Shells for TEX

Phons Bloemen

Information & Communication Theory
Dept. of Electrical Engineering
Eindhoven University of Technology
The Netherlands
phons@ei.ele.tue.nl

Abstract

This is a review of two 'shell' programs for emTEX, the PC-TEX implementation of Eberhard Mattes. The 'shells' provide an integrated development environment for making TEX documents. The two 'shells' presented here are TEXSHELL by Jürgen Schlegelmilch, and *4TEX* by Wietse Dol, Erik Frambach and Maarten van der Vlerk.

1 Introduction

The well-known emTEX package of Eberhard Mattes has brought TEX and its companions into the reach of the PC-compatible computers. Besides a good implementation of TEX and METAFONT (even for various processors like 8088, 80168 and 80386, which make use of additional features like 'protected mode', and special 'big' versions), emTEX offers an excellent previewer, which works on various graphic screens from Hercules to (S)VGA. There are also printer drivers for HP Laserjet and for nearly all dot matrix printers (by making a little configuration `.dot` file). A dozen of handy utilities come with the package, like BiBTEX, MakeIndx, TEXcad, `mfjob`.

Because this is a collection of various programs, some way of integration must be found, especially for users who want to make TEX their main document preparation system, and do not want to be bothered by MS-DOS commands. Furthermore, an integrated environment consisting of an editor with function keys for the various TEX operations and utilities would greatly improve the acceptation of TEX by a large group of users, especially in 'WORDPERFECT country: The Netherlands'.

I reviewed the two environments by installing them on my home emTEX installation, and on a 'clean 1990 emTEX distribution'. The latter is the distribution of 6 high-density disks of emTEX, as it was released in 1990. Since, there have been various updates on several programs, but more about that later. My own emTEX distribution runs on a 386/25Mhz clone, and is up-to-date, with the beta-test releases of TEX, `dvidrv` and METAFONT installed. I also installed babel and NFSS, and a large collection of fonts, and the AMSPELL spell checker. The packages were also tested on a 286/8Mhz clone with Hercules display. After installation, I used the two environments to make some documents (including this one).

2 TEXShell

TEXSHELL is an integrated environment for the emTEX family, authored by Jürgen Schlegelmilch. It is based on Borland's Turbo Vision, and it looks a lot like the integrated environment of Borland's programming languages like TURBO C and TURBO PASCAL. Version 2.6 comes in a `.ZIP` archive of 619 kilobytes. It can be obtained from various FTP-servers, like `ftp.uni-stuttgart.de`. The archive contains a German and an English version. Version 2.6 is released in september 1993. Most of the evaluation is based on version 2.5.2.

2.1 Installation

The installation is simple: unzip the archive and the files will be placed in the correct directory in the `\emtex` hierarchy. Then the file `\emtex\TEXSHELL.CFG` must be edited to record the drive on which it is installed. It is possible to install TEXSHELL in a user-defined directory, but this is not recommended. No files of the already installed emTEX need to be changed. Start the program with

```
c:\emtex> texshell
```

Now the TEXSHELL manual advises to check the *Options* menu, to examine and adjust the other options to your own needs. The *Options* menu has a consistency check built in: if you enter something impossible (a reference to a program you do not have, or an incorrect path name), the input window will not disappear, and the cursor keeps blinking on the error. The preloaded configuration is sufficient for a base installation of emTEX (i.e. the '1990' distribution, with the german format files).

2.2 Customization

If you want to change the external commands, or install new ones, you have to dig into the manual. TEXSHELL is very powerful in calling external programs, but it takes some study to understand how to make such a call.

There are a few predefined menu options for the 'standard' emTEX utilities such as `dvidrv`, BiBTEX, TEXcad, MakeIndx, these also have their own predefined 'hotkeys'. With respect to the dvi-drivers: it is possible to install a screen previewer, a draft and a final printer (in most cases: `dviscr, dvidot, dvihplj`). This flaw has been corrected in version 2.6: here you can install up to 32 printing devices, and a screen previewer. There is also a *User* menu, where you can install your own programs: I tested this with AMSPELL.

For all external programs, the installation procedure is the same: you can choose from a series of *templates* to construct a command-line to call external programs like TEX itself, `dviscr`, `dvidot` and so on. These 'templates' depend on the filename being edited, the format file used (TEX, LATEX) and various directory names you can install in the *Options* menu. Furthermore, you can assign an *environment* to each call. Settings in this environments are provided as DOS environment variables to the program when it is called from TEXSHELL. If you call a batch file as external command, the COMSPEC variable must be set in the environment (this is not mentioned in the manual)! The process of installing your own tools is tedious, and there are a few caveats.

For the 'bare metal hackers': the configuration file `TEXSHELL.CFG` is a plain ASCII file. It looks a lot like a Windows `.INI` file, and you can edit it to your wishes.

2.3 Editor

The editor provided by TEXSHELL is much like a standard TURBO PASCAL/TURBO C editor. The command structure is like old Wordstar with its Ctrl-K and Ctrl-Q commands, it works very quickly. The editor has a limit of 64 kilobytes per file, but it may have several files opened at the same time. It is possible to transfer text from one file to another using a clipboard and cut-n-paste. There are 10 function keys to put macros under. The editor provides wordwrap, but there is no way to make this the default. It also provides full mouse support for pointing the cursor, moving, sizing, closing windows and menu bar selection.

The package looks very colorful. On the slow machine with the Hercules display, the default 'colors' do not look very well. A little playing with the 'color customization' menu improved things a lot, but the program does not recognize monochrome displays.

There is a nice edit file selection scheme, which again much resembles the one used in TURBO PASCAL. The editor provides a 'primary' TEX file: when TEX is run, it starts with this file. TEXSHELL records its configuration together with the primary file, by writing a `.DSK` file.

2.4 Compile and preview

There is a menu *TeX* on the menu bar which gives access to these functions. Most of them have a 'hotkey'. A very nice feature is an separate menu to view the log file of the command just executed in a *Log Window*, varying from TEX to previewer and BiBTEX. The 'e' option of TEX (provided when there is an error) is fully implemented, and with Ctrl-Q E / Ctrl-Q W you can go to the next/previous TEX error or warning! You can also use the *Log Window* to find out from which file an error originated, by point-and-shoot on the errors in the log file. Great if you are editing a big document, or are playing around with user-written style files.

The previewer can be operated by a 'hotkey' without any problems (only the installation of `mfjob` took some time, but everything works fine).

2.5 Help system

F1 gives context sensitive help: it provides help for the function you have selected. There is 'help on Help' and with Sft-F1 you can browse through an index and view the whole help file. The help files are in English or German. There is also online TEX help: in an edit window you can put the cursor on a TEX command, and get help on that command with Ctrl-F1.

2.6 Manuals

The TEXSHELL manual seems to be 'not yet ready'. It is more a listing of all the features of every window. It contains a little section on what files you should have, and how to install TEXSHELL. Then there is a extended explanation of the use of *templates* and *environments*, and how to install new commands. The last section of the manual explains how to use TEXSHELL: it is a list of command grouped per menu on the menu bar, and a list of the editor functions. Everything is documented, but there is no such thing as a 'guided tour'. There is an appendix containing a revision history.

2.7 Miscellaneous

This section contains a few remarks on other features of the system.

- Installation of new format files is simple when you have generated the appropriate `.FMT` files. However, it is not possible to assign emTEX command line options to each format file. Version 2.6 has a new option to do this, but you can also work around it with batch files.
- TEXSHELL provides good printing support. Before printing there is a menu to enter first and last page, and the number of pages to print. The 'complete' print job is given as a default. You can enter your own `dvidrv` options (if you want to print two A5 sheets sideways).
- TEXSHELL provides no spelling support. With some effort, I managed to install AMSPELL in TEXSHELL, using the *User* menu and a couple of batch files. AMSPELL provides all spelling checking you want.

- TEXSHELL does not support METAFONT when run 'stand-alone' You then have to install it in a batch file in the *User* menu. METAFONT is supported through `mfjob` for automatic font generation.
- I had to change the 'default' `TEXSHELL.CFG` to customize it to my needs (installed AMSPELL, METAFONT, `mfjob`, BiBDB). Contact me for copies of the `TEXSHELL.CFG`) file.

3 4TEX

4TEX is authored by Wietse Dol, Erik Frambach and Maarten van der Vlerk, from the University of Groningen. This integrated environment is based on JP Softwares 4DOS and SemWares QEDIT. The system is a large collection of 4DOS 'batch' `.BTM` files, which perform the various file-handling actions and call the programs from the emTEX collection. To make the system work nice, the authors have written several little utility programs to perform selection functions.The system was written to operate in networked environments, where the emTEX files are placed on a Novell network server, and the users run the programs from the network disk. The review is based on version 2.15 of 4TEX. I will make some references to version 2.20: this is a 'βtest-version' for the upcoming version 3.0 of 4TEX.

3.1 Installation

4TEX was founded as a 'complete' system, containing the complete emTEX package, AMSPELL GHOSTSCRIPT and so on. The archive `obelix.icce.rug.nl` in Groningen has organized the package in several parts: take a look at the various `.TXT` files. The shell itself hides in `4TEX.ARJ`. This package is intended for use on an existing 'standard' emTEX installation. It is about 3 megabytes, and contains the 4DOS `.BTM` files, updates to various emTEX utilities, and the shareware distributions of `PKZIP`, QEDIT and 4DOS version 4. As a bonus, over a 100 new style files will flood your `\emtex\texinput` directory. The 4TEX shell itself is about 300 'compressed' kilobytes, and the 'isolated' shell can be installed on an existing emTEX installation.

Installation is done as follows:

- Unpack the archive in the root of your emTEX drive with `c:\> arj x -v a:4texupgr.arj`. The use of `ARJ` is necessary because the archive spans more than one floppy (and `ARJ` compresses better than `PKZIP 1.1`). The whole package is put into the right directories.
- Some emTEX files must be deleted or moved: the 'compilers' `tex.exe, btex.exe` etc. must be moved to `c:\emtex\compilers`, and the utility programs `bibtex.exe, texchk.exe, makeindx.exe, texcad.exe, texchk.exe` to `c:\emtex\utils`.
- You must put `'LASTDRIVE=T'` in your `CONFIG.SYS`
- You must edit `c:\emtex\btm\system.set` and `c:\emtex\btm\texuser.set` to configure the system to your needs. If you have a 'standard' emTEX, you don't have to touch `system.set`. In `texuser.set` you have to set your personal defaults. Both files provide extended comments on each of the options, but they are quite long at first sight.

The system is started with

```
c:\emtex\btm> tex
```

There are two batch files `tex.bat` and `tex.btm`: if you are already running 4DOS, it will not load it again (However, there are some restrictions to your personal `4DOS.INI` file, and you must have loaded `KSTACK`). If you are running MS-DOS or DRDOS, a new 4DOS command shell will be loaded with the right settings. After a while, the main menu of 4DOS is presented.

3.2 Customization

All customization of 4TEX is done via the configuration files `texuser.set` and `system.set`. There is not much need for the possibility to install 'own' utilities, because there are a lot of utilities predefined, like GHOSTSCRIPT, various graphics utilities like BM2FONT, bibliography maintenance with BiBTEX and BiBDB, spelling check with AMSPELL and much, much more. Even some commercial software like WORD FINDER and EUROGLOT is supported. However, recent versions of 4TEX do offer the opportunity to install own utilities, in user-defined menus.

A system that is made out of 'human-readable' batch files (`.BTM`) is a paradise for 'hackers': if you study the manual of 4DOS (included in the distribution), you can add your own enhancements to the system. Not recommended to keep compatibility with the official release, but hard to resist.

3.3 Editor

The editor of the 4TEX system is the shareware editor QEDIT. The authors have defined a set of macros to use with QEDIT. These extra macros provide 'hotkeys' to start the emTEX 'compiler' and the previewer, to start the AMSPELL spell checker, the BiBDB bibliography database program.

An editor help screen is brought up by F1, and it shows the standard QEDIT editor commands, as well as the special extensions. The QEDIT editor is capable of having more files open at the same time. Spell-checking is done by AMSPELL on a word basis in the editor, or on file basis from the main menu. Before AMSPELL is started, you can choose your language.

The mouse is supported in the QEDIT editor for cursor pointing, block definition and window switching. In the main menus, the mouse can be used to point to menu choices. 4TEX also provides the 'primary file' system as described in TEXSHELL. It also records its configuration together with the primary file, using the extension `.OPT`.

3.4 Compile and preview

The emTEX 'compilers' are started from the main menu in most cases. There are also 'hotkeys' to start them from the editor. A nice option is to start the emTEX compiler and previewer from the editor with a block selected: then only that block of text is TEXed and shown. 4TEX is intelligent about choosing the right documentstyles for such a text block: it examines the primary file and puts its preamble together with a surrounding `\begin{document}` and `\end{document}` around the text block. The error processing is not as good as that of TEXSHELL: the `'e'` option works, and you can view the log files, but that is all. The error checking will be improved in version 3, according to the authors.

The complete 4TEX system comes with LATEX format files with babel and NFSS installed. The format file contains hyphenation patterns for English, German, French and Dutch. In emTEX's memory organization, they set aside 65K words for hyphenation patterns where 36K should be enough according to my information. This leaves little main memory to do the real TEX jobs in (like processing elaborate tables and figures). So the notorious message `TeX capacity exceeded` appears (too) soon.... Version 2.20 has the capability to generate new formats 'on the fly' where you can choose the languages to include from a menu. It then makes a format with just enough space for the hyphenation patterns you choose. A TEX with 7 languages loaded is possible!

3.5 Help system

4TEX provides two different help systems. First, there is help available for each of the menus of the system explaining the various options in short. Second, there is a good 'online' TEXhelp system called TEXHELP. This is a popup command which you can access everywhere with Alt-F2, Alt-F3, Alt-F4. It provides help on the TEX command the cursor is on, and you can navigate through its index. TEXHELP is a TSR program, taking 9 Kb of main memory, and 250 Kb EMS / disk swap space. The help is in English, but it is possible to install a help file in a different language.

3.6 Manuals

The 4TEX manuals are very elaborate. In fact, they provide information on every aspect of the whole emTEX system, and all its utilities. Chapter 2 is fully devoted to using 4TEX itself, and to install it. It contains a section of format-file generation, and installing babel and NFSS.

Then the manual covers the following topics in brief, concise sections: the emTEX 'compilers' themselves, the `dvidrv` programs, Postscript, bibliography and index support, the AMSPELL spell-checker, a section on importing graphics in various ways (TEXcad, BM2FONT), some miscellaneous utilities like `detex`, the TSR programs like TEXHELP and commercially available translation programs, and finally something about fonts and METAFONT/`mfjob` font generation. This is more than a manual, it is an example of how a 'local guide' should look like.

The manual concludes with a list of sources for more information, like TEX users groups (TUG, NTG), and distribution lists (`TEX-NL`). An extended bibliography for further reading is added.

3.7 Miscellaneous

- 4TEX provides support for different types of printers. The printer type can be chosen from a menu in the *Output* menu. There it is also possible to enter the printer port, first and last page, and the number of pages to print. You can enter your own `dvidrv` options Postscript printers are supported via `dvips`, and even the use of GHOSTSCRIPT to get the postscript files on a simpler printer is possible (not tested).
  In earlier versions, the printer support was poor, especially for matrix printers: only a 9-pin was supported, and there was no opportunity to install user-defined printers. This was reported to the author, together with a proposal for a better way to do it (4TEX hacking is easy). Version 2.20 contains much better printer support.
- 4TEX provides an extra *Graphics* menu where you can do all types of conversions of pictures to get them into your TEX documents, provided the necessary programs are installed. It also provides extra *BiBTEX* and *MakeIndx* menus to manage bibliography databases and automatic indexing.
- Because 4TEX was initially developed to run on a network, there is an extra menu called *TEXbatch*. Here you can send print jobs to network printers and batch TEX jobs to fast computers on the network.

4 Conclusions

To conclude this review I give a short 'pro-and-con' list for both systems. First I want to remark that the authors of both systems did a good job, and that the systems are still in a development phase. The use of a 'shell' around the emTEX system is very useful, and it can speed up things even for experienced TEX users. It is also a step on the way to make TEX more attractable to the 'average WORDPERFECT user', and to a TEX 'beginners package'. Some attention must be given to the fact that TEX 'beginners' should not be bothered by lengthy installation procedures: it should be 'plug-and-play'.

TEXSHELL in brief:

+ The use of Turbo Vision gives TEXSHELL a professional look.
+ There is an excellent system of tracing TEX errors in your file.
+ The package is rather small (about 300K).
+ Powerful mechanism to call external programs.
+ The package is reasonably fast, even on slow machines.
– Hard to configure. A major drawback, if there is a 'universal' `TEXSHELL.CFG` file which covers the complete emTEX distribution and some related things, this would ease the installation in a great way.
– Lacks 'integration' of some utilities like AMSPELL.
– The manual needs some rewriting (especially if it is going to be used as a beginners guide).

4TEX in brief:

+ The package covers almost everything in the emTEX package, and even more.
+ Excellent manual, discusses also the emTEX utilities.
+ The TEX help is given by a stand-alone TSR program.
+ Spell-checking with AMSPELL is an integrated part.
– The package is slow (annoying on the 286 clone).
– The package is rather bulky, because it is packed with a lot of other stuff. The shell itself is 300K.
– Installation procedure is elaborate.
– The use of 'human-readable' programs makes the system vulnerable to users who 'customize' it by changing the program itself. But this also may be an advantage...

Some notes about TEX and MS-Windows

Wietse Dol*

Landbouw-Economisch Instituut (LEI-DLO)
P.O.Box 29703
2502 LS Den Haag
The Netherlands
w.dol@lei.agro.nl

Abstract

Many people know about TEX compilers and TEX utilities for the MS-DOS PC. However, in the world of WINDOWS there is a lack of knowledge. This note is intended to inform people who like to run TEX under WINDOWS and want all the good stuff that WINDOWS and TEX offers.

1 Introduction

I am a real DOS (4DOS) addict and in principle do not like to work under WINDOWS. I really detest the many mouse clicks one needs to get simple things done. Things that are done with one or two really simple 4DOS commands (e.g. file moving) need many mouse movements. Perhaps I am an old fashioned guy not realizing that the world is changing...

There are two application that make it worth while using WINDOWS. The first one is the multitasking environment of WINDOWS. Of course there are ordinary DOS programs (e.g. Quarterdeck Desqview) that do the same but a nice graphical environment with icons makes WINDOWS a real userfriendly multitasking environment. The second and most convincing advantage of WINDOWS is its graphical interface. There are many excellent graphical packages (e.g. Correldraw) that allows you the create, manipulate, convert and print all kind of graphics. So when talking graphics one really should work with WINDOWS.

People who work with TEX are all people who like to create texts of the highest quality. The greatest disadvantage (others would say advantage) of TEX is that it is not WYSIWYG (what you see is what you get). After the compilation of a TEX document we all want to use some kind of graphical interface to view (see) the results. So we are talking graphics. We also would like some kind of multitasking, even better a straight compilation and viewing of our TEX code while we are typing the text. This all should be possible with WINDOWS...

An operating system quite similar to WINDOWS for a PC is OS-2 (sorry but OS-2 is much better than WINDOWS). For OS-2 there are several excellent TEX programs and utilities. For example EMTEX (absolutely free of charge!) and ASTEX (see MAPS 93.1 page 41). These programs all have the multitasking and excellent graphical display as mentioned above. People who really like to use some kind of TEX for WINDOWS will find out that there is no such thing yet. There is a commercial package Scientific word wich claims to be a TEX à la WYSIWYG (see MAPS 92.2 page 147). It really looks promising but is not a real and complete TEX system for WINDOWS.

After reading this introduction the question arises 'what should we use under WINDOWS?'. The next sections will discuss some WINDOWS and TEX topics and hopes you to inform how one could set up a WINDOWSTEX system.

2 emTEX and Windows

When looking at TEX PC packages there is one that is state of the art: EMTEX. It is a pity that E. Mattes 'only' developed a MS-DOS and an OS-2 version. EMTEX is free of charge and is to my knowledge the best TEX PC package there is. It offers for the novice and advanced TEX user everything one wants. The only drawback to the system is that E. Mattes did not developed a userfriendly TEX shell. This is nowadays no real disadvantage because there are many good shells (of course I think the best is 4TEX, see the discussion about EMTEX shells elsewhere in this MAPS).

Our first attempt for a WINDOWSTEX should be a DOS-window running EMTEX. Many people who have tried got really disappointed. When running the 386 version of EMTEX under WINDOWS we get the error message 'DPMI not supported.' We can use the slower 8088 and 80186 versions of EMTEX but we want more... The errormessage 'DPMI not supported' is the result of E. Mattes own DOS-extender. Running TEX needs

*I am not a WINDOWS specialist so don't be offended when I do stupid and clumsy things. This note is intended to inform people and stimulate them to write more about TEX and WINDOWS.

a lot of memory or diskswapping. When you have a 386 PC or higher with a lot of memory (the manuals suggest 3Mb) the DOS-extender will claim all available memory and use this instead of the slow diskswapping. The DOS-extender will also use the fast 32-bits processor optimally and gain a lot of speed. The second advantage of the 386 version of EMTEX is that it uses memory settings of the BiG-TEX and BiG-LATEX versions. This means that you can compile large documents with many references and labels.

There is nowadays a solution to the DOS-extender problem. Perhaps not the perfect solution but it works and I think in the near future will become idiot proof. Mattes wrote a program (`emxbind -x`) that deletes his own EMX DOS-extender from the `tex386.exe` and adds a small program (`emxl`) to `tex386.exe`. Now `tex386.exe` will by default look for Mattes own DOS-extender (`emx.exe`) on the DOS-path or will look in the environment variable EMX what DOS-extender to use. There are two possible memory-extenders you can use

1. The DOS-extender `emx.exe` (version 0.8f), i.e. E. Mattes own DOS-extender. This will only work under DOS and not under WINDOWS. To use this DOS-extender set the environment variable EMX to
   `SET EMX=C:\EMTEX\COMPILER\EMX.EXE`
2. The DPMI-extender `rsx.exe` (version alpha 0.51) written by R. Schnitker. This will only work under WINDOWS and not under DOS. To use this DOS-extender set the environment variable EMX to
   `SET EMX=C:\EMTEX\COMPILER\RSX.EXE`

So, simply by updating your `tex386.exe` and adding two memory-extenders to your EMTEX system you now can use EMTEX under DOS as well as under WINDOWS. The DPMI-extender is still in a testing phase (an alpha release), so there could be some bugs. Indeed some bugs are already documented and will be solved in the near future. I have noticed that after running EMTEX under WINDOWS my log file as well as my dvi file and several others became read-only. No serious problem because with a simple DOS statement these files are no longer read-only (i.e. `attrib -r *.dvi *.log`). A simple batch file solves these inconveniences. All the testing I did under WINDOWS didnot result in crashes or strange error massages. The errormessages that appeared were all the result of the know bug that some files appear to be read-only after `rsx` has opened them. All these errors are easily corrected.

The conclusion of this section is that people who want a TEX for WINDOWS, should consider to upgrade/install the EMTEX system.

3 WinTEXversion 1.0

WINTEX is a text editor for WINDOWS specialy written for the use with TEX and LATEX. It is written by S. Morin and helps you with all the difficulties of writing documents in LATEX. WINTEX is shareware and only costs $25,-. Not only the 'normal' editing facilities are supported but also some 'tool bars' and 'dialog boxes'.

There are three TEX tool bars, i.e. a Text-bar, a Math-bar and a LATEX commands-bar. With the Text-bar you can easily and graphically select fontsize, font type and all kinds of accentuation. The Math-bar graphically displays all LATEX symbols and helps you selecting the correct statements for all math symbols. With the LATEX commands-bar one can select from a list of all LATEX commands. WINTEX also has some dialog boxes. There are dialog boxes for the mathematical array, the eqnarray environment, the tabular environment and the LATEX preamble. With dialog boxes the making of the above structures becomes easy.

When you open a new document, WINTEX gives you access to dialog boxes that will help you to build the preamble and the style options. With the main dialog box you choose the document style and associated options. Clicking on Page style, Math style or Floating bodies style buttons will open environment specific dialog boxes. Once selected, the options are inserted in the newly opened document.

I could not read many of my own LATEX documents and I really missed the powerful macro possibilities as for instance with Qedit (or TSE). My general conclusion is that the tool-bars and the dialog-boxes makes it really simple to type documents but that there need a lot to be done before WINTEX is a real text editor and TEX tool.

4 DVIwin version 2.7

The DVIwin driver is written by H. Sendoukas and lets you preview and print DVI files under MS-Windows 3.1. Its main advantages are: speed, compatibility with any raster device with a WINDOWS driver, and graphics capability. All screen and printer handling is done through WINDOWS, so it should work on any printer supported by the system. You can insert arbitrary graphics files produced by most WINDOWS applications, or other standard graphics files (e.g. TIFF, PCX, etc.) provided that you have the appropriate graphics filter. The EMTEX specials and the PostScript specials to include graphics, however, are not supported.

DVIWIN is easy to install and also reads fonts from EMTEX fontlibrary files (`.fli` files in the directory `c:\emtex\texfonts`). The font-substitution looks much like the one used by EMTEX. There only difference is that DVIWIN does not allow wildcard characters (e.g. `cm* 150 -> cm* 300` is not allowed).

I really like DVIWIN because every time you switch to DVIWIN, it checks if the dvi-file that is currently displayed is updated. If updated it will load the new dvi-file and position at the same position (page) as the old dvi-file. This makes it really simple to perform the edit-compile-view cycle.

My general conclusion is that DVIWIN is an excellent dvi-viewer and printer for WINDOWS. I think it will be only a question of time or all EMTEX possibilities that are not available yet in DVIWIN (e.g. automatic fontgeneration, specials) can be used with DVIWIN.

5 A TEX for Windows system

In this section I will explain what I have installed under WINDOWS and how I use WINDOWS to produce TEX documents.

First I updated the EMTEX `tex386.exe` and added the two memory-extenders. These programs will be available on the 27 high density diskettes the NTG will distribute as the complete TEX distribution for the PC. I will also try to get them on all the CTAN servers.

The TEX shell I use is 4TEX. This user friendly menu system can be used to perform all (novice and advance) TEXing. 4TEX version 3.00 (promised to be released in november 1993) will be updated so that it can run every thing in a DOS-window when running WINDOWS. The only thing one has to do is to open a new program group, i.e. select in the Program Manger the FILE and then NEW and PROGRAM GROUP and type the new group name (e.g. TeX). After this you can install 4TEX as a new PROGRAM ITEM and use `c:\emtex\btm\4tex.pif` as the COMMAND LINE and use `c:\emtex\btm\4tex.ico` as the icon.

Of course I have installed WINTEX, DVIWIN and some other TEX WINDOWS utilities (e.g. `dvips`, `ghostscript` and `gnuplot` for WINDOWS) as Program items in the same Program group. Now I have a complete TEX system for WINDOWS.

One thing to remember when installing DVIWIN is that the number of files in the `config.sys` should be at least 50 (due to font loading). Also one needs to copy the `*.dll` files from the directory where one installs WINTEX (e.g. `c:\emtex\win\`) to the window system directory (e.g. `c:\windows\system\`). The font substitution file `dviwin.sub` we need to adjust to our own (extra) fonts and then copy it to our font directory (`c:\emtex\texfonts\`). After installing DVIWIN one has to start the program and adjust some OPTION settings. First we change the resolution to 300 dpi and then change the OPTION FONT DIRECTORY to `c:\emtex\btm\texfonts \$rdpi;c:\texfiles\fonts\$rdpi`. Now we are ready for action. . .

DVIWIN also has two nice utilities. The program `clipmeta.exe` can be used to take a metafile or a bitmap file from the system clipboard and save it to a disk metafile. E.g. you can use `gnuplot` to make nice plots and then paste then to the clipboard and convert this with `clipmeta.exe` to a metafile. The program `wbr.exe` is a text file browser under WINDOWS. It is for instance used in combination with DVIWIN to display the log files.

When I use TEX I first start a 4TEX session and at the same time a DVIWIN session. I use the menu of 4TEX for all TEXing and switch to DVIWIN whenever I want to view and print the document. 4TEX has much to offer what is not yet available under windows (e.g. automatic fontgeneration and many other utilities).

The general conclusion is that in the world of WINDOWS much is on the move. It will only be a matter of time and there will be a perfect TEX system for WINDOWS. Especially the inclusion of all kinds of graphics and the mulitasking (even better real time display while typing) will be possible under WINDOWS. Perhaps all this will not be possible under DOS 7.0.

A Future for TEX*

Roger Hunter

TCI Software Research, Inc.
1190 Foster Road
Las Cruces, NM 88001, USA
roger@nmsu.edu

Abstract

The future of TEX is invisibility. The role of TEX should be similar to that of the microprocessor in a PC. The microprocessor is the heart of the system, but is completely invisible except for the sticker which says 'intel inside.' TEX must be made invisible with appropriate front-ends. These front-ends should emphasize the manipulation of content over appearance and reverse the trend toward WYSIWYG (What You See Is What You Get) interfaces with their emphasis on manipulation of appearance. Content-oriented interfaces provide far greater user productivity than WYSIWYG systems, and TEX is the ideal basis for such systems.

1 Introduction

TEX has a guaranteed future only if its use grows significantly. That growth can occur only if TEX is made much easier to use than it is now. Back-ends to TEX are necessary for any form of output so there are many of them. There must be strong pressure to create front-ends that make TEX much easier to use. The onslaught of WYSIWYG clickery makes the survival of TEX entirely dependent on good front-ends.

2 The Good and the Bad

Listing the good and bad features of TEX seems to be a favorite pastime of TEX lovers, and I am no different. The main difference in my list is that features often considered advantages are listed as disadvantages. First, the good features of TEX. The primary goals of Don Knuth's original TEX project head the list.

1. TEX produces superb output. This was Don's primary motive when he set out to create TEX.
2. TEX source is archival. The documents are in a standard ASCII form. The TEX language provides a linear, ASCII form which can be used as a standard for storage and interchange.
3. TEX is available on most platforms. This, together with its archival nature, ensures that TEX documents can be created and used anywhere there is a reasonably capable computer.
4. Many scientific journals accept compuscripts in TEX and provide style files.

And now the disadvantages.

1. The TEX language is a compromise. It has been said that the TEX language is understandable by everybody because instructions are written in plain English, not undocumented numerical codes. If Don Knuth had felt that the TEX language was the way we should read and write mathematics, then there would have been no need to create TEX, the program. Simply specifying the language would have been enough. The only justification for the form of the TEX language is as a linearized portable input to TEX, the program. In its present form, it is a compromise between the need to provide some support for direct entry and the need to process the result by computer. It would be wonderful to remove this compromise in favor of computer processing, but it is probably much too late.

The two-dimensional mathematical notation evolved because it optimizes the use of the high-bandwidth human optical system. There are many mathematical expressions which are virtually impossible to grasp in TEX input form—all are instantly comprehensible in TEX output form.

Publishers had hoped that TEX would be the solution to the rising cost of typesetting, and now they are not so sure. A major reason for this is that authors do not write style-independent TEX code. Leslie Lamport defines and discusses visual design and logical design in the LATEX *User's Guide & Reference Manual*. Logical design is the key to good TEX documents, but without a way to enforce it, most authors end up with a large component of visual design in their code. This is a nightmare for publishers who need to typeset using a specific style, and is the main reason why author submissions are so costly. It invariably costs more to use the author's original TEX document than it does to re-key the entire thing. If the publishers will not champion TEX, the future cannot be bright. Stamp

*Published in TUGboat **14.3**, 183–186 (1993).

out abominations like `\it word \rm`!
TeX will have no future unless authors are completely isolated from the TeX input language. Although it is archival, the TeX language is unfit for humans. A proper front-end eliminates these problems.
2. TeX is in the public domain. As wonderful as this may seem at first, it means that now that its creator has stopped working on it, everybody wants a say. We have gone from a committee of one to a committee of the entire world. How much progress can a committee of this size make?
3. TeX is extensible. This is marvellous for developers of macro packages and styles. It is a disaster in the hands of authors. Authors delight in creating new sets of macros and in using them inconsistently in a document. Publishers find it much cheaper to rekey an entire document than to rewrite an author's macros to fit a style.
The policy of the American Physical Society, the American Institute of Physics and the Optical Society of America on submitting documents is the right one. To have a paper accepted, you must use the REVTeX styles, but most importantly, you are prohibited absolutely from defining and using macros. Amusingly, the REVTeX guide carefully explains that there are two classes of macros, and then states flatly that you cannot use either kind!
4. TeX is stable and unchanging. Whatever the arguments or the reality, making this statement gives TeX a dead feel. Even if TeX itself does not change for the forseeable future, the continued development of packages like LaTeX3.0 provide the necessary life. This is mostly a matter of public relations.

3 TeX Must be Invisible

TeX is the microprocessor, LaTeX is the operating system, and appropriate front-ends are the application programs. Just as the average user has no need to know how a microprocessor works, and a user of an application program needs only a rudimentary knowledge of the operating system, the average user should never be exposed to TeX. These days, most drivers of cars do not know how an engine works. Although knowing how an engine works may somehow make you a better driver, requiring that you know how an engine works would be ridiculous. It would ensure that most people would not drive. We have roughly the same situation for TeX. Requiring that users know TeX will ensure its demise. A technology is mature when most of its users do not know how that technology works. We should strive to make TeXnology mature.

4 The High Cost of Visual Design

A recent study estimates that 2% of the United States gross domestic product is lost through unproductive use of computers. At the head of the list of offending behaviors is 'font futzing'—endless fiddling with the appearance of a document. A section head at Sandia National Laboratories told me that his researchers spend huge amounts of time preparing reports using WYSIWYG Windows word processors. They spend most of the time changing fonts and page layout. When the documents are submitted, they must be reformatted to fit the required style. The process takes hours because all of the formatting is local and visual.

The same effects exist in the TeX world. Most of us are familiar with people who fall in love with TeX and run around saying, 'Look at this incredible effect I just produced' or, 'Look at this fantastic macro I created.' Highly paid professionals endlessly playing with TeX macros to get just the right visual effect are wasting their time doing work that is unproductive and for which they are not trained. The only way to avoid this problem is to provide a front-end which enforces or strongly encourages the principles of logical design.

5 Interface is Everything

Given that invisibility of TeX is essential and that logical design has a large productivity payoff, interface is everything. Attractive interfaces are the reason WYSIWYG word processors are simply taking over. They are addictive. Their addictive nature and their total focus on visual design makes them one of the most insidious productivity sinks in existence today. The salvation of TeX lies entirely in the development of good interfaces, and those interfaces must encourage and, if necessary, enforce logical design over visual design.

The main reason for using TeX instead of one of the leading word processors is to obtain the far superior output. Because there are no alternatives, you are willing to put up with the input language. The situation with symbolic systems like Maple and Mathematica is the same. The benefits of these systems must outweigh the disadvantages of their unnatural user interfaces before someone will choose to use them. This restricts use to a tiny fraction of the potential audience. By making the interface much better, the number of TeX users could be increased several orders of magnitude. The same argument applies to Maple and Mathematica.

6 The Right Interface

The essential features of a good TeX interface are as follows:
- The interface must encourage authors to work directly at the computer.
- The interface must encourage logical design over visual design.

The penalty for using the computer over the blackboard or a pencil and paper should be minimal. The language you use to read and think about your document should be the language you use to enter it into the computer.

The time taken entering TEX codes is wasted time which could be used for developing the content.

The current crop of Windows-based word processors (Word for Windows, Ami Pro, and Word Perfect are the three most popular) define WYSIWYG. The essential feature is visual design. One manifestation is that you are encouraged to interact with an image of the printed page. Another is that you select text and give commands which determine the appearance such as the font face, point size, and weight.

The fact that all of the best-selling word processors use a WYSIWYG interface has lead to the perception that there is no other way. In fact, the use of a GUI (Graphical User Interface) has become synonymous with WYSIWYG. The result is that millions of people are forced to view crude representations of the printed page through screen windows which never match the pages. At the same time, they have come to spend much of their time at the computer worrying about page layout and typography.

Interfaces which emphasize logical design provide a much better way to create, edit, and interact with documents. The main features of a logical interface are as follows.

- Lines are broken to the screen window.
- You select text and designate it as a section head or apply an emphasis.
- Fonts and colors used on the screen are chosen to maximize screen readability and are independent of the choices made for the printed output.

Just as there is a perception that GUI implies WYSIWYG, there is a corresponding perception that logical implies linear. People seem to think that an interface which uses logical design requires that you enter obscure codes to get the results you want. The primary example in the TEX world is the notion that using the TEX input language directly is the only right way. This is simply false—it is possible to create a logical interface which displays and has you interact with mathematics in its natural (TEX output) form.

7 Some Interface Issues

TEX is a batch system. There are a number of interesting problems which arise when you consider implementing a much more interactive system.

The first problem has to do with TEX's line breaking algorithm. I have often heard people say that the ultimate system would allow you to interact with pages in the way you do with a WYSIWYG word processor, but the page layout would be updated instantly using TEX. Even if you translate this desire to a logical system, there are drawbacks. For example, you could be typing or editing toward the end of a paragraph and have all of the lines above you in the paragraph jiggling about as you type. This is because TEX's line breaking algorithm can change the breaks throughout a paragraph when you make a change anywhere in the paragraph. The effect could be very distracting.

Another question which simply doesn't arise in a batch system has to do with spaces. Who owns the spaces? When TEX puts extra space around operations, relations and punctuation in batch mode, the question makes no sense. When you are dealing with an interactive system, the insert cursor must be placed somewhere, and the choices made have a significant effect on the feel of the system. For example, where should the cursor be placed as you move through the expression $x + y$? TEX inserts extra space around binary operations. Should the cursor position between x and $+$ be next to the x, next to $+$, or somewhere in between? If you take the position that the $+$ owns the extra space, then the cursor should be placed next to the x. This seems like a very minor point, but it has a large effect on the feel of the system.

Blue Sky's Lightning Textures provides a way to enter TEX codes and see the resulting TEX output almost instantly. This is a completely different approach which I view as complementary to the interface I have described. Lightning Textures provides the greatest value for typesetters and others performing high-end layout work. The interface I have described is meant for authors.

8 Scientific Word

Scientific Word is a Windows-based scientific word processor based on the principles that I have mentioned. It provides a logical interface to documents and stores LATEX files. It includes Richard Kinch's TurboTEX for previewing and printing.

Experience with users of *Scientific Word* has been very interesting. Initially, many users feel extremely uncomfortable with the fact that they are not interacting with a page image. They spend a great deal of time previewing to see if they really will get the results they want. As they continue to use the system, the frequency of previews decreases. Once they have learned to trust the system, they relax and focus on the content exclusively. Only in the final stages do they concern themselves with the printed form. The habits developed by using WYSIWYG systems are difficult to break, but once they have been broken, users realize how much more productive they can be.

Direct interfaces between *Scientific Word* and symbolic systems are also being developed. An experimental version of *Scientific Word* lets you interact directly with Maple using the same principles employed for TEX. Maple's input language is invisible in this system—the notation for input and output is the standard, natural, mathematical notation.

This page contains a MS-DOS diskette with the Scientific Word Demo in Hypertext.
The installation instructions and first help is included.

Scientific Word, . . . een eerste indruk

Philippe Vanoverbeke

Langenhoekstraat 21B,
B-8210 Veldegem
+32 50 279853

Abstract

Eind 1992 kreeg ik een β-versie van Scientific Word in de bus. Dhr. Kruger[1] was toen zo vriendelijk om mij dit pakket toe te sturen ter evaluatie. Hierna volgt het rapport(je) met mijn bevindingen dat ik hem destijds toestuurde. Zonder twijfel is het product inmiddels verder geëvalueerd en zijn mijn bevindingen dan ook hoogswaarschijnlijk niet echt meer 'up to date'.

1 Testomgeving

SciWord werd geëvalueerd op de volgende configuratie:

- Atari PC-4 16Mhz (in feite een Mitac 286 PC);
- 3Mb RAM intern;
- 60Mb + 125Mb extern;
- Windows 3.0;
- DOS 5.0;
- Printers:
  - HP Laserjet IIIP
  - HP Deskjet 500
  - Citizen Swift 24nld

2 Installatie

SciWord schept geen problemen bij de installatie. De manier waarop de installatiecyclus vertoont wordt naar de gebruiker toe mag trouwens als voorbeeld dienen voor veel andere (Windows)applicaties.

3 Gebruikersinterface

De gebruikersinterface verdient alle lof. De logische opzet laat toe om met weinig of geen voorkennis van TEX of LATEX wiskundige documenten op te maken van een hoge kwaliteit.

Om dit toch even na te gaan heb ik mijn echtgenote even aan het 'werk' gezet. Zij heeft weinig ervaring met LATEX maar kon toch binnen de kortste keren een blad met formules intikken en afdrukken. Mijns inziens is dit de grootste troef van SciWord, t.t.z. de leerdrempel – die bij TEX en/of LATEX eerder hoog is – is hier dermate laag zodat een leek snel aan het werk kan.

4 Importeren bitmaps

Hier is er een grote lacune. Men kan enkel `EPS` en `HPGL` bitmaps importeren. De opname van tekeningen in TEX-documenten is belangrijk als ik mag afgaan op de vragen die mij regelmatig worden gesteld binnen de NTG. Zelf gebruik ik continu tekeningen binnen mijn documenten (opmaak van cursussen e.d.) en het is dan ook wenselijk om diverse formaten te kunnen opnemen. EPS is interessant in zoverre dat men over een PostScript printer beschikt (of Freedom of Press, of GoScript). Gezien de kostprijs ervan beschikken slecht weinig (privé)gebruikers over deze mogelijkheid. Op zijn minst zou de mogelijkheid moeten bestaan om het `PCX` formaat te importeren èn de tekening in kwestie te kunnen vertonen bij het previewen.

Bijkomend punt is dat SciWord binnen Windows gebruikt wordt. Het zou dan ook mogelijk moeten zijn om de meest voorkomende formaten (van de meest populaire software paketten) te kunnen opnemen. Ik denk hier aan de eerste plaats aan Excel en Coreldraw. Sciword zou zich niet mogen beperken tot 'Wiskunde', veel gebruikers wensen diagrammen (Excel e.a.) en tekeningen op te nemen. Zolang dit niet mogelijk is zullen producten zoals Ventura en Pagemaker verkozen worden boven Sciword gezien de polyvalentie bij deze producten om 'vreemde' tekeningen op te nemen in de documenten.

Overigens biedt SciWord de mogelijkheid om een tekening zonder veel poespas goed te positioneren, wat bij TEX of LATEX geenszins het geval is.

5 WYSIWYG en fonts

Op het scherm worden inderdaad ATM fonts vertoond, terwijl voor de afdrukken CM fonts gebruikt worden. Dit vond ik niet storend.

Wel zou het m.i. wenselijk zijn om over de mogelijkheid te beschikken om andere PS fonts te kunnen gebruiken (Avantgarde, Palatino. . .) en dus documenten te kunnen afdrukken (en previewen) met deze fonts.

[1] M.J. Krugers, Technical Marketing Consulting, Gagelweg 3, 4651 VL Steenbergen; distributeur van Scientific Word; NTG lid # 215.

Dit schept problemen m.b.t. het afdrukken van wiskundige formules maar dat mag geen reden zijn om de mogelijkheid niet te bieden.

Binnen de NTG merk ik dat de meesten TEX en LATEX vooral beschouwen als een (exclusieve) tool om mathematische documenten te produceren. Ik kan dit wel bijtreden doch ben de overtuiging aangedaan dat ook de aanmaak van 'gewone' tekstdocumenten enige aandacht mag verdienen. Zelf produceer ik hopen bladzijden documenten met LATEX en voel geenszins de behoefte om over te gaan naar een ander software pakket voor gewone tekst (met EmTEX in combinatie met DviPS bestaat wel de mogelijkheid om PS fonts te gebruiken).

Wat SciWord betreft kan ik mij inbeelden dat de meeste concurrentie te vrezen valt van producten zoals Ventura, Pagemaker, Word etc. (en niet van EmTEX en vergelijkbare software). Deze bieden veel mogelijkheden op het vlak van fonts wat binnen de huidige opkomende DTP-rage een belangrijk voordeel is.

6 Compatibiliteit

Ik heb tal van eigen (LATEX) documenten geladen binnen SciWord. Dit stelde voor kleinere documenten weinig of geen problemen. De style-sheets die ik zelf gebruik (meestal via de NTG verkregen) werden moeiteloos geassimileerd door SciWord.

Wel bleef mijn computer 'hangen' bij het inlezen van langere documenten (80 à 120 blz). Een thesis van 90 blz. (eenvoudige LATEX file) was na 2 uur (!?) niet ingelezen. Deze problemen kunnen misschien te wijten zijn aan het beperkte geheugen (3Mb)?

In dit verband zou het trouwens nuttig zijn dat SciWord een 'tijdsindicatie' zou vertonen bij het inlezen van documenten (bv. een tijdsbalk die de voortgang weergeeft in procenten). Dit zou vermijden dat de gebruiker in het ongewisse blijft over de gang van zaken. Niet iedereen maakt documenten van een twintigtal blz. TEX biedt nu eenmaal het voordeel om grote documenten (boeken...) aan te maken. SciWord meldt wel het aantal 'verorberde' paragrafen doch dit is geen echt bruikbare informatie.

7 Afdrukken

Enkel maar lof in dit verband. De mogelijkheden welke geboden worden qua afdruk zijn goed.

Ook de conversie naar PostScript (Apple Laser Writer) gaat als een trein en schept voor de gebruiker geen enkel probleem.

8 Bugs

Ik heb enkele problemen gedetecteerd welke te wijten zijn aan of een bug of een beperking van SciWord.

- Een lang document (80 blz.) inlezen lijkt onmogelijk (of duurt een eeuwigheid);
- Nadat ik Chemdoc.tex (geleverd als voorbeeld bij SciWord) had gepreviewed en afgedrukt bleek de originele file 'leeg' (0kb)!! Mocht mij dit gebeuren bij een (groot) eigen document zou ik de wanhoop nabij zijn (of erger). Wel moet ik vermelden dat mijn harde schijf (de betreffende partitie) quasi vol was toen dit zich voordeed;
- Na een document 2 maal door LATEX gehaald was had ik nog altijd geen inhoudstabel (enkel de titel);
- Wanneer men een `EPS` tekening opneemt en een document afdrukt (dvi, niet alw) wordt deze niet afgedrukt. Dit is logisch vermits er niet wordt geconverteerd naar PostScript doch een melding naar de 'niet-geroutineerde' gebruiker zou hier op zijn plaats zijn dacht ik.

9 Gewenste verbeteringen

Er zijn aan dit – overigens verbazend goed – product wel enige verbeteringen toe te brengen.

- Vooreerst (en dat vind ik persoonlijk essentieel) zou de mogelijkheid moeten bestaan om bij het previewen de gehele bladzijde te kunnen bekijken. Als deze mogelijkheid bestaat heb ik hem in iedere geval niet ondekt. Wanneer men een document opmaakt is het m.i. essentieel om blad per blad het document te kunnen bekijken. Dit geeft een algemene indruk van de presentatie (witruimtes etc.) en eventuele lacunes desbetreffende. Nu is men in wezen verplicht eerst een afdruk te maken wat een eerder tijdrovende bezigheid is.
- Een spellingscontrole in het Nederlands, maar dat ligt voor de hand.
- De mogelijkheid om PostScript fonts te kunnen gebruiken (previewen en afdrukken).
- Tijdsbalk (of andere indicatie) die de voortgang bij het inlezen van een document aangeeft (ook bij het 'previewen').
- De mogelijkheid om tekeningen in `PCX` formaat in te kunnen lezen. Eventueel ook de mogelijkheid om grafieken/charts/tekeningen van bekende Windows producten te kunnen importeren.

10 SciWord versus emTEX

Zelf maak ik intensief gebruik van EmTEX. Zowel de geboden mogelijkheden van dit pakket als de verwerkingssnelheid op mijn (bescheiden 286) PC schenken mij voldoening. Niettemin heeft het mij een hoop moeite èn tijd gekost om TEX (en EmTEX) onder de knie te krijgen.
SciWord daarentegen biedt aan eenieder de mogelijkheid om de voordelen van TEX snel in de praktijk aan te kunnen wenden. Bij EmTEX is dit zeker niet het geval[2]: voorkennis van TEX is zeker benodigd. Daar-

[2] Toen ik dit schreef was er nog geen 4TEX ...

naast moet men ook enige PC ervaring hebben. Veel nieuwe PC gebruikers (het gros) koopt hedentendage een PC met Windows-omgeving gezien het geboden gebruiksgemak. In combinatie met SciWord kan een 'informatica-onkundig' persoon (dit is niet pejoratief bedoeld) onmiddellijk aan de slag om documenten van een hoge (TEX) kwaliteit aan te maken. Dit is met EmTEX zeker niet het geval.

11 Prijsstelling

De grootste handicap van SciWord zal wel de aanschafprijs zijn vrees ik. Naar ik ervaar is TEX vooral 'populair' bij universiteiten (rekencentra...) en studenten. Deze laatsten beschikken vaak niet over een budget die een dergelijke aanschaf toelaten.

Anderszijds is het mogelijk dat door het geboden gebruiksgemak SciWord binnen andere milieu's ingang vindt.

12 NTG vereniging

De NTG vereniging streeft ernaar (dacht ik toch) om TEX c.q. LATEX een grotere bekendheid te geven bij computer gebruikers in het algemeen. De hoge leerdrempel van het product ligt m.i. echter aan de basis dat het niet echt aanslaat (ondanks nu ook goede PC, Atari en Mc Intosh versies beschikbaar worden).

Indien NTG leden SciWord zouden 'bekritiseren' omdat het niet 100% puur TEX is lijkt mij dat onterecht. Een gebruiksvriendelijk product als SciWord kan de verspreiding van TEX in het algemeen enkel maar ten goede komen.

13 Conclusies

SciWord is een merkwaardig goed product, de link met TEX is daar natuurlijk niet vreemd aan. De geboden gebruiksvriendelijk is opmerkelijk en tevens van een hoger niveau dan producten als Ventura en dergelijke. Dat SciWord voor de echte TEX-goeroes misschien een vreemde eend in de bijt is, is irrelevant omdat het product — naar ik mag aannemen — niet in de eerste plaats voor deze personen bedoeld is.

— — * — —

Laatste nieuws van het Scientific Word Front[‡]

In de huidige toestand — versie 1.1 komt over enkele weken ter beschikking — kunnen de volgende opmerkingen gemaakt worden m.b.t. het artikel van Philippe Vanoverbeke

- Scientific Word leest nog immer alleen EPS en Laserjet bestanden in en positioneert ze inderdaad goed (beter in versie 1.1). Er zijn geen plannen om deze tekeningen zichtbaar te maken binnen Scientific Word noch om PCX formaat te ondersteunen. Er zijn twee oplossingen:
  1. Maak de tekening in een tekenpakket bijvoorbeeld DesignCad, en importeer het resultaat in Scientific Word. Deze pakketten geven vrijwel altijd zowel Postscript als Laserjet formaat af.
  2. Gebruik een conversie routine van PCX naar Laserjet formaat. De practijk leert dat dit niet altijd succes hoeft op te leveren.
- Er zijn nog immer plannen om een versie te maken die Postscript fonts ondersteund. De makers van Scientific Word zoeken naar een leverancier die een betrouwbare LATEX versie hebben onder Windows met de Postscript font ondersteuning. Deze blijkt echter onvindbaar.
- Grote documenten worden in versie 1.1 goed ingelezen. Maar 3 MB geheugen onder Windows 3.0 is inderdaad heel erg krap. Met Windows 3.1 is zeker 4 MB nodig, maar 6 MB is vereist om met de meeste pakketten betrouwbaar te werken. Overigens is Scientific Word onder Windows 3.1 sneller dan onder Windows 3.0.
- De makers van Scientific Word denken nog immer dat het melden van aantallen paragrafen voldoende is. Ik heb ze niet weten te overtuigen dat te veranderen.
- In versie 1.1 wordt een wat betere check op beschikbare schijfruimte uitgevoerd maar wat krap bemeten ruimte kan altijd op d'een of d'andere manier problemen geven.
- De inhoudstabel wordt nu wel goed gegenereerd, wellicht dat ook hier de 3 MB een rol speelde.
- In versie 1.1 kan nog steeds niet een hele pagina bekeken worden. Wel is de previewer een stuk sneller geworden.
- Er zijn nu spellings controle dictionairs beschikbaar voor vele talen a *f* 100,– per stuk. Dit zijn de dictionairs van Proximity.
- De prijsstelling is veranderd. De prijs is nu *f* 1500,– excl. BTW en tot einde jaar 25% korting voor NTG leden.

In versie 1.1 zijn de volgende veranderingen:

- Zoek en vervang is mogelijk, niet alleen op text *maar ook op formules*.
- Er is een apart venstertje gekomen met de letters met accenten voor Europese talen.
- Er worden zeer veel stylesheets bijgeleverd in Europees en Amerikaans A4 formaat. Ik hoop dat in de naaste toekomst de Kluwer Academic stylesheets standaard meegeleverd zullen worden.
- Er is een postprogramma bij om Scientifc Word bestanden via E-mail te versturen zodat ieder ander LATEX systeem ze kan weergeven.
- De gebruiker kan zijn eigen strings definieren die als wiskundig symbool herkend dient te worden.

[‡]Bijdrage van M.J. Krugers, Technical Marketing Consulting, Gagelweg 3, 4651 VL Steenbergen; 10 oktober 1993.

- Bij iedere style kan een schermopbouw gedefinieerd worden bijvoorbeeld voor weergave koppen e.d.
- Strategisch gezien zijn de makers van Scientific Word veel meer bezig om het product voor mathematisch gebruik te perfectioneren dan om er een algemene tekstverwerker van te maken, zoals uit bovenstaande blijkt. Het beste bewijs is de ontwikkeling die nu aan de gang is om een koppeling te maken tussen Scientific Word en Maple, een mathematisch rekenprogramma. De eerste beta versies laten duidelijk (en betrouwbaar) zien dat in Scientific Word complexe formules ingetikt kunnen worden. Roep dan Maple aan die de formule of evalueert of uitrekent en het antwoord in Scientifc Word plaatst. In tegenstelling tot Maple zelf kunnen ook zo delen van formules bekeken worden.

Ik hoop hiermede de NTG leden weer op de hoogte gebracht te hebben van de laatste ontwikkelingen rond Scientific Word. *Het is inderdaad een programma om een niet-TEX deskundige zeer snel op gang te brengen met TEX.*