General Delivery

Changing TEX?

Malcolm Clark

The topic of a 'future' TEX has been a source of discussion ever since TEX managed to slip out from Knuth's office. And twice there have been 'official' developments to the fundamental model: first with the transition from the original SAIL implementation through Pascal and eventually WEB — to what became known as TEX82 when it was first released; and more recently the mild tinkering (by comparison) which is colloquially termed TEX3. Knuth has made quite clear that he will not change TEX again, and that while he is happy for anyone else to embed the program code in other applications, those applications are not to be termed TEX. In passing I recall how much time I spent between 1984 and 1989 extolling one of the great merits of TEX — that it was fixed and need not be relearned, unlike the tide of of other (and commercial) software; only to have to retract a little in 1989. But this time I'm even more convinced.

But still, there are repeated reminders that TEX is not perfect and that in order to stem the tide of newer, and obviously more desirable, systems for typesetting, document preparation, and electronic publishing, some radical surgery is needed. It seems hardly surprising to me that later systems should not surpass TEX in their quality: I am more surprised that they often fail to do so. For example, given the relative ease with which the mathematical typesetting of TEX might be extracted and used elsewhere, it comes as more of a surprise that even now, few other systems approach the standard which is already 14 years old. In my cynical moments I suspect that the world may not be as concerned with quality as TEX claims to be.

How can these many suggestions for enhancement or improvement be accommodated? In the first place, we have to acknowledge that many 'enhancements' have taken place: from the very beginning there were variants — the shadowy Tyx, and latterly VORTEX, MLTEX, VTEX and so on. They appear to meet Knuth's requirement that they are not called TEX. They each offer some extensions to meet some perceived lack or need (although VORTEX's pedigree is a little different). But what really are the most significant TEX developments over the last ten years? The important development has been LATEX — a change to the user interface, rather than an enhancement of some concept of typographic quality. Like its host, TEX, it is an imperfect tool, but pragmatists recognize its widespread generality and applicability.

The major extensions of BIBTEX and MakeIndex were designed for use with LATEX. If there are two things we need right now, they are 'finished' versions of BIBTEX and MakeIndex for every platform that runs LATEX — and the documentation that goes with them. If the new books which are appearing managed to include these support facilities, they would being doing us all a service. (And if BIBTEX format were to be an output format of the many on-line bibliographies, that would be even nicer.)

In another area of 'support' facilities, it seems to me that one of the great imponderables is slowly disappearing: I almost always know how to invoke (LA)TEX, but figuring out how to magic up the printer driver is often a mystery. As POSTSCRIPT continues its inexorable growth, Tom Rockiki's DVIPS appears to be emerging as the most widely supported driver. This give us the chance of *de facto* standardization and the chance to use the `\special` command without grief (probably to incorporate some graphics). If we couple this with the various applications which convert POSTSCRIPT to what your laser printer or screen can display (and some of these are public domain) we are heading for a device independence which was only a dream a few years ago.

Am I just being complacent? Is it sufficient to accommodate increased functionality and an improved user interface within the present shell? — one of the things which the LATEX3 project should deliver. Should I rather be worrying that it is difficult to create magazines like Newsweek with (LA)TEX? Would I be grateful for a copy of Frame, Interleaf, Grif, 3B2, or even Quark XPress? I quail before the thought of all that re-learning, frustration, the need to keep up with upgrades, the inability to swap documents easily between my Vax, SparcStation, Macintosh and the crufty MS-DOS machine gathering dust in the corner. Will I have to learn how to design documents myself? Life is too short. In future columns I will discuss how Shakespeare's *Sonnets* need drastic re-writing, and how Mozart's *Magic Flute* requires a change in the underlying paradigm to accommodate the developments in rap music and sampling.

But the 'stasis' strategy only retains the existing users: would anyone volunteer to adopt (LA)TEX from scratch? The one area we have done the most effective and consistent job is in telling everyone how difficult (LA)TeX is to use, and what a dreadful typeface Computer Modern is. It used to be the English who had a reputation for under-statement and self-deprecation. (LA)TeXies have easily overtaken them. Why should this be?

Somewhere in this hyperbole serious questions are lurking. To what extent should TUG be pursuing the 'future' of TeX? And which future? If we examine the TUG Bylaws, we will note that TUG was set up to 'identify, develop, operate, fund, support, promote and encourage charitable, educational and scientific programs and projects which will stimulate those who have an interest in systems for typesetting technical text and font design'. The german-speaking group, DANTE, addressed the topic of a future TeX at their Hamburg conference (reported in this issue of *TUGboat* by Phil Taylor), and Rainer Schöpf has since set up an electronic discussion list. There is a paradox here of course: those who do want to change TeX are more likely to participate than those who don't. It will be useful and instructive to see what shakes out of these discussions. There has already been a wide range of opinions expressed, from creeping featurism through to the adoption of new paradigms.

Of course, the choices are not simple, or exclusive. Improvements will take place in the user interface; at the same time, some brave souls will modify the underlying code. If changes are not generally available, and are restricted for proprietary or platform reasons, they are unlikely to be adopted by the present user base: if there is insufficient upwards compatibility, the inertial mass of existing documents may also discourage adoption; the prospect of change is ambiguous — it excites some and depresses others. Consider two examples of the diffusion of changes in the TeX world: the change from Almost Modern to the Computer Modern typeface took an age, perhaps because the changes did not seem noticeable (so much for quality!); the change to TeX3 appears to have been very swift — the lure of 8-bit input and the enthusiasm of the non-English speaking users seems to have been a major driving force here. Interesting times.

◇ Malcolm Clark
Information Resource Services
Polytechnic of Central London
115 New Cavendish Street
London W1M 8JS, England
UK
Janet: `malcolmc@uk.ac.pcl.mole`

Editorial Comments

Barbara Beeton

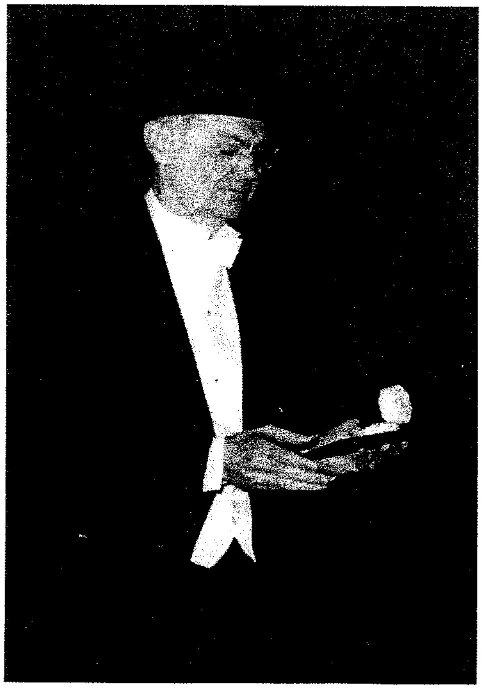
Another honor for Donald Knuth

During a ceremony held in the Stockholm City Hall on November 15th 1991, Donald Knuth was appointed Honorary Doctor of Technology by the School of Computer Science and Engineering, KTH, Stockholm. The appointment was accompanied by this citation.

> Professor Donald Knuth is very well known to us, not only in Computer Science, but also in the fields of Mathematics and Typography. He has through his creative research and his monumental work *The Art of Computer Programming* made major contributions to the modern research area of mathematical analysis of algorithms and their complexity (performance), as well as given the virgin computer science a firm mathematical structure of great importance to undergraduate and graduate studies.

Roswitha Graham, head of the Nordic TeX User Group, has provided the following report.

"Professor Knuth has for a long time had close contacts with researchers within the School of Computer Science and Engineering at KTH, and he is also present daily through his advanced computer tool TeX for production of technical and



Donald Knuth receiving his KTH degree