U.K. TUG, Oxford,
Sunday, 12 September 1999
Question & Answer Session with
Donald Knuth

Philip Taylor: I'm sure everybody here by now knows Don Knuth. Don has very kindly agreed to do a question and answer session this evening. In order to record both the questions and the answers, we have two microphones. One for Don's use and a radio microphone so we can capture your question on the tape, as well as the answer. Many thanks. Thanks Don.

Donald Knuth (DEK): O.K., I hope you can hear me.

Audience: Use the microphone.

DEK: No, I can't use the microphone, it doesn't work for me. It's only for the tape, but I will try to project. Since I have been retired for a few years and my voice isn't what it used to be, I'll do my best.

A part of the rules are that when you ask me a question, you give your name first. Another rule is that I get to ask questions too. Occasionally.

I did prepare one small thing, because, as coming to Oxford I re-read one of my favourite novels, a Dorothy L. Sayers mystery called *Gaudy Night*, which is all about Oxford. I wanted to read you just part of it, which has to do with typesetting. [*Laughter*] In Chapter 3 it mentions a Miss Lydgate, who had prepared her magnum opus, about — I don't know how to pronounce it — prosody, the study of meters in poetry. It said her handwriting was difficult to read, her experience in dealings with printers was limited but she had invented a novel and complicated system of notation that involved the use of twelve different varieties of type; and then she had all kinds of sheets in page proofs and so on. And she said 'Don't prick your fingers on that bit of manuscript that's pinned on, I'm afraid it's rather full of marginal balloons and interlineations, but I suddenly realised I could work out a big improvement in my notation, so I've had to alter it all the way through.' Then Harriet said comfortingly, 'Well, the Oxford University Press is no doubt accustomed to deciphering the manuscripts of scholars.' [*Laughter*] Now this work of Miss Lydgate appears to play a kind of minor role in the entire book and I have a question for you here, because it said her system of scansion required five alphabets with a series of pothooks for its expression, and this is a term I don't know. What are pothooks? If any of you can tell me ...

Audience: Spell it out.

DEK: It's spelt like pot hooks, P-O-T-H-O-O-K-S.

Audience: Pothooks are hooks for hanging pots.

Male Voice: Yeah, but in typography it's an 'S' like an 'S'.

DEK: Is it some kind of special symbol? O.K. So she had these twelve fonts plus pothooks, which she used to typeset. So anyway you have some idea. Does this mike actually magnify? Can you hear me a little better? O.K. Well hello. I was reading from one of my favourite books..., but that's enough. I have one more thing prepared then I will go for questions. This was for Phil, I wanted to show you.

Laughter & Clapping: (Don shows he's wearing a T-shirt under his shirt that celebrates the Aston UK TEX Archive, 'oldest and best')

DEK: I always try to wear the appropriate T-shirt for the day. O.K., your turn now.

Dave Pawson: My name is Dave Pawson and I have an answer to your question. I was brought up in the 1940s and 50s in the north of England and one of the houses that I moved into was built in the mid 18th century, and around the coal burning stove and above it and beneath the mantelpiece there were a number of hooks and they were the 'pothooks' for hanging the pots on. [*The OED discusses this usage.*] Does that answer your question?

DEK: O.K. So this must be a similar shape that you would use for the symbols. I think that someone said that shorthand uses pothooks — some systems of shorthand. Thank you.

Dave Pawson: I think it's what you get when you type a left angle bracket into TEX and you are expecting a straight translation, and one of your assumptions in the big blue book comes unstuck, because it comes out like a pothook. [*Laughter*] Is that true?

DEK: Might be. The other thing I wanted to say is thank you very much to Phil Taylor for arranging that we could have the 10th anniversary celebration during the brief window of time that I could be in the U.K. I don't get to travel very often, and so now I get to remember not only the 10th anniversary celebration of TEX Users Group in America but also the one from here. At the 10th celebration in America we had the president of TUG dressed as

Thanks are due to the following:
Philip Taylor, for tape recording the Q&A session;
Jonathan Fine, for managing the transcription;
Pam George, transcribing from tape and making corrections;
Don Knuth, for diligently and promptly correcting proofs.

META the Lion — that was Bart Childs — and now here we have Mrs. TEX as well so now my delight is complete.

Sebastian Rahtz: On your recommendation last year I bought a copy of *Life: A User's Manual* by Georges Perec, which I am still trying to read. I wonder while I'm still trying to read it whether you would like to recommend a film or a piece of music which has equal meaning to you as Georges Perec's book.

DEK: I guess when I made my home pages, a few years ago, one of the pages listed books that I was recommending to read. There's this incredibly different book by Georges Perec called *Life: A User's Manual*, which is a combination of many different kinds of artistry. It has a mathematical basis, but still becomes, I think, a great work of fiction. He developed this book with 99 chapters. (There should really be 100 chapters, but each chapter was based on a lot of mathematical constraints and one of his rules was that you had to break one of the rules, so naturally it has only 99 chapters instead.) It is the story of the people who live in an apartment block in Paris. There are 10 floors and 10 apartments on each floor, and you go through the apartments — actually some of the apartments have several rooms — but you go through the apartments in the order of a knight's tour. Eventually you find out about the lives of all these people, and there are many other very interesting constraints that he put into the book. Each chapter is a little short story, kind of independent of the others. Now you are asking if there's something else analogous, in the domain of music or ...

Sebastian Rahtz: Film.

DEK: ... or film. O.K. The closest thing in film is this new movie *Run Lola Run*, from Germany. If you were to do it the way Perec did it you would have many more chapters, but *Run Lola Run* gives you a story three times. The first time ends in disaster and so Lola says 'No take me back. Let's do it again.' So we start over and she does something slightly different in the first scene and then we go through the whole story again, but everything happens five seconds later, so certain accidents don't occur in the streets and the whole plot changes. At the end of the second telling of the story it's another disaster, not for her, but for her boyfriend, and that's too terrible to accept. The third version of the story leads to a happier fate.

In music I suppose I think of a theme that had been proposed to Bach, I think by one of the noblemen of his time. He supposedly improvised a theme on that melody spontaneously, but then he was fascinated by it afterwards. During the last year of his life he prepared a manuscript that he left unfinished at his death, called the *Art of the Fugue*. That work is analogous to Perec's, because the idea is to make a thing of beauty while working within tight constraints.

Sebastian Rahtz: Good, thank you. [*Pause*]

DEK: Well, if there are no more questions ... [*Laughter*]

Elizabeth Gilliart: What are you going to write next?

DEK: Yes, I tend to be writing about a page a day and so there's nobody alive that has read everything I've written, except perhaps me. But I'm just finishing now a book that is incredibly specialised. It will be in the Springer Lecture Series, Lecture Notes on Computer Science, and it's called *MMIXware*. It's a set of computer programs to simulate the new RISC computer that I designed this year. MMIX is a computer for the new millennium. The first 'M' is for millennium and it replaces a computer called MIX that I used in my books on computer programming. I had the privilege of working with the designer of the DEC Alpha chip, Dick Sites, who is one of my students; John Hennessy, the designer of the MIPS chip, also was a participant in this design, and a few other people in Silicon Valley. We came up with something I think would be a fairly good machine to build in about ten years. It tries to be the cleanest computer design, and easy to learn, fairly nice to look at, and to make theories about; there is a group of people that are helping me use this new computer to rewrite the algorithms that I had written for the old computer. *MMIXware* is a set of software programs that make MMIX live even though it hasn't been built yet. The exciting thing to me is the pipeline simulator, which is a meta simulator — which means that it can simulate millions and millions of different kinds of possible implementations, or even implementations that nobody knows how to build. You can say for instance how much memory it has, what kind of caches it has, what kind of strategies the caches use to remember the recent information; a pipelined computer has different functional units, you can say how many multipliers you have and you can define a functional unit to handle any subset of the 256 operation codes; you can have any number of functional units, you can issue any number of instructions simultaneously, you can look ahead different ways and control all kinds of things. And then you can find out how fast your programs run. Now if we had that machine we could put TEX

onto MMIX and see how fast it works with different kinds of caches. This program for the meta simulator is probably the most difficult computer program I ever wrote.

To me it was most interesting because it depends on the idea of literate programming that I used when I worked on TEX; for me perhaps the greatest spin-off of TEX was this idea of literate programming, which has helped me write computer programs of all kinds. I don't think I could ever have written the MMIX meta simulator without literate programming; it would have been too mind boggling. I would not have been able to get the whole thing together and debug it; it would have been too much of a mental strain, overburdening my head. In the past, literate programming has often helped me write better programs, but here for the first time it was crucial, or I couldn't have written the program at all. I don't think *MMIXware* would have been possible without a good documentation language to help me understand what I was doing as I went along. So this will be a book about 400–500 pages, but it's mostly just the typeset version of these literate programs for that new computer.

Then also I'll be finishing a sort of sequel to *Digital Typography*. *Digital Typography* is volume three of a set of collected papers; all the scientific papers that I've written are being divided into eight categories. The first book contained the things I wrote about *Literate Programming*. The second book was called *Selected Papers on Computer Science*; those were the papers I wrote for audiences that weren't primarily computer scientists; it collects the general works. And then the third volume was *Digital Typography*, and I told Phil I'd come here tonight because I'll do anything to sell copies of that book. The fourth volume will be called *Selected Papers on Analysis of Algorithms*, and those are my mathematical papers for what I think is my main unique life's work — the study of computer methods in a quantitative way: How good are particular computer methods, from a definite exact mathematical point of view? That book is estimated to be about 750 pages, and it will contain, I think 37, 38 papers on that subject. The material has all been scanned and put into TEX form. But I have to spend a few months in spare moments going through dotting the i's and crossing the t's and making the index, putting the bibliographies in a consistent format, finding out people's middle names for the index, things like that. And I also go through every paper and put it into the form in which I would like it to be remembered. So if a paper was written in the 70s and I used sexist pronouns, I change that; I try to rework it so that instead of saying 'he did it', I'll say 'they did it' or something. Also I change 'which' to 'that' a lot. It's an American thing.

Male Voice: Why?

DEK: It's actually because of an Englishman named Fowler who wrote *The King's English* and gave rules for 'which' versus 'that'. The people in America believed him and they started teaching English classes, and it was taken over then by the *New Yorker* and other well-edited journals until finally the copy editors who used Fowler's rules — Fowler's book came out in the 20s I think[1] — the copy editors who used Fowler's ...

Audience: Roberts' rules!

DEK: Whose rules?

Audience: Just after Roberts' rules!

DEK: I never heard of Roberts' rules of English, only *Roberts' Rules of Order*[2] — and I think your comment is out of order. Anyway, Fowler essentially gave an algorithm that pretty much boils down to this: Look at what comes before the word 'which'. If it's a preposition or a comma, then it's fine; 'which' can also be a pronoun. But otherwise you change it to 'that'. Well, it got to the point where almost all of the magazines in America and most of the newspapers by the end of the 70s were using 'which' versus 'that' in Fowler's recommended style. This was because the copy editors had risen through the ranks and won the battle. I had written the first draft of *The TEXbook* in the old way; but Guy Steele who was visiting Stanford from MIT, marked it all up, everywhere I had a wicked 'which'. Simultaneously my copy editor from Addison-Wesley was revising volume 2 of *The Art of Computer Programing*, which I had in TEX form before the 2nd edition came out. I was doing the typesetting in 1979 and 80, and that copy editor was also a member of this new generation. So that's when I learned the algorithm for wicked 'whiches'; I could do a search with a text editor, checking to see if it's following a preposition or a comma.

Soon I became very sensitive to this, as were a lot of Americans. I started to be irritated when people would quote a sentence from my earlier papers where I used a wicked 'which'. It was even hard for me to read the New English Bible because I would silently have to translate 'which' to 'that'. I go into

[1] H. W. Fowler *The King's English* (Oxford 1907). H. W. Fowler *A Dictionary of Modern English Usage* (Oxford 1924)

[2] Somebody named Paul Roberts wrote *Understanding Grammar* in 1954, but I don't think it was terrifically influential.

more detail in my book *Mathematical Writing*, published by the Math Association of America. The British perspective I understand is different still, but in America it's very much a stylistic lapse that calls attention to itself if you don't follow the convention. You can go through now say 100% of the magazines in the last 10 or 15 years and you very rarely see a wicked 'which'.

James Foster: You said that the MMIXware book was for Springer; did they ask you to use their LaTeX 2.09 style files? [*Laughter*]

DEK: I take it that LaTeX 2.09 style files are a joke because that's an old version of LaTeX?

James Foster: We had to use them recently.

DEK: I was given a free hand here. But I'm supplying camera copy so we'll which is a special version of CWEAVE that makes little indexes on the right-hand page. One of the chapters in *Digital Typography* talks about this: 'Mini-Indexes for Literate Programs' is the title of that chapter. Basically when you are reading a computer program every name of a variable on the page that you are reading is either defined on that page or you can find it in the mini-index. And the mini-index will tell you that the variable is, say, defined in section 5, and it is a procedure name or a constant and so on. It gives you a quick reference. It's an idea that I picked up from some textbooks on languages, where people would be learning French or Russian. Given a short story in some other language, every word in the vocabulary that you might have to look up in a dictionary appears in an index on that page. With CTWILL I have a bunch of macros that check the uses and definitions of everything on a page. The macros automatically prepare mini-indexes; there is a separate pass to sort the words, but the first pass does the layout. Some of the more difficult — probably the most difficult — TeX macros that I ever wrote are involved in that. I call this program TWILL because it's sort of a double WEAVE. Anyway the CTWILL program gives me the output that I have to send to Springer.

So no, I won't be using their style files. I did one other book in their series but that was some years ago, before they had learnt to insist on those things I suppose. For some purposes of course these extra format restrictions are not only for your creativity but also for electronic archiving. It's not the kind of restrictions that Perec would have added, but I imagine the publisher sees that it's to their advantage to have as many things in that style as possible. It depends on what you are doing, because that can also stifle what you want to say. One of the things uppermost in my mind from the beginning of TeX was that I would have the freedom to introduce a new notation if it was the right thing for the subject I was writing about. If I wanted to make up a new notation I wouldn't have to go through any middlemen who wouldn't understand the notation. I would be able to typeset it and I would know that it was getting through in the way I wanted it.

Philip Taylor: I have some bad news: They really do want us out of here by half past ten, so there's time for just a couple more questions.

Jonathan Fine: I hope it's not too late to ask this question. Why did you introduce category codes?

DEK: Well, if you look at this book *Digital Typography* it shows the original draft of TeX that I made the first night when I stayed up late typing a proposed design. The feature was even more awful in those days, it handled not only category codes but all kinds of penalty amounts and other parameters; anything that I didn't know how to fix as a constant in TeX I just left for later, so that I could change it. But you know originally I didn't think of having many users; TeX was a system for me and my secretary, so I wasn't thinking of things in much generality. So the same mechanism that I had originally for category codes was also used for things like how many characters before breaking for a hyphen, what was the penalty for widows, et cetera, all done with the same mechanism 'chpar'. C-H-P-A-R it was called: Change Parameter. I had no idea of generality in the first place.

Then later on, as I saw the applications Math Reviews were making, I realized that I'd better have some way to allow more flexibility... Jonathan, you'd be interested to know that the very first implementation of proto TeX was done without recognizing control sequences as words; each character was read in as a character by itself and then the 'hash table' lookup would go on afterwards. The first implementation, which was done by my students in the summer of 77 while I was in China, was very much like Active TeX in that sense. We found out that we could make the program run a lot faster in its inner loop if we distinguished control sequences from ordinary text, but originally every character was active in the very first draft. In this book *Digital Typography* I resurrected the computer files that I had used when first getting my thoughts in order.

Dominik Wujastyk: Don, you made a change in TeX when you made it able to read an eight-bit character, and that was very important for European users and others. I completely respect and like the idea that TeX is fixed, that you've finished with it as

it were for the moment, and that it is a fixed point and it's not going to change anymore. However it does look as if TEX will have to go Unicode. There is of course Omega already; I've never used Omega, I haven't had the courage to take it on yet, but I think it's there in my future, and of course other people are using it. I just wonder about whether it's going to just always be Omega or something like it and TEX, and they're just going to become separate things and go their own ways; or could you give a Unicode version of TEX some imprimatur or some special blessing, so that it becomes the one that everyone uses. I wonder whether all the development efforts can somehow be brought together again.

DEK: Yeah! It seems so difficult. I'm a great fan of Unicode, but I also know enough about it to know that it's incredibly complicated, and that I would never have a system based on Unicode that I would be able to say has no bugs in it because of the extra complexity. Well, I like TEX, I like having a program that is, if not 100% reliable, it's 99.9999 — it's as reliable as anything. So solid that we can build on it. This means however that I'm not supporting all of the important languages that Unicode supports. Still the job of doing all of those in one system is so incredibly hard that I don't know where the expertise is going to come from to get it to such a well debugged level. For me to put my imprimatur on something would require so much work, I would have to check that the thing had been done right, and there is so much involved in getting it done right. As you know, Unicode 3.0 which will be coming out early next year, really covers almost all the languages of everyone alive today; they have filled in the last gaps, they've got Burmese and the Maldive Islands, Sri Lanka, the places where the political difficulties were; they have several thousand extra Vietnamese and Chinese characters and so on. And Yi, and Mongolian, and native American — various Inuit and Algonquian languages — are all there now.

But each of these languages has special difficulties involved in the typesetting. It's not just a matter of getting the symbols; all kinds of ligatures and things must go in, and rewriting of characters. Many of the languages have no spaces between words, and special hyphenation conventions, and a total user community of a few thousand. Moreover, all the people who do use some of these languages are educated enough that writing in English gives them more job security; thus the more they do to make it possible for everybody else in their group to use the system, the less chance they have of being uniquely able to do anything. So it's going to be hard to support this commercially; it's surely going to be a volunteer effort. The effort is not only an order of magnitude more difficult than what I had to do, but it also has to be done pretty much as a labour of love, as I did it. So it looks to be a while before it could converge like that — not that it's impossible, but I myself wouldn't be in a position to bless it. All I can do is provide an example of one of the world's nearly bug-free programs so that other people can try to emulate the good points and correct the bad points.

Well, thank you very much. [*Applause*]

Philip Taylor: Don, I'd like to thank you very much indeed, on behalf, not only of the Committee of the U.K. TEX Users Group, but of every one of the members here who, I'm sure, are absolutely delighted that you have spoken to them. Thank you very much indeed for your time, for joining us. Jill, thank you very much indeed as well for coming along. It's been a great pleasure to have you both in our midst. We wish you a very safe and happy stay in the U.K. for the rest of your trip. Thank you very much indeed.

– – ∗ – –

Editor's note: Earlier interviews and Q&A sessions with Don can be found in these issues of *TUGboat*:

- **7** No. 2 (1986): The "coming out party" for *Computers & Typesetting*, May 1986
- **13** No. 4 (1992): Q&A with the Nordic group and conversation with Roswitha Graham, November 1991
- **17** No. 1 (1996): TUG '95 Q&A
- **17** No. 4 (1996): Q&A Amsterdam (NTG) and Prague ($\mathcal{CS}$TUG), March 1996
- **21** No. 2 (2000): Interview by "Advogato", January 2000

Several of these sessions are already posted on the TUG web pages, and the rest will be posted when time permits.