

# Dreamboat

## NextTeX: A Personal View

Malcolm Clark

### Explanation

Last year, at the Portland TUG Conference, I was invited to give the keynote address. What was printed in the conference proceedings was not what I talked about. This was perhaps a bit arrogant on my part, but since the conference preprints were available to those who wished to read the 'official' paper, I felt that it was not stretching the prerogative too far to talk about something which, at the time, I thought more important to the TeX community. Perhaps unsurprisingly, the talk was mis-reported. Joachim Lammarsch, President of DANTE, the German-speaking heard it as an attack on NTS, the 'New Typesetting System' which his group had initiated. Since Lammarsch expressed his displeasure in DANTE's 'Die Technische Komödie', reported in *TUGboat* 14(1) as 'he (Lammarsch) expresses his strong disappointment over the statements on NTS (...) made by Malcolm Clark', I feel it is appropriate to have the opportunity to see what was actually said. Naturally I cannot guarantee that what I said was exactly what is written below, but it is the text from which I was working (and one which I gave to Lammarsch later in 1992 so that he would have an accurate original which he might use). I have not included all the overhead slides I used, since they were a little too fragmentary, but they do not diverge from the argument developed below. I have corrected one or two grammatical errors, and added the footnotes. Nothing substantive has been changed.

It would have been difficult for me to say anything about NTS at the time, since it had hardly been reported in the English-speaking world, except in an email (NTS-L) list, where the status of the project was not particularly clear. It was not until September of 1992 that Philip Taylor [19] presented a paper at the Prague EuroTeX conference in which details were given on a wider basis, but even this hardly amounts to widespread dissemination. Perhaps Taylor's later exposition at the Aston'93 conference [20] will give the NTS project the exposure it warrants. Joachim Lammarsch [10] also accepted an invitation to talk on the subject.

## 1 Introduction

One of the consistent recurrent themes present at any gathering of two or more TeXies is the conversation about the deficiencies of the program, and the need to enhance by adding a number of features, both to do something in particular, but also to ensure that TeX remains in the forefront of quality technical publishing.

On examination, it often, but not always, turns out that TeX is well able to do the particular task which provided the perceived requirement to enhance the program, but that the code needed to achieve the result is not immediately obvious or intuitive (Spivak gives a good example [18]). There can be no doubt that TeX is a very subtle beast and has depths that few of us will ever plumb. But equally, there are some things which TeX does with great difficulty: a well-known example is the (almost) impossibility of finding out exactly where on the page you are (but see Hoenig's solution [6]). Various people, with a deep understanding of the program, have listed some features that they would like to see enhanced: the papers of Stephan von Bechtolsheim [1], Frank Mittelbach [13] and David Salomon [15] are recent examples, but if we delve back into the TeX literature (exemplified by *TUGboat*), we will find other examples. It is quite arresting to read Lynne Price's words [14]: 'One refreshing quality of the TeX user community, and particularly the system's creator, is that TeX is viewed, in fact intended, to be the ancestor of an evolving family of document formatters rather than as a static piece of software that will be used for decades.' In the same article, I was astounded to note an account of L<sup>A</sup>TeX: 'a hybrid of TeX and Lisp', where text manipulations too difficult or impossible in TeX are done in Lisp. (I had thought I had merely been joking when I had from time to time suggested implementing TeX in Lisp for just this sort of reason!) As a result of this note by Price, proposals for future enhancements were given a column in *TUGboat* — the Dreamboat column (one recently revitalized by Barbara Beeton). In 1987 L<sup>A</sup>mpport [11] bemoans the 'idiosyncrasy' of dvi format and suggests a switch to PostScript.

## 2 Change already

Looking at the problem historically, there have been two major jumps in TeX. But not all jumps are alike: the first change was a major one — the change from TeX78 to TeX82. TeX82 is the one with which most of us who have used TeX will probably be familiar. It survived mostly unchanged save for bug fixes until 1988. The transition from TeX78 to TeX82 was radical. Some of the language primitives

changed: one of the most striking was in font handling: I was fortunate that I learned  $\text{\TeX}$  as  $\text{\TeX}78$ , when the manual was a scant 200 or so pages long. I doubt that I would have started if the manual had been 500 pages long. Internally, the changes were even more marked, since the language was changed from SAIL to Pascal. This also meant that  $\text{\TeX}$  became much more portable, inaugurating a whole new concept in software development.

The other change which will still be in our immediate memory is the change to the so-called  $\text{\TeX}3$ , which began in 1988. The magnitude of the change is much less great than the earlier change. In essence it was to enhance  $\text{\TeX}$  to handle eight-bit characters, instead of the seven-bit characters with which it originated. The immediate benefits of this change were felt mostly with respect to the ease with which accented characters could be dealt with—among other things, making it possible, at last, to hyphenate accented words properly. There were one or two other relatively minor changes too. I have to admit that the transition to  $\text{\TeX}3$  has made hardly any difference at all to me, although I regularly use  $\text{\TeX}3$  on Macintosh, UNIX and VAX/VMS.

In between times, there were a few other changes in the  $\text{\TeX}$  world, although not directly to  $\text{\TeX}$  itself. For example, METAFONT was upgraded in 1984, in rather the same way that  $\text{\TeX}$  had been: in general, the change was hardly noticed by the mass of  $\text{\TeX}$ ies, since they do not use METAFONT explicitly. Similarly the Computer Modern typeface started out as Computer Modern, reverted to *Almost Computer Modern*, and then re-asserted itself as Computer Modern (and as recently as 1992 was still being subtly altered). Those of us around in the days of this transition will recall the confusion caused between those machines which had the *Almost fonts*, and those with the more final version. In particular, PCs seemed to hang on to these older versions.

### 3 We are not alone

Naturally,  $\text{\TeX}$  does not live in noble isolation. In the years since its birth, we have seen a number of notable developments which have produced resonances within the somewhat hermetic  $\text{\TeX}$  universe. The dramatic rise in personal computing power spread the use of  $\text{\TeX}$  widely, and to some extent loosened the ties between  $\text{\TeX}$ ies. Reflect that the  $\text{\TeX}$  and  $\text{\LaTeX}$  books have both sold into the hundreds of thousands. I think that the combined figure is now over 150,000—that's an expenditure of approximately \$5,000,000: if we take that as a crude measure of the number of  $\text{\TeX}$  and  $\text{\LaTeX}$  users (and ex-users), and compare it with the num-

ber of TUG members (about 3,500), and then the number at the recent TUG conference (about 150), we see there may be a lot more people doing it than talking about it (maybe they are too embarrassed to talk about it).

In passing it is surprising just how long it took before the first non-canonical  $\text{\TeX}$  and  $\text{\LaTeX}$  books appeared (my guess is that the first properly published follower was Norbert Schwarz [16], first in German in 1988, and then translated into English [17]). Maybe *The  $\text{\TeX}$ book* really is crystal clear.

### 4 Diffuse

But this takes us away from the main theme I would like to develop. We have a vast increase in the number of users, and the majority have  $\text{\TeX}$  on their own individual machine with limited support from elsewhere. This has quite far-reaching consequences, especially when coupled with the near demise of commercial vendors outside the USA and the widespread availability of public domain implementations. To whom does the user turn? And how does she or he get information about changes and developments? To take a specific example, did you realise that the Computer Modern fonts had been tweaked earlier this year? The sub-text here is that changes may not diffuse too readily. A similar slowness of diffusion rates is experienced with  $\text{\LaTeX}$  styles. The current version of  $\text{\LaTeX}$  is 2.09. Most users seem to have this. But this version number is not sufficient. One must also know the date. The files should be dated February 1991.<sup>1</sup> Experience shows that this is not always the case. Similarly, the complete lack of clarity of the availability and distribution of the New Font Selection Scheme (seldom part of a vendor's offering) bodes ill for the acceptance and widespread availability of  $\text{\LaTeX}3$  (whenever it appears). There is a counter-example in the relative speed with which  $\text{\TeX}3$  appears to have swept around the world.

### 5 Commercial

The rise of personal machines stimulated the widespread adoption of improved printing facilities, especially the 300 dpi laser printer. This was a development on which  $\text{\TeX}$  was well able to capitalise. But it is probably not a development which had been anticipated when METAFONT and Computer Modern were created. Laser printers were seen as low resolution devices used at a stage prior to the final high resolution photo-typesetting. Computer

<sup>1</sup> Wrong! Even at the time of writing, the latest release was March 25th, 1992, but since then  $\text{\LaTeX}2\epsilon$  has been announced at Aston—let's watch its diffusion.

Modern fonts (like very many others) are not ideal 300 dpi fonts (and the even lower resolution screen versions leave much to be desired — sometimes the METAFONT rather falls apart). But the point being embroidered is that this identified two new foci, the laser printer, which quickly became identified with PostScript, Warnock & Geshke's page description language, and then direct manipulation word processing programs. Remember that T<sub>E</sub>X's avowed aim was to assist publishing (masterpieces of the publishing art); the new generation of personal publishing was initially very happy with relatively low resolution laser printed copies. But in time quality and scope improved, up to the level where contemporary publishing packages, like Quark Xpress, PageMaker, InterLeaf, FrameMaker and 3B2 (to name a few) can arguably produce masterpieces.

Commercial software has some interesting qualities: it evolves. In order for the vendor to survive, it is essential that new versions of the software are released, correcting some of the bugs, introducing some new features, and basically keeping the software in the public's eye. T<sub>E</sub>X is not commercial software, except in a very limited sense. It is almost always possible to find a public domain implementation. But there is no development of the core software; there is no reason to keep releasing new versions. The only real exception to this rule is when a version for a new machine or version of an operating system is released.

In order to pay lip service at the altar of fair play, I have to admit that there is software around which is not commercial, and yet which has evolved. Kermit springs to mind, although I am not sure if it is still evolving now. I have versions of Kermit which work for the machines I use, and until they fall over badly, I won't bother replacing them. Much of the Gnu (Free Software Foundation) project's software is also still being developed. If we ignore the forbidding air of messianic fundamentalism surrounding the Gnu project (just as we expect everyone else to ignore our very own missionary position) we have to admit that they do provide a model of public domain software development.

I think there is a difference between this development and T<sub>E</sub>X, or some successor to it. There is a fixed mark, something to aim for: Kermit did develop along the way, but the main issue was to have something which worked on many platforms and performed a reasonably well-defined function. The Gnu project is aiming to provide substitutes for software which already exists (like a C compiler), and is therefore specified already (or even mis-specified already). The T<sub>E</sub>X successor will first have to decide what features it will encompass.

## 6 Quality

One of the arguments put forward for the need to develop T<sub>E</sub>X further is the quest for quality. It is said that there are areas where the highest quality is just not obtainable. I do not wish to challenge this statement, but rather to question the quest for quality. I appreciate that this is heretical. Currently, my organization,<sup>2</sup> a self-styled educational institution, is going through a sort of managerial restructuring. Part of the new baggage of management is the idea of 'total quality'. It is difficult to stand up and say that you do not believe in quality. But as far as a publishing system is concerned, I think it is possible to say that aspiration to the highest typeset quality is not the sole criterion.

I am not sufficient of an aesthete to recognise the highest quality. I think I can often find things which I consider to be pleasing to the eye, but when it comes to qualitative judgments, absolutes are so very elusive. Typographic quality at least has the advantage that there is often a function lurking underneath, and we can always appeal to the extent to which the form and function complement one another, or appeal to notions of 'fitness for purpose'. But sadly it often seems that the consensus for quality is a rather conservative one. Apparently, within a few years of Gutenberg's 42 line Bible being produced, there were vociferous complaints by the cognoscenti bemoaning the sad reduction in quality from traditional hand-lettered manuscripts. And we can see this pattern repeated again and again. We can be relatively confident that a departure from the norm is perceived as bad. In a few years it may become acceptable, but at the time, it is new and suspect. Of course, the iconoclasts will be prepared to pick up the new, for good and bad reasons. But even if we hedge around the problem of identifying the highest quality, we can usually acknowledge that some things are suspect.

But who actually worries? A few years ago, it was common to see typewritten manuscripts published by reputable publishers as whole books. The argument was usually that it was better to have something published at this lower quality than nothing published at all. It does seem to indicate that quality is only one of several issues, even among 'quality' publishers. Even today, using the same sort of argument, we often see books published from laser printed masters (even T<sub>E</sub>X books!). This is sad. The difference in cost is really not great. The publisher, for whatever reasons, economic or aesthetic, clearly feels that typographic quality commensurate with the book's 'worth' may be met with

<sup>2</sup> My ex-organisation!

inferior production. Let me take two contemporary examples. The quality of the paper used in the softback  $\text{\TeX}$ book has deteriorated over the years (in my opinion): I will not rise to the bait of the abysmal binding of the softback; even the hardback is not designed to last for ever—I was very disappointed when my Knuth-autographed hardback fell apart last year. And yes, I do look after books and take great care not to break the spines. Another example would be Victor Eijkhout's recent book [3]. Victor obviously spent a good deal of time and effort in the design of his book, even to the extent of eschewing the delightful (if traditional) Computer Modern typeface. Sadly, at least half the copies I have seen were under-inked. Both these examples emphasise that getting the marks on the paper in the right place is only one of the problems facing us.

In recent years, a number of word processing programs have acquired so-called mathematical ability. For example, MicroSoft Word even has an advert for Word 5 with some equations in it: they are acceptable, but not really of the highest quality: they are not even of the quality of  $\text{eqn}$ . Either quality is not an issue, or mathematics is such a strange pursuit that no-one recognises when it is done badly.

I have a problem with 'highest quality', as is probably evident. I expect  $\text{\TeX}$  or whatever to be pretty good. I do not expect it to be perfect. Like a Persian rug, it ought to have at least one mistake in it. The fear of hubris is just too great. Even the concept that perfection could be achieved by a program worries me. I expect, indeed I am duty bound, to get in there and meddle. Obviously there are levels and magnitudes of meddling.

But there is an interesting question: why would anyone re-invent the mathematics typesetting wheel? or why would you not incorporate  $\text{\TeX}$  mathematical typesetting in Word, or WordPerfect, or Ventura, or Frame, or Interleaf? Can anyone explain this? Sometimes we find  $\text{eqn}$  in there instead: sad. Having brought up  $\text{eqn}$ , we have to point out the presence of a computerised typesetting tool which seems to keep running, without moans and groans about its total inadequacy to face the future— $\text{troff}$ : it just goes on as every UNIX system rolls off the production line. It doesn't aspire to excellence, it just comes as part of every system, and all the manuals expect its availability—for goodness sake, it isn't even device independent (well, it is now, but that took for ever to achieve— $\text{ditroff}$  produces  $\text{dvi}$ !). It is surprising to see the longevity of the  $\text{nroff}$ / $\text{troff}$  tools. They seldom produce anything very exciting, and they make no pretension towards quality. They seem to meet a very real need and in a very straightforward way, although I was sur-

prised to see a book produced recently which had as its topic  $\text{tbl}$  [12]. Maybe it's a subject area a whole lot more difficult than it seems.

## 7 Time

Let's briefly consider time spans. It isn't easy to work out just how much effort went into  $\text{\TeX}$ . Somewhere, Knuth records that in 1977 he announced to Jill that he was going to take a year off his academic work to write a typesetting system. In fact we actually know when he started working predominantly on  $\text{\TeX}$  (Thursday May 5th, 1977) (see [7] and [8]). Even more bizarre, we know what films he went to see that weekend (*Airport 77* and *Earthquake*). In the midst of this trivia, we have the estimate from Knuth, arguably one of the most talented programmers to have existed, that the program would take one year (or perhaps less) to complete. More realistically it appears to have taken at least four or five years in elapsed time (this is a wild guess: improved estimates would be appreciated): from this we might have to subtract the time spent on  $\text{METAFONT}$  and Computer Modern (and  $\text{WEB}$ ), but on the other hand we should add in the efforts of his graduate students and all the others (like Art Samuel, David Fuchs, Luis Trabb Pardo, Frank Liang, Michael Plass, Arthur Keller...) who contributed to the program. I suspect that four or five man years is still a conservative estimate. Four or five man years of a small, highly motivated team, with one person in control who could decide what and what not to include.

This was not a democratic process, although it is clear that there was feedback. Even more recently, the transition to  $\text{\TeX}3$  seems, to me, to have taken a shade longer than anticipated. There are probably many reasons for this. After all, Knuth was not really planning to change  $\text{\TeX}$  in 1989. Forces conspired against him there, and marshalled some convincing arguments, and it is evident that he already had the feeling that seven-bit character representations were inadequate. The point here is that Knuth, with his intimate knowledge of the program, still appears to have taken longer than he expected to complete the changes.

One of the things that we have surely learned over the last fifty or so years of programming is that it takes longer than you expect. The folk-lore of computing (backed by some extremely readable books like Brook's *Mythical Man Month* [2]) knows that a project will take at least twice as long as you estimate; that doubling the estimated time has no effect on this inflation factor; and that the program will always be finished 'in another four weeks'. Changes to  $\text{\TeX}$ , or a re-write, are going to take a

- the designer of a new kind of system must participate fully in the implementation
- writing software is much harder than writing books
- the designer should also write the first user manual

**Figure 1:** Knuth's lessons

long time. It will be a pity to have any new development labeled vapourware, but there will necessarily be a long time spent in development. It is unlikely that we will find some wealthy benefactor who will turn round and say 'take this million dollars: take your time: improve T<sub>E</sub>X'.

Knuth [8] himself says 'If I had time to spend another ten years developing a system with the same ideas as T<sub>E</sub>X — if I were to start all over again from scratch, without any considerations of compatibility with existing systems — I could no doubt come up with something that is marginally better.' My point here is the word *marginally*.

## 8 Or money

Because of T<sub>E</sub>X's public domain status, we sometimes lose sight of the fact that it did cost money to develop. Knuth [8] records 'generous financial backing' from a number of sources, including the System Development Foundation, the US National Science Foundation, and the Office of Naval Research. How much money is indeterminate, since it is unlikely that any of the funding detailed 'work on T<sub>E</sub>X'.

Any future work will have to be done by interested individuals, probably working in their spare time, or, if we are exceptionally lucky, by graduate students working together on a funded project, although note Knuth's 'lessons' from the T<sub>E</sub>X project ([8], Figure 1). I am not clear I see who to approach for the funds. Inter-disciplinary research has not been too well funded (certainly in the UK) in recessionary times. The core areas let the peripheral stuff go in times of crisis.

Where does computerised typesetting fit? Is it computer science; is it a branch of engineering; is it part of some typographic or fine arts discipline? Let's hope it isn't the latter, since they are particularly badly funded. But is this really research in the commonly accepted sense? What will we end up with? Something which is in some sense better than an existing program. How are we going to sell this? How will we convince some body with loose cash to support this? Do we indicate just how dreadful T<sub>E</sub>X is, exposing all its warts and deficiencies? Why are we using it in the first place if it is so bad? Would a cheaper and easier solution not just

be to use an existing program which has none of these deficiencies? Never mind that there is no such paragon. The other contenders must offer some improved or needed features or they would not be in use at all. The chances are this proposal will have to go through a committee. If those on the committee have ever prepared their own documents (and remember there are still some oldsters out there who do not; their secretary does it), they will have their own favourite software. So we will end up telling a reasonably influential (maybe) bunch that T<sub>E</sub>X is deficient and needs changing. In the end we are asking them to invest a fair chunk of money in order to benefit whom? This is one I find difficult.

### 8.1 Cui bono?

The people who seem most likely to benefit are book publishers: correct me if I am wrong here. But it appears to me that the principal beneficiaries are organisations like Elsevier, Springer Verlag, Addison Wesley and so on.

Oh dear. I confess that I would anticipate that printing and publishing organisations might reasonably be expected to underwrite research into the development of quality typesetting. There are research organisations founded and financed (at least in part) by them. In the UK, PIRA (Printing Industries Research Association) does just that, although in recent years it has become much more commercially oriented. There are others in other countries.

A ray of hope might be seen in some projects funded through initiatives which ultimately derive from Brussels and the EC. The Didot project is/was a three-year project set up to re-establish European pre-eminence in typography (in the sense of type design), and, from the outset, had a very strong digital component. It seems to have been successful in bringing type practitioners and computing people together (and maybe even a few engineers). The outcome of the project is to develop training programs, and an increased awareness and facility with digital type design. The project should finish in 1993. It does not quite do what we want, but it indicates that there are precedents. Although Didot started out with a rather strong chauvinist element (basically to prevent Europe being overwhelmed by the US, always a populist rallying call in Europe), it mellowed quite considerably and there is apparently effective interaction with North America now. But it remains a suspicion in my mind that an appeal to some external threat could be the most effective, if least ethical, way of appealing for funds.

## 9 I'll be in Scotland afore ye

I see two main routes towards a descendant of  $\text{\TeX}$ . One is an evolutionary approach, where the perceived deficiencies are remedied, and a few new features are added. Basically,  $\text{\TeX}$  itself changes only slightly, and in a well-defined way. Vulis'  $\text{V}\text{\TeX}$  [21] can be seen as an example, where the handling of fonts has been substantially changed, and arguably enhanced. Similarly, Ferguson's  $\text{ML}\text{\TeX}$  [4] which allowed multilingual hyphenation,<sup>3</sup> falls into this category. It might even be reasonable to place Harrison's  $\text{VOR}\text{\TeX}$  project [5] into this model. I am quite a fan of the project, partly because I feel that the model they developed, of multiple views of documents, has much to commend it. The fact that the program itself was rather machine specific is a side issue. Almost five years or so ago, it accomplished at least some of the things that we presently feel we need.

There is probably not a single route, but several. If people go ahead and add some features to the underlying code, is there any guarantee that the full range of features added will be compatible with one another? I can envisage a whole cluster of similar but incompatible descendants. With luck an existing  $\text{\TeX}$ -encoded file will produce identical output, but there may be no way to use the extended features of more than one. Perhaps one will out-evolve the rest. There are examples of this happening. Tom Rokicki's  $\text{DVIPS}$  is arguably the *de facto* PostScript driver. This was not always so. There are, or have been, at least eight PostScript drivers, but Tom's has the advantage of being versatile, up-to-date, and runs on most platforms. It is also in the public domain.

If this is one route, what is the other? Why, a radical restructuring. Throw away the baby, bathtub and water, but keep the mission—that of creating a device for typesetting of the highest quality. I confess I find this a somewhat vague statement at best. How will the model be chosen? Who will be involved? In the worst possible case it may be totally democratic, and we can look forward to interminable referenda on desirable features. Let me quote from Knuth [8]:

I was constantly bombarded by ideas for extensions, and I was constantly turning a deaf ear to everything that did not fit well with  $\text{\TeX}$  as I conceived it at the time . . .

I was perhaps able to save  $\text{\TeX}$  from the 'creeping featurism' that destroys systems whose users are allowed to introduce a patchwork of loosely connected ideas.

- an altered  $\text{\TeX}$  is not ' $\text{\TeX}$ '
- will descendants be accepted widely?
- will they be public domain?
- who authorises or legitimises?
- will there be a trip test?
- may be multiple, mutually incompatible, descendants
- will they be widely ported?
- begins a tradition and expectation
- what time scales?

Figure 2: Some fears for a future development(s)

Apart from a warm and fuzzy glow, I am not too clear what I or any other existing  $\text{\TeX}$  or  $\text{\LaTeX}$  user will get out of either route, apart from more upgrades. I feel I may even be tempted to do nothing, and just hang onto my working and apparently almost perfectly satisfactory current version of  $\text{\TeX}$ . For remember this: you will not be able to call this new beast ' $\text{\TeX}$ '. This alone seems to me to mean that any small enhancements are likely to be still-born. It will be viewed with suspicion. It is  $\text{\TeX}$ , but it isn't ' $\text{\TeX}$ '. Perhaps the highly  $\text{\TeX}$  literate will understand the differences, but the great unwashed will have to be sold the idea. How do you sell ideas when you are not commercial? and not very fashionable? Some of my fears are summarised in Figure 2.

I do not want to appear gloomy and despondent. I do not feel that way at all. I know that  $\text{\TeX}$  is not perfect. I can see several minor blemishes (and at least one major one). I would prefer the program to be truly modular, although that confers no immediate benefit. But I am not altogether convinced that the next generation will please me any more. What pleases me most about  $\text{\TeX}$  is its solidity. It has not changed much in the last eight or so years. And I do not feel too dissatisfied, although I think I have been using it seriously. Maybe I do not use it to its limits, but that is largely because its limits are pretty wide and the little I have learned about software indicates that when you push it to its limits, it breaks. That is not to say that developments will not take place, but like many others, I see them around the periphery (Figure 3).

This conclusion is awesome: in my self-view I like to feel I am some sort of radical, an iconoclast (in spite of my love of the Macintosh and its icons), and here I am saying do not change the core. This is so embarrassing. But equally it indicates that maybe it's a valid view. I may now go on and

<sup>3</sup> Now, of course, superseded by  $\text{\TeX}3$

- improve the support environment
  - editors
  - drivers
  - overall integration level
- widen the scope
  - additional macros/styles
  - dvi processors for increased functionality

**Figure 3:** Already suggested alternatives for development

show how many angels may stand on the head of a pin.<sup>4</sup>

### A An editorial paraphrase

Lammarsch's editorial comments [9] in the German-speaking group's 'Die T<sub>E</sub>Xnische Komödie' were published in August of 1992. They throw some useful light on what has been done, although the details are perhaps still unknown to those who do not read the Komödie. Paraphrased and translated (for which translation I am grateful to Peter Schmitt), Lammarsch stated the following

- Knuth is positive with regard to the project;
- funds, amounting to 20% of that required, have been secured already; in an earlier report, Lammarsch estimated that the project would cost DM 500 000, over 5 years;
- 'big publishers' have promised to support the project;
- commercial T<sub>E</sub>X dealers have accepted the project;
- the program will remain 'freeware'.

Like many others, I look forward to details of Knuth's endorsement, the extent of publishers' support, and the progress of the project. It is to be hoped they will be circulated widely.

### References

- [1] Stephan von Bechtolsheim, 1990, T<sub>E</sub>X in practice: comments on a 4-volume, 1400-page series on T<sub>E</sub>X, *TUGboat* 11(3), pp. 409–412.
- [2] Frank P. Brooks, 1974, *The Mythical Man Month*, Addison-Wesley.
- [3] Victor Eijkhout, 1992, T<sub>E</sub>X by Topic, Addison-Wesley, 307 pp.
- [4] Michael Ferguson, 1985, A multilingual T<sub>E</sub>X, *TUGboat* 6(2), pp. 57–58.
- [5] Michael Harrison, 1989, News from the VORTEX project, *TUGboat* 10(1), pp. 11–14.
- [6] Alan Hoenig, 1990, Line-oriented layout with T<sub>E</sub>X, in T<sub>E</sub>X, Applications, Uses, Methods (editor, Malcolm Clark), Ellis Horwood, Chichester, pp. 159–183.
- [7] Donald E. Knuth, 1989, Remarks to celebrate the publication of *Computers & Typesetting*, *TUGboat* 7(2), pp. 95–98.
- [8] Donald E. Knuth, 1989, The errors of T<sub>E</sub>X, *Software practice and experience* 19(7), pp. 607–685.
- [9] Joachim Lammarsch, 1992, Grußwort, *Die T<sub>E</sub>Xnische Komödie* 4(2), pp. 4–5.
- [10] Joachim Lammarsch, 1993, A new typesetting system: is it really necessary? *TUGboat* 14(3), pp. 167–170.
- [11] Leslie Lamport, 1987, T<sub>E</sub>X output for the future, *TUGboat* 8(1), p. 12.
- [12] Henry McGilton and Mary McNabb, 1991, *Typesetting tables on the UNIX system*, Addison-Wesley, 280pp.
- [13] Frank Mittelbach, 1990, E-T<sub>E</sub>X: guidelines for future T<sub>E</sub>X, *TUGboat* 11(3), pp. 337–245.
- [14] Lynne Price, 1981, Dreamboat, *TUGboat* 2(2), p. 58.
- [15] David Salomon, 1991, personal communication to TUG Board.
- [16] Norbert Schwarz, 1988, *Einführung in T<sub>E</sub>X*, Addison-Wesley.
- [17] Norbert Schwarz, 1989, *Introduction to T<sub>E</sub>X*, Addison-Wesley, 278pp.
- [18] Michael Spivak, 1991, A contrarian view on T<sub>E</sub>X extensions, *T<sub>E</sub>Xline* 13, pp. 1–3.
- [19] Philip Taylor, 1992, The future of T<sub>E</sub>X, in *EuroT<sub>E</sub>X'92, Proceedings of the 7th European T<sub>E</sub>X Conference, Prague* (editor Jiří Zlataška), pp. 235–254; reprinted in *TUGboat* 13(4), pp. 433–442.
- [20] Philip Taylor, 1993, NTS: the future of T<sub>E</sub>X? *TUGboat* 14(3), pp. 183–186.
- [21] Michael Vulis, 1990, V<sub>T</sub>E<sub>X</sub> enhancements to the T<sub>E</sub>X language, *TUGboat* 11(3), pp. 429–434.

◊ Malcolm Clark  
 Computing Services  
 University of Warwick  
 Coventry CV4 7AL, England, UK

<sup>4</sup> As many as want to.