

I'd also like to say a few words about the history of my own work on computers and typesetting. Last week I went back to my diary of 1977 and found an entry from Thursday, May 5, where it says 'Design of T_EX started'. My diary says that I worked intensely on the design all day Thursday, Friday and Saturday; then I went to see *Airport 77* and *Earthquake* to relax! The entry for the following Thursday says: 'Wrote draft report on T_EX, stayed up till 5 a.m. typing it into machine'. That weekend I went with my wife on a tour of the Sacramento area with Stanford's Library Associates. We saw many examples of fine printing during that trip, and this encouraged me to read a lot of books about font design during the following week. My diary entry for Saturday, May 21, 1977—exactly nine years ago today—says that by 5 a.m. that day I had made 'rough drafts of lower case and upper case Roman and italics and digits 0-9'. After a few hours of sleep, I spent the rest of that Saturday writing computer programs to plot curves on a raster. Oh, how little I knew in those days about how difficult it would be to complete this work, which I had sketched out in about two weeks!

Why did I *start* working on T_EX in 1977? The whole thing actually began long before, in connection with my books on *The Art of Computer Programming*. I had prepared a second edition of volume 2, but when I received galley proofs they looked awful—because printing technology had changed drastically since the first edition had been published. The books were now done with phototypesetting, instead of hot lead Monotype machines; and (alas!) they were being done with the help of computers instead of by hand. The result was poor spacing, especially in the math, and the fonts of type were terrible by comparison with the originals. I was quite discouraged by this, and didn't know what to do. Addison-Wesley offered to reset everything by the old Monotype method, but I knew that the old way was dying out fast; surely by the time I had finished Volume 4 the same problem would arise again, and I didn't want to write a book that would come out looking like the recent galleys I had seen.

Then a nice thing happened. I was on a committee to revise Stanford's reading list for our department's comprehensive exam, and one of the things we had to do was evaluate a book that Pat Winston had just written about *Artificial Intelligence*. We received galley proofs of that book, and the story we were told was that these galleys had been made on a new machine in Southern California, all based on a discrete high-resolution

raster. Apparently one of Winston's students at MIT had flown to Los Angeles with that book on magnetic tape, and the galley proofs we saw were the result. Well, I had had lots of experience with rasterized printing, but only at low resolution, so I thought of it as simply an amusing toy. When I saw these galleys of Winston's book, I was astounded, because the resolution was so good I couldn't tell that the type was actually digital. In fact the digital type looked a lot better than what I had been getting in my own galley proofs.

Digital typesetting means patterns of 0s and 1s, and computer science can be thought of as the study of patterns of 0s and 1s. Therefore, it dawned on me for the first time that I, as a computer scientist, would be able to help solve the printing problem that was worrying me so much. I didn't need to know about metallurgy or optics or chemistry or anything scary like that; all I had to do was construct the right pattern of 0s and 1s and send it to a high-resolution digital typesetter like that machine in Southern California; then I'd have my books the way I wanted them. In other words, the problem of quality printing had been reduced to a problem about 0s and 1s. Therefore it was almost an *obligation* for a computer scientist like myself to study the problem carefully.

Within a week after seeing the galleys of Winston's book, I decided to drop everything else and to work on digital typography. Although Winston unfortunately couldn't be present here today—Pat, I can't thank you enough for having written that book!

Ever since these beginnings in 1977, the T_EX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best. Once upon a time, computers could deal only with numbers; then several years passed when they had numbers and uppercase letters; then they became able to deal with both uppercase and lowercase; then they became capable of working with letters of variable width; and by 1977 there were several systems that could produce very attractive documents. My goal was to take the last step and go all the way, to the finest quality that had ever been achieved in printed documents.

It turned out that it was not hard to achieve this level of quality with respect to the formatting of text, after about two years of work. For example, we did experiments with *TIME* magazine to prove that *TIME* would look much better if it had been done with T_EX. But it turned out that the design

of typefaces was much more difficult than I had anticipated; seven years went by before I was able to generate letterforms that I began to like.

The second major design goal was to be *archival*: to create systems that would be independent of changes in printing technology as much as possible. When the next generations of printing devices came along, I wanted to be able to retain the same quality already achieved, instead of having to solve all the problems anew. I wanted to design something that would still be usable in 100 years. In other words, my goal was to arrange things so that, if book specifications are saved now, our descendants should be able to produce an equivalent book in the year 2086. Although I expect that there will be a continual development of “front ends” to \TeX and **METAFONT**, as well as a continual development of “back ends” or device drivers that operate on the output of the systems, I designed \TeX and **METAFONT** themselves so that they will not have to change at all: They should be able to serve as useful *fixed points* in the middle, solid enough to build on and to rely on.

Today I'd like to brag a little, and say that I think that these goals of top quality and technology independence seem to be achieved; and volumes A, B, C, D, E tell everything about how it was done. Today I'm seeing these books for the first time, and I'm happy that all of you can be here to help me celebrate this event. These books are somewhat unusual because they describe themselves: They describe exactly how they were typeset. All of the formatting was done by the \TeX system described in volumes A and B. Also every letter and every symbol that appears in all five volumes, as well as on the covers and book jackets, was done by the **METAFONT** system described in volumes C and D. Volume E tells how I dotted all the i's and crossed all the t's, literally. If copies of these books were sent to Mars, the Martians would be able to use them to recreate the patterns of 0s and 1s that were used in the typesetting. Essentially everything I learned during the past nine years is in here.

All of the methods described in these books are in the public domain; thus anybody can freely use any of the ideas. The only thing I'm retaining control of is the names, \TeX and **METAFONT**: products that go by this name are obliged to conform to the standard. If any changes are made, I won't complain, as long as the changed systems are not called \TeX or **METAFONT**.

Volumes A and C are user manuals; I tried to write manuals that would suit users at all levels as they grow with the systems. And I also strove for

a high standard of excellence in the choice of the quotations from other works that are included at the end of every chapter.

Volumes B and D contain the complete program listings of \TeX and **METAFONT**. These books are specifically for computer scientists, not for casual users, but I'm especially pleased with how they came out because they represent an unexpected payoff of my research. This is something that I had no idea would be possible when I began nine years ago. As I wrote the programs for \TeX and **METAFONT**, I wanted to produce systems that would represent the state of the art in computer programming, and this led to the so-called **WEB** system of structured documentation. I think that **WEB** might turn out to be the most important thing about all this research—more important in the long run than \TeX and **METAFONT** themselves—because **WEB** represents a new way to write software that I think is really better than any other way. Using **WEB**, it was possible to write programs that are so readable that I think there already are more people who understand the inner workings of \TeX than now understand any other system that is as large. Furthermore I think it's fair to claim that **WEB** has made \TeX and **METAFONT** as portable, as maintainable, and as reliable as any other pieces of software in existence. The programs are now running and producing essentially identical results on almost all large computers; there are thousands of users, yet no bugs have been reported for more than half a year. I think there is at most one more bug in \TeX , and I'm willing to pay \$20.48 to the first person who finds it. (Next year the reward will double, to \$40.96, etc.)³

Volumes B and D also contain another innovation that improves on the basic **WEB** system previously available: Every pair of facing pages has a mini-index on the right-hand page, for quick cross reference to anything that's referred to on either page; this saves a lot of time thumbing through the master index at the end.

In recent years I've been making a pitch for programs as works of *literature*. Although there still is no Pulitzer Prize for the best-written computer programs of the year, I tried to write volumes B

³ (Editor's note.) A listener asked, how much had it cost to pay off the finders of bugs in the programs and errors in the books? Depending on how many checks were actually cashed, Don estimated the total to be between \$2,000 and \$5,000. It is doubtful that the checkbook in question is easily balanced.

and D in such a way that I would be a candidate for such a prize if it were actually given! More seriously, I intended these books to be useful to computer scientists for self study as well as for study in college seminars. Volume D, in particular, should make a good text for a group of advanced students.

The fifth volume, volume E, is the most fun of all. I hope you will all open a copy and riffle through the pages, so that you can see what I mean. **METAFONT** is a computer language that is not very much like any other, so my goal in this book was to provide lots of examples of how **METAFONT** can be used to produce fonts of reasonably good quality. Over 500 examples appear here; they cover every letter, digit, punctuation mark, and other symbol that was used in printing these books.

The fonts you get from these programs have the general name 'Computer Modern'. My colleague Charles Bigelow has contributed an introduction that talks about Modern fonts in general. The book explains how you can make your own personal variations of the fonts, which are designed with many parameters so that they can be generated in almost limitless variety. At the end of the book there are sample pages that show specimens of 75 standard Computer Modern typefaces; and thousands of additional varieties could be generated with ease.

Even if you don't read the **METAFONT** programs in this book, I think it's appealing just to look at the pictures of these constructed alphabets,⁴ and to 'know' that the program on the page facing each letter was what 'drew' that letter; it's all there. Somehow this gives a satisfying sense of completeness and order.

The most important thing I want to talk about this morning is *HELP*. I had lots of help — literally hundreds of people who volunteered to assist this project in significant ways — beginning with Hans Wolf of Addison-Wesley, who taught me the details of the Monotype systems that had been used to typeset *The Art of Computer Programming* in the 60s. I was especially fortunate in my work on font design to have had extensive help from world leaders like Hermann Zapf and Matthew Carter.

⁴ (Editor's note.) The pictures, it was pointed out, were generated separately from the text of the examples, and pasted in. If both the raster images and the text had been incorporated at the same time, it would have exceeded the capacity of the machines used to produce the book.

Another stroke of luck was to have outstanding research associates like David Fuchs and John Hobby. Furthermore my research project at Stanford had generous financial support, most notably from the National Science Foundation and the System Development Foundation. With so much help, it would have been very hard for my research to fail. And my wife Jill gave the most help of all. (Next month we will celebrate 25 years of marriage!)

One final note: People often ask me why **T_EX** and **METAFONT** are symbolized in these books by a lion and a lioness. When Duane Bibby first came up with the lion idea, I instinctively felt that it was right, but I never understood exactly why this was, until about a month ago when I was in the Boston Public Library. I passed by the magnificent stone lions on the library's grand staircase, and I thought: "That's it! **T_EX** and **METAFONT** try to be like these lions, fixtures that support a great library.⁵ I love books, and lions represent books!" No wonder I'm so happy when I realize that **T_EX** and **METAFONT** have already contributed to the making of several dozen books of fine quality; it makes me extremely pleased to think that this research will probably contribute to the making of many more fine books in years to come.

Comments on Document Design Prompted by the New TUGboat Format

Cal Jackson
California Institute of Technology

I've been looking at the latest issue of TUGboat and wondering if I should comment. I decided that I should. It is unfortunate that I reach such a decision when we have a guest editor; the responsibility has always been there.

I *think* I now have a *little* more understanding about what a designer and a typographer and a compositor are trying to achieve in the sense of basic qualities. Note "little."

The guest editor idea was (is) fantastic. And, I can't think of better people to do it than Kellerman and Smith. They're serious and demand the best

⁵ (Editor's note.) One is also reminded of the lions that grandly guard the entrance to the New York Public Library, which celebrated its 75th anniversary during this same week.