Two Sides of the Fence[1]

Nico Poppelier

Elsevier Science Publishers
Academic Publishing Division, R&D Department
Sara Burgerhartstraat 25
1055 KV Amsterdam
the Netherlands
n.poppelier@elsevier.nl

Abstract

The purpose of this talk is to give an overview of the four days of the twelfth annual TUG meeting; it is an attempt to show that the different streams in the programme of the meeting are connected, that they are part of a whole.
Also, I make some comments and observations regarding the current status and the future of TEX, and the future of publishing in general.

1 Introduction

In his book *Zen Buddhism* [5], Christmas Humphreys writes:

> How then, does it work, this faculty of the mind [the intellect] which men so highly prize and far too lightly claim to be infallible? The answer is, by the interaction of the opposites.

The purpose of this talk is to give an overview of the four days of this conference, and I will use pairs of opposites to guide me through it.

If you talk about pairs of opposites, you also talk, implicitly, about a fence, a boundary between the two opposites. And if you consider any of these fences you can ask yourself: do we make an opening in the fence, i.e. make a pragmatic decision in order to bridge the gap, to integrate seemingly irreconcilable views? Or will we remain passive, will we stay 'sitting on the fence', i.e. not decide anything? There is of course a third possibility, namely that the fence is there for a real purpose.

I hope that this conference will result in gates through the various fences I will discuss.

2 Dichotomies

The first pair of opposites came into my mind very quickly: the TEX-using author vs. the TEX-accepting publisher. From the TEX files we've received so far at Elsevier Science Publishers I've gotten the impression that the average TEX-using author wants as much freedom as possible to typeset the text, the tables, the math and the figures. He/she wants to use TEX in any possible imaginable way and, according to TEX experts at a few physics institutes, spends sometimes up to 50% of the total time for the article or book on its presentation.

Suppose he has to deal with publisher X, who has a TEX macro package plus instructions to authors. Then maybe the author isn't very happy with it, since it limits him in his creativity and furthermore, since he has to deal with many publishers, he has to figure out a way of dealing with these different macro packages and instructions. A very likely solution is that he just ignores them all!

The publisher who accepts TEX has a slightly different point of view. Of course, on the one hand, a publisher wants to be as friendly as possible to an author and accept his compuscript. But, on the other hand, a publisher wants to convert the TEX compuscript into a printed book or journal paper in the shortest time possible with a minimal amount of effort.

[1] Keynote presentation at TUG91 meeting in Dedham, USA; to be published in TUGboat,

There are several constraints to be met in this publication process: the house-style for the particular journal or book series, the quality of the publication (language, layout), the time it takes to publish the article or book, and the cost of all this. Most publishers are commercial firms, not philanthropic institutions, so cost efficiency is an important criterion. In most cases, the publisher would really like to see authors following the instructions.

How do you solve this dilemma? A compromise might be to agree upon a certain standard or set of standards between various publishers. In our company, we think that we will not be able to handle TEX compuscripts efficiently if we accept all varieties of TEX, especially because the material ranges from very simple to very complex with lots of math and tables. Efficiency is particularly important for journals, where you have a steady flow of material, a fixed house-style and a routine way of working.

Our choice is: one variety of TEX, namely LATEX. For book and proceedings projects this preference is somewhat less strong, although making a book ready for publication, in a house-style or in the style of a particular book series, complete with a table of contents and an index, is easier if the book was prepared with LATEX—and the author has used LATEX well!—than if it was prepared with plain TEX.

Besides the problems just mentioned, there are several other matters you have to solve anyway, regardless of whether you use plain TEX, LATEX or, say, ΦΥΣTEX:

- complex tables
- page layout
- font selection (other fonts than Computer Modern)
- illustrations in PostScript or other format

So now I've come to my second pair of opposites, one that will be addressed by several speakers this week: TEX versus LATEX.

The key concept of LATEX is, as you of course know, the concept of logical design: an author writes his text in terms of abstract building blocks, in terms of the logical structure of the text. Content and layout are decoupled as much as possible. The visual structure is derived from the logical structure, and is specified in the document style.

As I said earlier, some authors appear to spend large amounts of time on the presentation of a paper that is submitted for publication in a journal: they write sets of macros ranging in size from one screen to many hundreds of lines, use any font they can find in all sorts of combinations, etcetera. This strikes me as odd for two reasons: (i) an author's main concern should be the *contents* of the article or book, and (ii) the presentation the author chooses will almost always be changed by the publisher anyway, whether he submits the material on paper, on a diskette or via electronic mail.

We have found that the LATEX-way-of-working is fine for both journals and books: document styles have been written for about ten journals and several books. The difference between conventionally typeset material and material produced from author-prepared LATEX files can only be seen by a well-trained eye.

Now of course, there is much more to this type of electronic publishing than just changing the document style: a technical editor has to look at spelling, punctuation, language in general, notation, the appearance of mathematical formulas in text and in displays, the layout of tables, the page layout, spacing, hyphenation, ... a lot of work, often difficult work. The combination of usual copy-editing with TEX requires skilled technical editors and a certain routine way of handling TEX.

But TEX is not the only document preparation publishers have to deal with. And so now I come to my next pair of opposites: TEX vs. non-TEX, or TEX versus the rest of the desktop-publishing world.

If we asked scientists who publish in one of our more than 600 journals whether they use a computer to write their articles and if so, what word processor they use, we would find enormous variety in their answers. In physics and mathematics, TEX is used by the majority of authors, but even there you find a significant number of authors who use `troff/eqn`, ChiWriter, Word or various Macintosh word processing programs.

In other scientific disciplines, TEX is used by only a few people—if at all! What I personally find most interesting is the many ways TEX is used, not by mathematicians and physicists, but by people working in, say, linguistics, humanities. My next pair of opposites.

Often there is no alternative but TEX for producing texts in languages that use non-Latin alphabets or the Latin alphabet with diacritical marks. With TEX you can produce remarkable, often beautiful results, after you have solved dozens of problems that others, who use TEX for texts written in English, with a lot of math and tables, have never thought of. I am fascinated by the work on

- hyphenation of other languages than English
- right-to-left text with TEX: Hebrew and Arabic
- diacritical marks and other embellishments: Hebrew, Vietnamese
- wonderful fonts: Greek, Hebrew, Arabic, Old German, Ethiopic, Korean *hangul*, Japanese *kana*, Chinese *kanji* or *hanji*, and the many languages of the Indian sub-continent
- vertical typesetting: Japanese and Chinese

and I hope to see a lot of these types of TEX applications during this conference. I think that, in principle, TEX has great potential as a text composition system for authors in *all* scientific disciplines and in *all* languages. But, I said 'in principle'—I will come back to that later.

Coming back to the observation that TEX is not the only software: when a publisher sees that he also re-

ceives papers prepared in Word and ChiWriter, what does he do with them? Does he handle them in the old-fashioned way, that is re-type the whole thing and introduce lots of typos, so that the author has to read the stuff for the umpteenth time? Or should the publisher convert it to one of the professional typesetting systems he uses? Or convert it to TEX, since there are several of these conversions available: WordPerfect to TEX, ChiWriter to TEX, ...

I think conversion will become important or is already becoming more and more important. Conversion of information from one format into another, from an author's word processor X to a publisher's typesetting system Y. Now suppose authors use M different word processors and that publishers uses N different typesetting systems: does this mean we have to wait for the development of $M \cdot N$ different conversions? This does not appear to be a feasible solution. Conversion, or translation, via an intermediate language, a standard exchange language for text, would require only $M + N$ different conversions, much less!

As most of you know, such an intermediate language already exists: SGML [6, 2, 4]. Aha, you might think: the fourth pair of opposites. Well, yes and no. Yes, in the sense that many people think that TEX and SGML are two alternatives for one and the same purpose. No, in the sense that I do not agree with this: I do not believe that SGML and TEX form a pair of opposites and I would like to explain why I think this is the case.

SGML is not a typesetting language, but an abstract language, or more precise: a meta-language. Just as you can define the computer programming languages Pascal and Modula-2 in BNF (Backus-Naur form), another example of a meta-language, you can define typesetting languages in SGML.

In SGML, there exists something that is called the *document type definition*. A document type definition (DTD) is a description of a class of documents. You describe a document instance, a document that is representative for a certain class of documents, say *book*, as a hierarchy of building blocks. To give an example:

```
book    = front_matter body back_matter
body    = chapter+
chapter = chapter_heading, paragraph?,
          section*
...
```

all the way down to the basic building blocks: paragraphs of text, mathematical formulas, ... This defines the contents of the book in terms of logical entities: you might call it 'object-oriented writing of a document'.

An alternative is to describe the visual structure of a document, which can also be regarded as a hierarchy of building blocks.

```
book       = pages+
page       = header_block text_block
             footer_block
text-block = ...
...
```

These are sketches of two DTDs. A DTD defines a set of tags, you could say typesetting instructions, and their hierarchy. The set of typesetting instructions is in fact a typesetting language. So in fact I've just given two typesetting languages. You could also define the syntax of a language like TEX in SGML. Mostly however, document type definitions are written with the logical structure of a class of documents in mind.

By the way: two parallel views of one piece of text—view 1: logical structure, view 2: visual structure—can be important or even essential in pre-existing text, something that is pointed out in the draft report of the Text Encoding Initiative [10], on which Michael Sperberg-McQueen will speak [9]. For example: inscriptions found on historical sites or texts in *real* manuscripts—you know: hand-written books.

At present however, publishers do not receive a great quantity of SGML-coded material—not yet! There are not many SGML editors available and the ones that are available are not or hardly ever used by the authors one finds in normal textbook or journal publishing. Furthermore, the word processors these authors use do not have an SGML export facility. So if a publisher wants to have material available in some form of SGML, it means converting it from whatever form the material is in when he receives it—at least for many years to come.

Encoding a piece of text with SGML means

- separating form from content, presentation from function
- adding structure to a text, enriching the text

In particular, the last activity is a time-consuming one, both for the author and the publisher, but it significantly increases the potential usefulness of the information. If a text is fully tagged, as it is called in SGML, if pieces of text are identified by their function, all sorts of information can be extracted, stored and re-used. For example: the article opening and the lists of literature references. If you use the text as part of a hypertext, links to figures, tables, references, footnotes and other parts of the text can be derived *automatically*.

But I would like to stress that SGML has nothing to do with getting a piece of text on paper or on screen. For that, you always need a separate program. So, 'SGML or TEX' is not a question at all, since you can't compare SGML and TEX. Valid questions to be asked are:

- do you combine SGML and TEX, SGML and Ventura, or SGML and you-name-it?
- *how* do you combine, let's say, SGML and TEX?

Suppose you use TEX as a back-end to a document-preparation system based upon SGML. What sort of problems do you encounter then? If you make a list of these problems and add ideas from various other TEX experts, you get a very long wish list indeed. What extensions do we need to add to TEX? *Are* we going to

change TEX or are we going to build a completely new program?

3 Future of TEX

I'd like to spend a few minutes of my talk on this subject, since I'm not really happy with the current status of TEX. If you think the following is a bit provocative, well . . . , maybe it's intended that way.

To put it simply: I think the program should never have been frozen. Its author should either have continued developing TEX or handed over this work to a new implementor, or preferably a group of implementors. If this happens with professional—or, if you like, commercial—software, if you do not listen to the users of your program, or if you freeze a program, the software will be as good as obsolete after a few years.

I will not try to improve upon Frank Mittelbach's excellent paper 'E-TEX: Guidelines for Future TEX Extensions' [8], which he presented at last year's meeting in Texas. Rather, I will add a few of my own comments, or observations.

A big deficiency in TEX is the page-breaking algorithm and the tools TEX offers to program complicated page layouts, for example two-column or three-column with footnotes and floating bodies of 1 or more columns. If you use TEX as it now is as the back-end to an SGML-based system, page layout cannot be achieved fully automatically: manual work is still required. And even though TEX is intended to be used by a typist, not as a fully automatic back-end system, the more work the computer does without human intervention, the better. This makes the SGML–TEX combination far from ideal.

The same problem occurs if you use LATEX, which has a pretty complex output routine for scientific journals with a two-column layout, with lots of figures, tables and footnotes.

TEX users who have tried it know how difficult it is to let TEX typeset text—let's assume ordinary left-to-right text—in a language with lots of accented letters, ligatures and complicated hyphenation. Why are there no under-accents, multiple accents? Why is hyphenation of accented words or compound words with hypehns such a problem? I will use a few technical phrases from my own background, nuclear physics, as examples to show that the problem of hyphenating compound words, for example, is not just a problem of, say, the German or Dutch language.

Compound words are quite frequent in Dutch, for example:

`schillenmodel-berekening`

(shell-model calculation). Most TEX users would like to see TEX hyphenate this as 'schil-len-model-bere-kening', which TEX of course doesn't do.

But compound words of this type also occur in English:

`formation of a compound nucleus`

is hyphenated by TEX as 'for-ma-tion of a com-pound nu-cleus', whereas

`compound-nucleus formation`

is hyphenated by TEX as 'compound-nucleus for-ma-tion', instead of 'com-pound-nu-cleus for-ma-tion'.

There should have been a switch for this in TEX, but there isn't! Why wasn't the functionality of TEX–XET and everything else I've mentioned added to TEX 3?

It is my opinion that TEX would have been a better program if its creator had agreed to re-think certain choices he had made years ago, especially when users argued their case by showing what sorts of problems TEX poses, as was done by several of them in articles in *TUGboat*. Barbara Beeton explained to me some time ago that the decisions regarding TEX's accent mechanism—`\accent` or ligature, single or multiple accents, only above or also below and to the side?—were Don Knuth's decisions and his only; they were not based on discussions with other experts, which I think is unfortunate. I sometimes think—and this is not intended as a bad joke!—that certain parts of TEX would have looked different if Knuth had been German or Greek, because English is such an easy language to typeset, relative speaking!

And while TEX is superior in mathematical typesetting, there is still a lot to criticize in that area as well. An example is the spacing between the eight basic types of math atoms, which is hardwired into the program as a sort of matrix, instead of being accesible via parameters. This results in a lot of handwork if a particular house style deviates from TEX's rules. Again, I would like to refer to Frank Mittelbach's article and the work on $\mathcal{AMS}$-TEX by Michael Spivak.

Another example: where's the missing lowercase Greek?

	upright form	slanted form
lowercase letter	?	π
uppercase letter	Π	$\mathit{\Pi}$

In other words: why was it arbitrarily decided that there was no need for upright Greek lowercase letters in the Computer Modern fonts?

A lot of work still needs to be done. Whoever is going to do it, I think that the successor to TEX 3—the matter of the name, TEX 4 or E-TEX 1 or God-knows-what, is unimportant, the important thing is that there should be *one* successor, not several incompatible systems based on or derived from TEX—should not be developed and maintained by

- one single person
- one or more persons all working in one field of work, for example mathematics or physics

  - otherwise the successor to the font set that is now more or less standard, Computer Modern plus $\mathcal{AMS}$-Fonts, will contain exotic symbols such as $\precapprox$ and $\rightsquigarrow$, but not basic ones like the male and female symbols
- one or more persons all speaking the English language

During this conference there will be a panel 'The future of TEX'. An important subject, something the TUG board, TUG members and TEX users in general should think about a lot. As I said earlier: in principle, TEX has great potential for authors in *all* scientific disciplines and *all* languages, but only if the program is developed further.

4 Future of Publishing

The last topic I would like to talk about is the future of publishing. I don't think I am the right person to make prophecies concerning the future of publishing. Instead, I would like to present some ideas I have found in recent science fiction stories and novels.

One of the most striking ideas I've come across in the past couple of years is the idea of direct brain-computer coupling, as used by the Canadian author William Gibson, who is called the founder of the sub-genre 'cyberpunk', in his *Neuromancer* novels. With the direct brain-computer coupling, you can access any collection of data and it is as if you navigate with a virtual body through the space of data, which Gibson called 'cyberspace'. It is not such a weird idea at all, although an idea of the far future, and it is related to what people call 'virtual reality', a very popular phrase in some circles nowadays.

An idea that might become reality in the near future can be found in a book by the American science fiction writer, David Brin, in his latest novel 'Earth' [1]:

> If only it were a modern document, with a smart index and hyper links stretching all the way to the world data net. It was terribly frustrating having to flip back and forth between the pages and crude flat illustrations that never even moved. Nor were there animated arrows or zoom-ins. It completely lacked a tap for sound . . . in a normal text you'd only have to touch an unfamiliar word and the definition would pop up just below. Not here though. The paper simply lay there, inert and uncooperative.

To leave fiction and come back to the here-and-now: according to the Faxon Planning Report 1992 [3], Faxon Press'[1] poll of 52 periodical publishers, half of them commercial publishers, the other half non-profit organizations, a small majority of these publishers were quite worried about the future of publishing as we know it. Almost all of them still believe in the primacy of printed books and journals for decades to come. Is the vision David Brin presents something of the very far or of the very near future?

Just a few points to think about:

1. There are still librarians and scientists who see nothing whatsoever in electronic journals and books.
2. But the amount of information printed on paper increases exponentially.
3. And finding the right information becomes increasingly difficult.
4. Furthermore, increase of paper usage is also a serious environmental problem.

Well, you can't halt progress: electronic books are here already and their number will grow. In the transition period there is still another problem. An electronic book has to be available in paper form as well, since most readers still prefer a paper book.

Suppose you use TEX for the paper version, what do you use for the electronic version? How do you handle the two presentation styles? This is something I hope John Lavagnino will address in his talk on simultaneous electronic and paper publication of Thomas Middleton's complete works.

Is DSSSL[2] the answer to these problems, or FOSI[3]? What will the role of TEX be in non-paper publishing? I really don't know, but we should all think about it.

TEX is superior compared to desktop-publishing programs. It can handle mathematical formulas and complex tables, and this is a capability that is often lacking or poorly developed in desktop-publishing programs. Existing programs for the creation of electronic books also lack these capabilities: they can handle only text and graphics. If you want to include mathematical formulas or tables, the most sophisticated you can do is prepare bitmaps of these components—by means of scanning, or perhaps TEX?—and put these in the electronic document in the form of graphical objects.

5 Conclusion

This conference offers a great opportunity for discussions between TEX users and commercial professionals, since the programme contains a lot of talks about many different current applications. There are interesting panel discussions and hopefully there will be plenty of

[1] A large, completely automated subscription agent in the United States, involved in many activities.

[2] An ISO standard under development for the specification of document processing, such as formatting and data management [7]. The acronym stands for 'Document Style Semantics and Specification Language'.

[3] See the paper by Andrew Dobrowolski in these proceedings.

time for discussions during the breaks and in the evenings.

One of the goals of this conference is to try and bridge the gap—apparent or real—between the two poles of my first dichotomy: the author who is a TEX user, and publishers or other commercial professionals who want to accept TEX material. Looking at and thinking about present applications of TEX, as well as an historical perspective, can help to bridge this gap.

This conference is also a good opportunity to discuss the future of TEX, the future of publishing and the future of TEX-in-publishing. And I hope that it will be a success in all respects: that we will be able to find solutions to the problems I mentioned and those that will be described in the next four days—that we will be able to make gates in the fences and not just sit on the fences.

I'd like to thank the organization for inviting me to give this introductory talk. It was a pleasure to prepare and give this talk, and I feel honoured having been invited here.

References

[1] Brin, David. *Earth*. New York: Bantam Spectra, 1991. [Quoted with permission from the author.]

[2] Bryan, Martin. *SGML: An Author's Guide to the Standard Generalized Markup Language*. Workingham, UK: Addison Wesley, 1988.

[3] *The Faxon Planning Report 1992*. Boston: The Faxon Press, Boston, 1991.

[4] Herwijnen, Eric van. *Practical SGML*. Dordrecht: Kluwer Academic, 1990.

[5] Humphreys, Christmas. *Zen Buddhism*. London: Unwin, 1984.

[6] International Standard ISO 8879: Standard Generalized Markup Language (SGML). Geneva, 1986.

[7] International Standard ISO 10179: Document Style Semantics and Specification Language (DSSSL). Geneva, 1991.

[8] Mittelbach, Frank. "E-TEX: Guidelines for Future TEX Extensions." *TUGboat* 11(2), pages 337–345, 1990.

[9] Sperberg-McQueen, C.M. "Specifying Document Structure: Differences in LATEX and TEI Markup." See elsewhere in these proceedings.

[10] Sperberg-McQueen, C.M., and Lou Burnard, eds. *Guidelines for the Encoding and Interchange of Machine-Readable Texts*. Chicago and Oxford: Text Encoding Initiative, November 1990. [Available upon request from the authors.]