways but can only be partially successful because TeX is macro based with extensible syntax. All results so far require afterwork to tidy up. To implement route 4, three methods have been used.

1. Replace TeX backend
2. Rewrite TeX in a new language – has been done in LISP
3. Write SGML code to the dvi file and extract it from there – used by Elsevier.

The work done was described in detail stressing the importance of the target DTD and the richness of the LaTeX, highlighting also a number of pitfalls. The acid test is that it works and real scientific papers can be translated, though human intervention at some level is frequently needed to perfect the end product.

For more detail, see *TUGboat* 16.3.

C SGML is here

12 Andrew Dorward and Neil Bradley, Pindar

Substituting for the speaker originally planned, Andrew and Neil gave a lightning rendition of the SGML story – principles, implementation considerations, current developments and criteria for use or rejection!

The principles of SGML are well understood within the group. It is an open system, defined by ISO 8879 (1986). There has been a recent expansion of interest, triggered by HTML – restricted and non-open. The current state of play is that SGML is used for many more 'pages', but has a lower profile. Recent developments, particularly DSSSL and Hytime provide enhancements to the use of SGML. Although SGML is independent of any software publishers, there are numerous products around – parsers, editors and DTD tools, and in a production situation it makes sense to adopt the best available. The new Frame SGML software suite – just out of beta-test and becoming available now, was strongly recommended.

Implementation considerations were discussed using a markup of an article as an example. Whereas the considerations are the same whatever the task, the use of appropriate software can aid efficiency, reduce errors and give more options on the use of tagged data and the control of style.

Take account of:

"Granularity" – how far to break down the material. High granularity = a fine breakdown. This adds value to the information at a price (in effort) and should be chosen appropriate to purpose.

"Hierarchy" – markup objects can contain other markup objects. Many levels are possible. Again this needs to be set appropriate to material and purpose.

"Attributes" – e.g., use of `<name 'personal'>` and `<name 'company'>` can enable separate listing.

"Hypertext" – the setting up of cross-reference jumps from one point in the text to another, internal or external to the document or database. Here again the software used can greatly facilitate the process.

DTD – the Document Type Definition controls granularity, quality, optional/mandatory items, alternatives, sequences, element names. A visual DTD tool such as "Near and Far" makes for faster and more accurate production of DTDs.

Authoring Environment – may be structured (software from Frame, SoftQuad) where the authoring process is interactively constrained according to the DTD, or 'loose' (software from Microsoft or other out-of-line parsers).

There is a checklist of reasons to adopt SGML. With two ticks against the list SGML should be considered, with four ticks, it would be very foolish not to use it.

- long shelf life data? (SGML is too expensive for transient data)
- for multiple media publication?
- frequent republication?
- need searchable database?
- for inter-department or inter-company exchange?
- new product extraction from existing data?
- need heavy hypertext?
- industry requirement?

A fictional "must use" example was given. A Company with a body of high value articles wishes to publish on hard copy, CD-ROM, WWW, also abstracts, also lists of articles and contributors as separate products. All can be pulled out with little effort after the initial investment in SGML.

D Java – The Krakatoa of the Web

13 Henry Rzepa, Imperial College, London

This item outlined a particular body of work done at Imperial College before the appearance of Java, the improvements made possible by Java, and some informed speculation on future developments.

Work started in 1994 to try to publish representations of 3D molecular structures (MIME type). The concept was to be able to click on a hyperlink and get a 3D rotatable model within a 2D document. This required a marked up dataset defining the model and a script on the user's computer which read and interpreted the markup. A result which worked was achieved, albeit with non-standard components and on Unix only. When VRML was brought into use in March 1995, improved communication between the 3D model and 2D document resulted, but the whole was still non-standard and Unix only.

The introduction of the Java language to the project in July 1995 achieved a seamless interface between the WWW client, data and action, and seamless memory and security models for the whole "document". Not only did this fulfill the original concept elegantly, but extra features became possible. Rotatable models of various representations could be interchanged and extra information delivered, e.g. by clicking on certain points of the model, Java applets could be invoked which delivered inter-atomic distances.

Java in its present form, as a C++ like language, is a powerful tool, used as directed, and had a major impact on this particular project. There are plans to use the project as a basis for an electronic conference in June 1996. Its full features are available on Unix and Windows 95 platforms only, and whereas this accounts for 90% of visitors to Imperial's web site, it really needs Windows 3.1 and Mac versions for universal take-up. There is also mileage to be had from a future integration of VRML with Java. Apple's CyberDog is an alternative but may be released 2 years too late for acceptance.

There are, of course, problems. Already it appears that early Java applets are not compatible with later; a serious flaw which must be addressed before Java can become a mainstream language/method. For publishing/WWW applications there is a particular danger that data incorporated within applets will "disappear" in the sense that it will not be searchable as is HTML data.

See it (and rotate it!) on:

`http://WWW.ch.ic.ac.uk/java/java_1.html`

X An aged archivist remembers. . .

Sebastian Rahtz

In his recent article on *CTAN past and present*, Robin Fairbairns' description of the events leading up to CTAN do not quite correspond to what I (as, I think, the longest-standing active TEX archivist) recall. Firstly, the history of the Aston Archive should distinguish between three phases: Peter Abbott's sole and herculean efforts in 1987 and 1988; the establishment of an archive group by Peter after the 1988 TEXeter conference; and the translation of the original VAX-based archive to a Unix box in 1992. That second stage was important, because the archive was the first in 1989 to have a clear *structure*, decided on at the second archivists meeting.

Secondly, what happened between about 1990 and 1992 at Sam Houston was not CTAN; it was an inchoate treasure trove looked after by George Greenwade, which was but one of a group of big archives which included `labrea` at Stanford itself, Don Hosek's `ymir` and Stuttgart (Stuttgart was around a long time before CTAN, with a good service).

Thirdly, the TUG working group which defined the CTAN concept started in mid 1992, but did no more than talk until Aston took the lead in the summer of 1992 and established the first working incarnation of the ideas thrashed out in email. I say this confidently, because I vividly remember the two days I spent in Peter Abbott's office setting up his new Sparc, and creating the first CTAN. At that time I also defined the gopher interface, nightly indexing, extensive mirrors and dynamic zipping and tarring. Subsequently, this 'Daughter' archive ran in parallel with the VAX (which was still physically visible to delegates to TUG'93 at Aston) for a long time. George Greenwade was delighted with what was created, and started to set up SHSU in the same way, and Rainer Schöpf also rapidly converted the Stuttgart system; by the time the three of us met physically at the TUG'93 conference, we were confidently claiming mirror status of each other (not entirely truthfully!), and George gave a successful live demonstration at the conference of the Aston service. The complicated email-based maintenance scheme was developed by Rainer *after* TUG '93, after we had experimented with NFS and other games.

I do not apologize for perhaps boring *Baskerville* readers with this slice of history. It is a sad fact (and I suppose a compliment) that the TEX archives have been used as pawns in the TEX-world politics that so deplorably weaken our joint endeavours; so it is only right that we should not lose sight of how things developed.

XI Malcolm's gleanings

Malcolm Clark

A In consistency

'A foolish consistency is the hobgoblin of little minds'. That tireless defender of TEX Allan Reese made no mention of TEX whatsoever in his last article published in the *Times Higher* (March 17th?) demonstrating his immunity from the charge of hobgoblin. That's fine this once, but if he wants his accolades to accrue he must adhere to the one true path.

B MINSE

Browsing through the World Wide Web Consortium's maths-related pages, I came across MINSE. Described by its author, Ka-Ping Yee, as a 'simple, extensible way to express mathematics' it does seem to contain many interesting and useful ideas. Besides taking a semantic description of an equation and rendering it into several different forms, it may also render it into audio (with some acknowledgement to T V Raman). Ping also says that a MINSE expression may be turned into a form which is understood by TEX. If you want to know more, point your Web browser at `www.lfw.org/math/demo.html`.

C Times change

Our chairman, now the proud possessor of a copy of 'TEX and Metafont, New Directions in Typesetting', pointed out the following quotation to me:

> "Those of you who wish to define control sequences should know that TEX has further rules about them, namely that many different spellings of the same control sequence may be possible. This fact allows TEX to handle control sequences quite efficiently; and TEX's usefulness is not seriously affected, because new control sequences aren't needed very often."

Robin adds,

> "It's plain that Knuth hadn't at the time yet discussed life, the universe and everything with Lamport!
>
> The rules themselves are indeed horrendous, and go out of their way to put automatic generation of control sequence names pretty much out of the question."

D *TUGboat*

It was good to see *TUGboat* recently. I did note that there has been a minor presentational change to the journal. The spine now merely notes the volume and number, omitting the month and year. It does make it that bit more difficult to see how late it is. Wise move, if ominous, just as it seemed to be getting on time. Not that I can really say anything about timeliness, the edition of *Baskerville* that I edited took an awfully long time to get through the process. I do think the content was good. Thanks to those who contributed. On the subject of journals, let me commend Don Hosek's *Serif* to you, now on edition 4 (see *Baskerville* 5.1). It's early preciousness seems to be evening out, and the proof reading has similarly improved. Since it is set in TEX, it is especially fascinating. Just how does Don manage to do all those things which we never expect TEX to do. It is a great pity that he did not provide the article for the Santa Barbara *TUGboat* Proceedings (it should have been in 15(3)) from the talk he gave there, where he discussed ways of bending TEX to his will. Hint?

XII Announcement of the Annual General Meeting

Malcolm Clark

The Annual General Meeting of the UK TEX Users' Group will be held on 16 October at the University of Warwick. Full details will be given in *Baskerville* 6.4.

The Constitution contains the following clauses:

> 17. The annual general meeting of the UKTUG shall be held in the United Kingdom not later than the last day of November in each year on a date and at a time to be fixed by the Committee and notified to members at least 35 days in advance for the following purposes—
>
> (1) to receive from the Committee a report balance sheet and statement of accounts for the preceding financial year
>
> (2) to fill the vacancies in the Committee and to appoint auditors for the ensuing year
>
> (3) to decide on any motion which may be proposed to the meeting in the manner provided below
>
> (4) to fix the entrance fee (if any) and annual subscription
>
> 18. Any member desirous of proposing any motion at the annual general meeting shall give notice in writing to the secretary not later than 14 days before the date of such meeting
>
> 23. No amendment (other than a motion for adjournment) shall be moved to any motion proposed at any annual or special general meeting unless written notice of the amendment shall have been sent to the secretary prior to the meeting

I think that Clauses 17 (3), 18 and 23 are self-explanatory.

A draft agenda, covering items (3) and (4) will appear in *Baskerville* 6.4. In the meantime, anyone wishing to propose any motion to the AGM must send me, in writing, the text of the proposed motion and their own name and address, to reach me by 2 October 1996. If there are any changes to the draft agenda, the final agenda will be sent to members in the 14 days before the AGM.

The chair's term of office has a further year to run.

As for other vacancies on the Committee, Clause 10 of the Constitution implies that at most 3 members of the present Committee may continue without being re-elected. Thus Clause 9 implies that there are at least 8 vacancies to be filled.

The following parts of Clause 10 describe the procedure for filling these vacancies:

> (3) Any two members who are individuals may nominate any member or members of the group who are individuals to fill any of the vacancies by giving at least 7 days notice in writing to the secretary
>
> (4) If there are more nominations than vacancies then the membership will be decided by an election by single transferable vote in which every member of the UKTUG who is an individual and is present at the annual general meeting shall be entitled but not obliged to vote

Nominations for committee members in accordance with the above sub-clauses are thus being sought. Written nominations for new committee members should be sent to me, to reach me by 9 October 1996. There is no official nomination form. If you like, you may use the forms provided with this *Baskerville*. Otherwise, what you send me should contain, clearly and legibly,

—name and address of nominee
—declaration that (s)he is willing to stand for this office, signed by the nominee
—name and address of the nominator
—statement nominating the nominee, signed by the nominator
—name and address of seconder
—statement in support of the nomination, signed by the seconder.

The nominee, nominator and seconder should all be members of the UKTUG and the last two should be different from each other.

Once elected, the Committee appoints officers from its membership.

In all the above, 'in writing' means 'on paper', not email.

My address is

IT Services
London School of Economics
Houghton Street
London
WC2A 2AE

Carol Hewlett
Honorary Secretary

XIII The UK TEX Users' Group

Next meeting

There will be a meeting in the afternoon of 16 October (after the AGM). The theme will be latest developments in (LA)TEX.

The 1995–96 UKTUG committee

R. Fairbairns	Chair
P. Abbott	Treasurer and Membership Secretary
C. R. Hewlett	Committee Secretary
M. Clark	Meetings Secretary

K. Bazargan; D. P. Carlisle; A. S. A. Jeffrey; S. P. Q. Rahtz.

Book Discounts for UKTUG members

We have arrangements with Addison-Wesley for their well-known TEX-related publications, and with International Thomson Publishing to supply any of the very excellent O'Reilly & Associates Inc. series of books to members.

The agreed list of books, together with the discounted (at least 20%) price, is distributed occasionally with *Baskerville*, but is always available from the Treasurer, Peter Abbott. The quoted price includes the cost of postage and packing.

We are only allowed to offer this service to **current** members of the UK TEX Users' Group and/or members of TUG. Please send your order and cheque (in UK £) to Peter Abbott (address in *Baskerville* masthead). Make cheques payable to 'UKTUG' please. Books from Addison-Wesley are delivered direct but books from O'Reilly will be routed through UKTUG. *In all cases* please notify Peter Abbott by email, phone, fax or letter when books are delivered.

XIV Obtaining TEX

edited by Carol Hewlett

From the network – CTAN

The UK TEX Archive on `ftp.tex.ac.uk` is part of the CTAN (Comprehensive TEX Archive Network) collaborating network of archives on the Internet organised by the TEX Users Group.

The CTAN archives run an enhanced *ftp* server which supports dynamic compression, uncompression, and archive creation options. Fetch the top-level file `README.archive-features` for information. The server also supports site-defined commands to assist you. Please read `README.site-commands` for a brief overview.

Please report any problems with CTAN archives via email to `ctan@urz.Uni-Heidelberg.de`.

The main directories which make up CTAN are listed below; readers are referred to Graham Williams' *TEX and LATEX Catalogue* which is available from CTAN as `help/Catalogue/catalogue.html`

biblio bibliography-related files, such as BIBTEX.
digests back issues of TEX-related periodicals
dviware contains the various dvi-to-whatever filters and drivers
fonts fonts, both sources and pre-compiled
graphics utilities and macros related to graphics
help overviews of the archive and the TEX system
info files and tutorials which document various aspects of TEX
indexing utilities and related files for indexing
language material for typesetting non-English documents
macros macros packages for TEX and style files
support programs which can be used in support of TEX
systems complete system setups, organized by operating system
tools the various archiving tools used on CTAN
web contains WEB-related files and utilities

Unix – CD-ROM

GUTenberg and UKTUG, in collaboration with TUG and NTG, have produced a plug-and-play CD-ROM based on Thomas Esser's teTEX distribution. As it uses the ISO 9660 standard, the platform-independent files can, in principle, be read on all operating systems which are compatible with that format.

Unix executables for the following platform/operating system combinations are included: Digital alpha-osf (2.0 and 3.2), Hewlett Packard hpux (9.01 and 10.01), Intel i386 bsdi2.0, freebsd (2.0.5 and 2.1.0) netbsd (1.0 and 1.1), Intel i486 (linux and linuxaout), m68k (linux, linuxoldld, and nextstep3), mips (irix 5.2, 5.3 and ultrix4.4) IBM RS6000 (aix3.2 and aix4.1.1) Sparc Solaris (2.4 and 2.5) and Sunos 4.1.3.

For full details see the article in *Baskerville* 6.2.

The CD is available to members of TEX user groups at £15 and to non-members at £25. See the section 'PC and Mac disks' for ordering details.

Unix tapes

David Osborne is no longer able to supply Unix TEX tapes, partly because there is no longer a reliable, up to date, master to copy.

DOS – CD-ROM

UKTUG distributes the comprehensive 4AllTEX CD-ROM, created by the Dutch TEX Users' Group (NTG), now in its 3rd edition. This costs £25 for 2 CDs, and is for DOS users.

PC and Mac disks

The UKTUG distributes an emTEX kit for PCs, and an OzTEX kit for Macintosh. The cost covers copying and postage costs, together with the shareware fee for OzTEX (and other Mac programs) and Eddi4TEX. Each set costs £30, and is available from Peter Abbott, 1 Eymore Close, Selly Oak, Birmingham B29 4LB. Cheques must be payable to