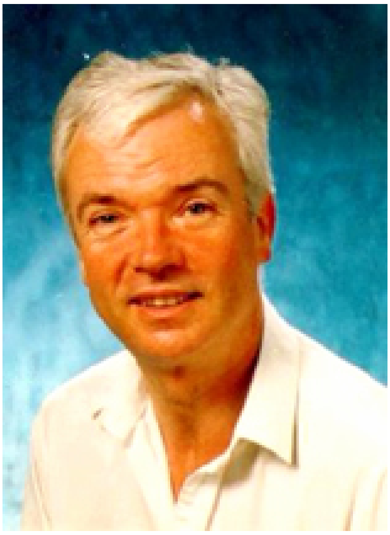
Philip Taylor

Philip Taylor is a long time TEX user/developer who chooses to work outside of the LATEX framework. He is also a member of the TUG board.

[Interview completed 25 May 2005.]



Dave Walden, interviewer: Please tell me a bit about your personal history independent of TEX.

Philip Taylor, interviewee: Professionally speaking, I have had a very small number of jobs, each of which has led fairly naturally to the next. I started my working life at the age of 16, as my parents were unable to afford to keep me at school any longer, and my first employer was Post Office Cable and Wireless Services, External Telecommunications Executive (`http://www.spencerweb.net/telegraphs/html/electra_house_old___new.html`). I started life as a (very!) humble "Youth-in-Training", and progressed via Technician IIA to Technical Officer. After seven very happy years with the G.P.O. (this would now be "British Telecomms"), I left when they wanted me to leave the duties I loved (Control Room & Electronics) and work in a Strowger telephone exchange known (not particularly affectionately) as "The Pickle Factory". I then joined Molins Ltd (a tobacco-handling machinery design company) as an Electronic Design Engineer, but left after just over two years when I realised that working for a profit-oriented organisation clashed too severely with my own work/life perspectives and ethics. From there I joined Westfield College (University of London) as a free-lance Computer Engineer, was made redundant after about three years, and moved to Bedford College (also University of London, in London's "The Regent's Park") where I had been teaching part-time, as a Computer Analyst/Programmer. I remained at Bedford College until its closure in 1985, at which time it (and I) merged with Royal Holloway to form Royal Holloway and Bedford New College (now Royal Holloway, University of London). At RHBNC I started to develop my awakening interest in computer typesetting (see next section), and when the demand for high-quality printed material began to diminish, transferred what I could of my hard-acquired skills to the world of electronic publishing, with particular reference to the World-Wide Web. I am currently Webmaster at Royal Holloway, University of London (`http://www.rhul.ac.uk/`).

DW: How did you first get involved with TEX and its friends?

PT: Peter Jackson, a friend from my Westfield days (affectionately known as "Pute", for reasons I have never understood) shewed me one day in (I think) 1986 some computer-typeset output produced using equipment *identical* to that which I had been using (Digital VAX/VMS running on a VAX 780, with a Digital LN03 laser printer) but which so far surpassed the quality I had been able to achieve that I was literally gob-smacked. I asked Peter how it had been produced, took a copy of Kellerman & Smith's TEX implementation for VAX/VMS with me on 1600 bpi magnetic tape, returned to Royal Holloway and started to implement TEX on our systems. I have never looked back.

DW: You have been a long-time contributor to the TEX community. Will you please tell me how this came about and enumerate and describe your contributions.

PT: I certainly won't "enumerate and describe my contributions"! How it (or they) came about is rather easier to explain. There are some things in life to which one is inextricably (and inexplicably) drawn, rather like a moth to a candle, and I have been singularly lucky in my professional life in that I have been allowed to spend much of it exploring some of these. During the last twenty years, two such things have figured more prominently than any other: the first was VAX/VMS, and the second TEX (my *very* first love was Algol-68). In many ways they could not be more different, and Don himself must have realised this for he once said that, in his opinion, Macro-32 (the assembly language for VAX/VMS systems) was possibly the worst assembly language in the world! I was staggered by this pronouncement, since I have always thought of Macro-32 as being possibly the finest assembly language ever developed, but I just had to accept that whilst Don and I agreed about TEX, we would never agree about VMS ... I must one day ask Don his opinion of Algol-68: it would be interesting to know into which of the two categories (awful or brilliant) he would place it, since it's inconceivable (to me) that he would place it at anywhere other than one of the two extremes....

So, fairly early on in my exposure to TEX, I came to realise that it and I were "as one", as it were: I somehow *understood* the underlying philosophy of TEX, and once that is understood, understanding how to use it just seems to come naturally. The primitives that were so difficult for some to grasp (`\expandafter`, `\futurelet`, `\catcode` and so on) were for me not "difficult" *per se*, but rather were an absolutely fascinating challenge, to the bottom of which I just had to get. The more I explored, the more I learned, and the more I learned, the more I enjoyed what I was doing.

Before too long, I started to receive invitations to teach TEX, and these were, for me, my real "contribution" to the TEX world, if indeed I have ever made one. The packages that I have written are few, and of restricted interest (`cropmarks`, `letterspace` and so on) but the real joy for me was not in writing the packages (I usually did so only because I, or someone with whom I was involved, needed functionality that did not pre-exist in TEX) but in learning ever more about how TEX functioned and then passing that information on to others.

On re-reading the above, I realise that there is one contribution of which I feel sufficiently proud not to be ashamed to mention it. When I started work at Royal Holloway, one of the people with whom I most enjoyed working was Dr Malcolm Smith, then a lecturer in the Department of French (and a very keen cyclist, who would think nothing of cycling over the Alps to consult a single book in a distant French library and then cycle back again) who later became Head of Department. Malcolm founded (with financial support from the College) Runnymede Books (motto: *Scholarship made accessible*), the aim of which was to publish French texts (`http://isbndb.com/d/book/antiquitez_de_rome.html`) at a price which students could afford. I worked with Malcolm on the design and typesetting of these, and the difference in appearance between the last and the first is very noticeable indeed. My "pride" in this has nothing whatsoever to do with the quality (or otherwise) of the typesetting itself, but rather with the fact that I was able to assist Malcolm in his self-appointed task of making scholarship truly affordable (and hence accessible). Malcolm is sadly no longer with us: he died of stomach cancer in his early forties, and is much missed. Other scholars with whom I have had the greatest pleasure in working (and continue to do so to this day) are Ian & Rosalind Gibson (the latter the author of the just-released second edition of *Principles of Nutritional Assessment* (`http://nutrition.earthlight.co.nz/`), and Julian Chrysostomides & Charalambos Dendrinos who are currently preparing the *Lexicon of Abbreviations & Ligatures in Greek Minuscule Hands* (`http://www.rhbnc.ac.uk/%7Euhra052/handbook.html`). I was also very pleased to be able to typeset Mr K D Somadasa's *Catalogue of Sinhalese manuscripts in the Wellcome Institute for the History of Medicine* (`http://tinyurl.com/ax4nj`). There are a number of other scholars for whom I have had considerable pleasure in designing and typesetting their books, but sadly too many to list here.

DW: Please tell me about your involvement in the UK TEX Users' Group and, for anyone who doesn't know much about it, a little about the group itself. Also, please tell me something about your involvement in the NTS group and, again, a little about that group.

PT: The UK TEX Users' Group (`http://uk.tug.org/`) (or UK-TuG, as I usually abbreviate it) was — if my memory serves me accurately — the brainchild and creation of Malcolm Clark. I *think* that the pre-formation meeting took place in London (certainly I recall it being well attended) and the following year "Exeter TEX88" took place (I still have the tee-shirt, so it was easy to check the date!). My memory of the following years is pretty vague (I attended the TUG meeting at Stanford, but that has little to do with UK-TuG) and the most important thing that I can recall in the early years of UK-TuG was Peter Abbott's creation of the UK TEX Archive (the forerunner of CTAN). Peter was assisted in this by a number of people: Brian Hamilton-Kelly, Neil Kempson and David Osborne amongst others. We "archive assistants" would occasionally be summonsed to Aston by Peter, who invariably entertained us very well indeed, and I remember one occasion when we were dining at Aston University that it took twenty minutes to convince the servery staff that I wanted plain, ordinary, straight-out-of-the-tap Birmingham water (actually Rhayader water, from the Elan Valley, but they may not have known that), that I knew what it tasted like, that I *didn't* want bottled water, and that I would get my tap water even if it took all day! We also had some splendidly memorable meals at *The Last Days of the Raj*.

Anyhow, back to TEX. I became involved with the UK TEX Users' Group fairly early on its life, and eventually became its Chairman for a while, taking over (if I remember correctly) from Robin Fairbairns. Most, if not all, of the well-known names in UK-TuG served on the committee at some time or another: apart from those already mentioned, others who immediately spring to mind are Kaveh Bazargan, Sue Brookes, Sebastian Rahtz, Jonathan Fine, Kim Roberts, Chris Rowley, Dominik Wujastyk (these are in no particular order, and all omissions are entirely accidental), and for virtually the entire time I was associated with the Committee Peter Abbott was its Treasurer. UK-TuG was for many years a very active group, with organised meetings, tutorials and so on, but recently the level of activity has markedly declined, which *may* be an indication of a declining interest in TEX within the UK population but may equally be an indication that the Committee is in desperate need of new blood (this is in no sense intended as a criticism of the present committee, since the decline in the level of activity was already noticeable at the time that I stood down as Chairman). For several years we published a journal, *Baskerville*, but that too has now disappeared, apparently without trace.

The NTS group is another kettle of fish entirely. This group owes its existence (and the TeX world owes a very great deal) to the foresight and vision of Joachim Lammarsch, for many years the President of DANTE e.V. (`http://www.dante.de/`) During the period leading up to the Hamburg meeting of Dante in 1992, Joachim wrote to all the TeX activists that he knew, asking if they would be interested in co-operating to design and develop a successor to TeX. I was amongst those who responded positively, and we who so did were all invited to the meeting in Hamburg at which the group was created. The group included (again, in no particular order, and with apologies for any accidental omissions) Joachim, Friedhelm Sowa, Rainer Schöpf, Peter Breitenlohner, Jiří Zlatuška, Bernd Raichle, Joachim Schrod and myself. Joachim Schrod and Rainer Schöpf left the group fairly early on in its history and the remaining members rather loosely partitioned their activities into two projects: the NTS project itself (with "NTS" standing for "New Typesetting System" and the ε-TeX project (where the "e-" can be thought of as indicating "extended" or "enhanced"). For some time the group concentrated mainly on ε-TeX, with Peter Breitenlohner taking the technical lead, and in many ways ε-TeX is also the most significant result of the group's work, since it now forms the engine on which LaTeX is based. The NTS project was put on hold until adequate funding could be secured, and once this was achieved (mainly through the generosity of Dante e.V. and TUG, but also as a result of an incredibly generous donation by one individual who wished — and wishes — to remain anonymous), Karel Skoupý was hired as a programmer. Karel's brief was to reverse-engineer TeX, and thereby to design and implement a TeX clone which behaved *exactly* as did TeX but without re-using any of Knuth's code (and thereby avoiding coming into conflict with Knuth's express wish that he, and only he, make any changes whatsoever to TeX). Karel did this very well (albeit taking rather longer to achieve his and our goal than he had originally forecast), working mainly under the immediate direction of Jiří Zlatuška but with regular meetings with the rest of us to ensure that things were proceeding as planned. Sadly, despite the best of intentions, the NTS project started to founder for reasons which I will not go into here (this is water under the bridge, and it will serve the TeX community much better if we put past disagreements behind us and look to the future rather than the past) and the status and future of NTS are now somewhat uncertain.

One point which might not be immediately clear from the above is why the group undertook two separate projects: if it was possible to extend (or enhance) TeX simply through the medium of an additional change-file (and a concomitant change of name), why did we need to re-implement TeX as NTS? The answer is that making extensions and enhancements to TeX itself (whether through the medium of a change-file or whether using some less rigorous route such as "diff"s) is not easy: the TeX program, although incredibly well documented and despite Don's very careful adherence to the precepts of "Literate programming", remains a rather opaque piece of code with which only the most diligent can interfere without causing unexpected (and undesired) side-effects. The group were very lucky to include Peter Breitenlohner and Bernd Raichle amongst its members: Peter and Bernd are two of the *very* few people (there are probably fewer than ten in the entire world) who are sufficiently familiar with TeX-the-program that they can make changes which have *only* the desired effect. The underlying idea of NTS therefore was (and is) to re-implement TeX not only so as to avoid conflict with Knuth's wishes (this could far more easily be accomplished by the simple expedient of a change of name, *cf.* ε-TeX, pdfTeX and so on) but also to re-implement it in such a way that modifications could *easily* be made. It remains to be seen whether NTS has truly accomplished this *desideratum*. The other question which is frequently asked concerning NTS is whether the group were right to insist on Java as the language of implementation: our decision was primarily based on two key issues — (1) Java's portability (write anywhere, run anywhere) and (2) its standardisation (there *is* only one Java, since Sun owns the name). We discounted the performance penalty which the use of a semi-compiled language implicitly incurs. With hindsight, we may have overstated the importance of the former and understated the importance of the latter, yet (in my personal opinion), no other language which either existed at the time or which has come into existence since then offers any significant advantage over Java. As an aside, programming language development seems to have slowed dramatically: when I was younger, new programming languages appeared with remarkable regularity (Fortran 4, Fortran 77, Fortran 9x, Pascal, Modula, Simula, Oberon, Algol 60, Algol 68, . . .) but since "C", its derivatives and its object-oriented cousins made their appearance, development seems to have slowed almost to a standstill: new interpreted languages appear from time to time (Python, for example) but compiled languages which offer genuinely new insights into programming methodology seem remarkably few and far between. I do *not* believe that this is because "C" and its derivatives are perfect: indeed, despite its almost universal adoption, I feel that "C" is in many ways fundamentally flawed, but that is rather off-topic for this interview.

DW: You have served as an officer in the UK TEX Users' Group and the TEX Users Group. You have participated on the internationally staffed NTS group, and you chaired the program committee of a TUG Annual Conference (Toruń, Poland, 1998). My point is that you have seen the TEX community from a quite international perspective and from a perhaps unusual variety of functional perspectives (research and development, education, governance, system administration, etc.). I am interested in your perspective on how the world-wide TEX community has evolved and how you think it should continue to evolve.

PT: I think that "evolution" in this context is distinctly non-linear: there was a very rapid evolutionary spurt early on, characterised by the formation of the various national groups (or "language-based" groups, as some would characterize them), a process which continues today but at a considerably slower rate, and then for a long time there was a period of stasis during which the national groups undertook the combined tasks of proselytisation, recruitment and education, tasks which are now in some cases beginning to falter . . . . In parallel with this, but starting somewhat later, various pan-national initiatives were launched, of which by far the two most significant are the development of CTAN (the "Comprehensive TEX Archive Network"), which started life as "The UK TEX Archive", *q.v.*, and the TEX Live series of CDs and (more recently) DVDs. Other pan-national projects which cannot pass without mention are the LATEX 3 project, which in practice is really the "LATEX 2_ε implementation and development project", and the Omega project, which might be characterised as a Unicode-based derivative of TEX.

Whilst speaking of pan-national TEX activities, one truly remarkable phenomenon is the *de facto* international status of the Polish TEX User Group's annual meeting at Bachotek (near Brodnica, Poland). This annual meeting, which takes place at the beginning of May in an idyllic forest setting by the side of a tranquil lake, was originally simply the annual meeting of GUST (`http://www.gust.org.pl/`), the "Grupa Użytkowników Systemu TEX". Very early on, however, the GUST Board and Secretariat took the remarkable step of inviting known TEX devotees from other countries (the UK, Germany, the Netherlands, Hungary, Lithuania and so on) and many of us accepted. Once there, we were completely hooked: BachoTEX (as it is now known) is a truly addictive activity, once experienced never forgotten, and what was at first simply a national annual meeting is now a regular meeting place for TEXies from around the world. Quite apart from the quality of the presented papers (which is invariably exceptionally high), BachoTeX is simply a joy to attend: there are bonfires and singing most evenings, people congregate in each other's log cabins to talk TEX, to eat and drink, or simply to socialise, the lake and its environs supports a splendid variety of wildlife (catching, photographing and releasing the many grass snakes which can be found around the lake is one of my greatest pleasures); in summary, the best word I can find to describe BachoTeX is simply "unique". If you've not yet experienced one, make it a top priority to do so: you will never regret it!

Returning to the main theme of the "evolution" of the world-wide TEX community, and considering this at a national rather than an international level, probably the most significant activity which has taken place (and continues to take place) is the "regionalisation" or "localisation" of TEX, primarily through the design and implementation of language-specific fonts. When Don designed and implemented the AM series of fonts (these were the forerunners to the more widely known CM fonts: AM is "Almost [Computer] Modern" whilst CM is "Computer Modern", both being based on the Monotype 8a design), he did so in a way which was the very model of orthogonal design: any diacritic ("accent") can be placed on any character ("glyph"), and TEX will adjust the position to suit (this is, of course, something of a simplification but it will suffice for the present discussion). A few special characters that could not easily be composed of individual elements were included (e.g., the Polish "dark L": "Ł", "ł"), which typographically is like an "L" with an oblique slash and which phonetically lies closer to the "W" sound of British Cockney than to the clear "L" of RP in words such as "roll" and "elk"), but in general the underlying idea was that accented characters would be constructed on the fly. Whilst this worked reasonably well for some western European languages (French, German, and so on; Dutch/Nederlands less so, since there is no "ij" ("IJ", "ij") ligature in CM), other languages such as Polish and Vietnamese were considerably less well served. The Poles needed an "ogonek" (think of it as a reversed cedilla: "ą", "ą"), whilst the Vietnamese needed not one but two diacritics on a single base glyph ("ế"), one to indicate a change of vowel sound and the other to indicate in which of the six possible tones the word is to be pronounced (Silvio Levy, amongst others, had encountered this latter problem a long time ago when designing and implementing TEX fonts for typesetting Classical (polytonic) Greek, for which one needs "breathings" as well as conventional diacritics and an iota subscript). As a response to this, the Poles, the Vietnamese, and many other nations, designed and implemented one or more TEX-compatible fonts which met *their* needs rather than being a general purpose font suitable for many languages but "ideal" only for those that eschew diacritics (English, for example!). Whilst in some cases this has been accomplished simply by re-engineering Computer Modern to include national-specific glyphs, in other cases (Polish is a good example) the designer(s) have jettisoned CM completely and derived entirely new TEX-compatible fonts based on traditional fonts from their own national typesetting tradition (Antykwa Toruńska and Antykwa Półtawskiego, for example, in the case of the Poles).

And what of the future? How should "the world-wide TEX community ... continue to evolve?" That is not for me to say! How it *should* evolve is for it to decide, not for any one individual to prescribe. All I can do is to express a wish: that TEX users world-wide continue to recognise and publicise the fact that TEX's philosophy is, and remains, at the very leading edge of typesetting technology. Whilst Franklin Mark Liang's hyphenation algorithm has now been incorporated into high-end word-processing and desktop publishing systems, TEX and its derivatives are virtually unique in terms of their programmability and scriptability. People such as Hàn Thế Thành have done more than most to ensure that TEX continues to have a rôle in the 21st century, firstly by integrating the generation of PDF (as opposed to DVI) into (pdf)TEX itself, and secondly by their research into ways in which elements of Herman Zapf's HZ algorithm (for micro-typography) can be integrated into a TEX derivative. The TEX world needs more such people, since if we are over-zealous in respecting Don's wish that TEX remain unchanged (modulo essential bug-fixes made by himself) for perpetuity, we will end up with a museum piece rather than an archetype of leading-edge technology. Yes, TEX itself must remain frozen in time, but the TEX community must adopt (and adapt) TEX derivatives with the same enthusiasm that it first adopted TEX itself.

DW: In your paper "Computer Typesetting or Electronic Publishing? New trends in scientific publication" (*TUGboat*, Vol. 17, No. 4, 1996, pp. 367–381) you give significant mention to the ARPANET RFC (Request for Comments) mechanism. As I have become acquainted with the TEX world over the past eight years, I have been disappointed that it appears that there is no TEX mechanism parallel to the RFC mechanism. It appears to me there are lots of fragmentary conversations on comp.text.tex, occasional proposals in conference proceeding, *TUGboat* or other journals, and undoubtedly lots of memos within different activity groups, but no unified mechanism for proposing a new "standard". First, am I right about my impression? If so, do you think there is any possibility of a more visible, standard mechanism for advancing the coherent development of TEX, or is the TEX world simply too fragmented?

PT: Oh, what a question! I think that we need to start by considering TEX itself, how it came into existence, and why it is what it is. I would suggest that TEX is what it is for one main reason: it is the work of one man (albeit a genius, but one man nonetheless) who was able to develop a very significant amount of time to designing and developing a program without the need to (a) justify what he was doing to anyone else, and (b) consult anyone else before making decisions. Yes, Don made great use of testers, and accepted and incorporated many of their suggestions, but the relationship was distinctly asymmetric with Don having the final say on any and every issue. It has even been suggested (but I have no evidence to support it) that TEX's somewhat arcane macro language is there *because* Don was keen to incorporate his ideas in a large and significant piece of software because the Computer Science community in general had not received these ideas with the enthusiasm which he thought they deserved. Whether this is true or not I have no way of knowing, but there can be little disagreement that TEX *is* idiosyncratic in many ways, and it is very hard to imagine that it would in any way resemble the program we know and love today if Dijkstra, Wirth, Hoare *et al.* had all had equal say in its design.

Now, it is fairly safe to assume that the majority of the audience who read this interview will agree that TEX is a *very* significant achievement, and that despite its idiosyncracies they enjoy using it. They might be hard-pushed to identify another piece of software which attracts quite such a cult following, although Linux cannot be far behind. But the Linux kernel, too, was created in a manner not entirely dissimilar to TEX: it is the work of one man, Linus Torvalds, although in his case there was a pre-existing model (Unix) on which his work was based. Even the World-Wide Web, use of which probably occupies more computer resources than any other computer-related activity today, was "invented" by one man, Tim Berners-Lee.

Of course, in each case cited, the statement "X was invented by Y" is not meant to suggest that Y worked in total isolation, shutting himself off from all human contact until the task was complete. Of course, each project was *influenced* by what had gone before, and (certainly in the case of TEX) by what was about to come, since it is safe to assume that Knuth had more than a passing familiarity with the literature. Nonetheless, all three works listed are essentially each the work of one man.

It is therefore surely relevant to ask whether “universal consensus” is an appropriate model for further development of a TeX derivative. If we look at what is undoubtedly the most successful example of TeX-related development extant (the LaTeX 2_ε/LaTeX 3 project), then what categorises it is anything *but* universal consensus. LaTeX (originally itself the work of one man, Leslie Lamport) is today maintained and developed by a small team (I would guess about eight individuals) with strong leadership and shared precepts and principles. The LaTeX team itself decides what should be done, which areas to prioritise and so on, and then works to achieve its goals. Once stable, the results are released to the TeX world at large. What I would suggest is noteworthy about this model is (a) that it works, and (b) there is no perceivable under-current of muttering “but why didn’t/ don’t they do X?”.

Now compare and contrast this approach with a closely-related field: the development of the web-based styling language CSS. Here we have a W3C working group composed of “members of member organisations”. At the last count, there were 13 distinct organisations (each of which might be represented by one or more individuals) plus “W3C Invited Experts” (see `http://www.w3.org/2004/08/invexp.html` to learn more about what one of those is!). The Working Group are answerable to no-one (save possibly the W3C itself), and it is they and they alone who decide what does, and what does not, go into CSS. Once a preliminary decision has been made, non-members are invited to comment. It is quite clear from watching the lists concerned that comments from some individuals are afforded considerably more respect than comments from others. Some comments eventually lead to change, some are simply noted, and some are virtually passed over as if never made. What *is* intriguing is to see that certain perspectives are very entrenched amongst the members of the working group, and that any suggestion which conflicts with these entrenched views is unlikely to be afforded more than cursory attention, no matter how many times the suggestion be made (and no matter by how many people). One brief example will suffice. TeX users are very familiar with TeX’s macro facilities: if one wants a command (say) `\boustrophedon` and no such command already exists either as a TeX primitive or as a macro pre-defined in a standard package such as LaTeX, one is at liberty to define the command for oneself. Whether one could *usefully* define `\boustrophedon` within TeX is not the point: one can define the command and then (attempt to) implement it through the medium of pre-existing commands and parameters. Compare and contrast this with CSS. In CSS, the vocabulary and syntax are defined by the W3C working group. There are no facilities (current or planned) to allow the vocabulary or syntax to be extended by a user to meet his or her specific needs. In essence, the W3C WG says “we know best: if we don’t deem it worthy of implementation, then it will not be implemented, and you will be unable to implement it for yourself because we will give you no tools within CSS so to do”. No matter how many times someone points out the usefulness of such functionality, the response remains the same: “What you are trying to suggest is best tackled with an authoring tool: this tool can then allow you to define whatever additional syntactic sugar you like, and it — and only it — will then be responsible for converting your syntactic sugar to standard CSS”. And no matter how many times the plaintiff points out that, in the real world, CSS is not written using “an authoring tool”, but rather is laboriously hand-crafted and carefully tweaked so as to produce (approximately) the same effect in all mainstream browsers, the W3C CSS WG position remains unchanged and unchanging.

So which of these models is better? The explicit “we know best” of the LaTeX 3 team, or the covert “we know best” of the W3C? The LaTeX team do not claim to need consensus before release: they do what they think best, and the rest of us live with it. The W3C, on the other hand, claim to be open to suggestions and comment, but many who have made such suggestions and comments are unconvinced that the W3C pay little more than lip-service to the idea. For myself, I prefer the *idea* of the W3C approach but the *reality* of the LaTeX: if the W3C CSS WG were less entrenched, more open to suggestions, then who knows what the outcome would be, and whether it would be better or worse than CSS as we know it today. *Quot homines* ("So many men") said the Roman dramatist Terence, about 2200 years ago. Had he been alive today, he might instead have said "Too many men, too many opinionsv".

DW: I'd like to hear a little about your personal life, if you don't mind.

PT: Not at all. I'm fifty eight, married, live in a maisonette (downstairs single-storey flat with private front and back door and gardens) overlooking allotments, have a dog "Cleo", a cat "Oscar", a pond in the garden for frogs, scythe my lawn rather than using a conventional mower, had a horse "Jingo" for eighteen years who died earlier this year at the age of 27, enjoy good food and wine (the latter is currently proscribed as my liver has issued a final public warning following anæsthesia late last year), and cycle for pleasure (lightweight, not mountain bike) when the weather is good. I spend far more of my time than I should sitting in front of a 19 inch monitor, have a steerable satellite dish (and some fixed dishes) in the back garden hooked up to both analogue and digital satellite receivers which in turn feed into my PC (used as a PVR — "Personal Video Recorder") and a television, recently invested in a hardware DVD recorder to augment my PC-based DVD burning facilities, and am currently (for this week and the next two) recording and burning to DVD the *Giro d'Italia* (a 3-week cycle race which takes place each year in Italy). My wife Lệ Khanh and I are currently trying to redecorate our home, which we took over from my father when he had to move into permanent residential care at the age of 91. The house has barely had anything done to it for the last forty years, so we have rather a lot to do, but progress is not bad and I finished laying the flooring in the hall while Khanh was away in America at a wedding (she returns this afternoon). I enjoy all forms of speed (fast horses, fast cars, fast motorcyles), drive a SAAB 9000 2.3 turbo and ride a Suzuki GSX-R1100. Khanh & I enjoy playing table-tennis when we can (we have no room for a table at home), and both of us enjoy travelling (we're off to Wales on Friday and to China in August; Khanh will later travel to her family home in Đá Nẵng (Việt Nam) in time for the lunar New year and the third anniversary of the death of her father Âu Duong Thịnh Hoài, where I may be able to join her for a part of her visit.

DW: Is there anything you think I should have asked you and did not? If so, please tell me about that.

PT: Mistakes. What mistakes have you made in your life, what mistakes do you believe others have made, and what (if anything) could now be done to redress these?

PT: And now, having asked myself the questions, I'll try to answer them . . . .

Yes, I've made mistakes. Thousands of them (probably more). I am (my wife's words) "confrontational". Fair comment. As an experiment, I asked Khanh to comment on a message I was proposing to send on a TeX-related matter: she did so, I changed the wording, and I was AMAZED. The first reply I received was supportive, none were abrasive, and the idea I was trying to get across was accepted without dispute (probably for the first time in my life!). So that's probably what characterises the majority of the mistakes I've made, at least in TeX-related matters. There was another one, however, the significance of which really didn't hit me until I was asked to take part in this interview. As a "TEX Implementor" (albeit minor, unlike Messrs Kellerman & Smith, Mattes, Popineau, Esser, Carnes, etc.), I was used to defining my own TEX structure. I knew where everything fitted, and the last thing I wanted was some d@mn committee trying to foist "The TEX Directory Structure" on me: what did *they* know about how my TEX system (was/ should be) organised? This remained my (entrenched, see above) position all the while I continued to use VAX/VMS as my TEX platform (which I did for as long as I possibly could). Finally I was forced to migrate to a PC. Even then I still knew best. ArborText's DVILASER/PS (as used on my VAX/VMS systems) was still the only possible way of generating PostScript from DVI (which meant I had to write my own `\PostScript` macro, to generate the right specials for DVILASER/PS) and my own PS-Fonts (since I was using ArborText's naming conventions, not Karl Berry's). I learned a greal deal by so doing (of course); I had orthogonal PostScript font selection long before PSNFSS saw the light of day, but it was of little use to anyone else since it was predicated on the use of ArborText names rather than Berry. I don't know how long it took me to finally try to install TEX Live (TEX-Live, as I insist on calling it, just to annoy Sebastian), but eventually I did. It didn't work, of course, because I took advantage of the installation option to change the default layout: Fabrice tried to help, but in the end even he had to admit that although the *option* was there, one really shouldn't try to use it, and certainly not try to use it such that one TEX-MF tree was nested under another.... (So what's *wrong* with `/TeXMF/Local/` rather than `/TeXMF-Local/`, I still want to know?!) But here I am today, using TEX Live (almost untweaked, although I still can't live with it *exactly* "out of the box"), using Karl Berry's font names (but I still don't understand them), using (G@d forgive me) pdfLATEX when I need `hyperref` and all the other nice goodies that Sebastian *et al.* have provided for the LATEX world but not for those of us that prefer Real-TEX [tm]. In short, I'm reformed but unrepentant. Plain TEX *is* better than LATEX, simply because it's possible to understand every tiny detail of what is happening inside, and to change it to make it do what *you* want it to do rather than what Messrs Lamport, Mittelbach *et al.* think it *ought* to do. What else? Well, I still hate this d@mn "open source/free-means-libre" ethos. I see nothing wrong with proprietary software, use and enjoy using Windows XP, believe that an author has every right to make his software freely available *without* waiving his right to be the only person allowed to change it, and so on. Some of the packages that I've written that *might* conceivably be of use to somebody will never appear on TEX Live because I won't play the licence game. I have better things to do with my time than to waste it adding some meaningless prose to a package simply to permit it to appear on TEX Live or in the "free" (means "libre") branch of CTAN.

[Note added retrospectively when the author briefly emerged from a time-trip in 2025: "OK, I was wrong. LATEX is the one true TEX macro package and Lamport is its prophet. All software should be open-source and freely editable and re-distributable by anyone, no matter how ill-informed or ill-intended. Licences are a Good Thing [tm] and if any of my TEX packages were likely to be of the slightest interest to anyone, I'd gladly add one and personally ask Saint Richard to approve the wording. Unfortunately they aren't of the least interest to anyone, so I won't need to...."]

And what of the other side of the question? "What mistakes do you believe others have made". Well, if the above wasn't heretical enough to cause apoplexy in one or two readers, the next bit will be. I think Knuth made a mistake (several, in fact) in *The TEXbook*.

The mistakes I refer to are all examples, where he happily intermixes what I perceive as the three distinct rôles of TEX in a single fragment of source code. Let me try to explain. Consider the following three lines of [LA]TEX. (I put the "LA" in brackets because the example could just as easily be TEX as LATEX: it is one of the Great Errors that many TEX users believe that only LATEX can handle constructs such as `\begin{whatever}` ... `\end{whatever}`.) I'll return to that point later, maybe).

```
\begin {abstract} ... \end {abstract}
\vskip \baselineskip
\def \firsttoken #1{\firstofmany #1\sentinel}
```

I will argue that each of these is *fundamentally* different from the other two.

The first is markup, pure and simple. It introduces, and then terminates, an abstract. It says nothing whatsoever about what an abstract is, or how it is to be typeset. In short, it serves only to MARK UP a stretch of text. Something else (we know not yet what) must interpret this markup and turn it into (presumably) some typeset copy, although there is absolutely no reason why it might not instead produce the text as nicely read speech, through the medium of a speech synthesiser or similar.

The second is formatting. It causes one additional unit of vertical white space to be contributed to the current page (or box). It knows nothing about what precedes it, nothing about what follows it, and nothing about how big (or how small) `\baselineskip` might be.

The third is (a pre-requisite to) analysis. It defines a macro, `\firsttoken`, that — through the medium of an adjunct macro `\firstofmany` — expands to yield the first token of whatever token list was passed as parameter. It knows nothing, cares nothing, about the context in which it will be used; it simply sits there, waiting to be asked to do its job.

The problem is, in many examples in *The TEXbook*, Don mixes two or even all three rôles in a single example. As a result, virtually every TEX user who first learns TEX through the medium of *The TEXbook* sees nothing wrong in interlarding his or her prose with `\bigskip`, `\bf`, or any of the other "syntactic sugar" so kindly provided by Don in the Plain format (or even with "inappropriate" TEX primitives). Once learned, habits of this sort are *very* hard to unlearn. The HTML/CSS world has already learned this lesson, and is desperately trying to remove all formatting tags from XHTML, replacing them by pure markup tags the rendering of which is left solely to the browser and/or CSS. In the TEX world, there is little evidence of a general acceptance that the problem exists (or even that it is a problem!). LATEX *tries* to inculcate good habits, but it is still possible to write the most awful mixture of markup, formatting and analysis without ever leaving the nanny-like world that one enters by embracing LATEX. How can this be addressed?

I believe that what is needed is a *radical* re-think of TEX-related markup and programming. No "user" should ever have to use backslashes or braces (at least, not with their conventional TEX meaning). By eliminating what are strictly "control words" from the users' vocabulary, space-gobbling will cease to be a problem. (How many books have *you* seen, typeset in LATEX, in which one word is accidentally elided with the next because the first word occurred so frequently that the author defined it as a macro (control-word), and then forgot on at least one occasion that the macro would gobble up the following space?) If we do away with backslashes and braces, with what should they be replaced? I believe that the answer is already clear: the number of people writing web pages is probably at least 100 times as many as those writing TEX documents. Although many of those web authors will be using some sort of authoring package (Dreamweaver, or whatever) a substantial majority will–at least from time to time — be writing in "pure" HTML, in which case they are writing things such as `<title>My First Web Page</title>` without significant difficulty. If they can do it, so can we! In fact, "they" and "we" is a pretty big category error: there can hardly be any one of "we" who is not also a "they".

Thus (almost all of us) are already familiar with an alternative markup paradigm: "all" we need to do is to make it accessible to the TEX world at large.

OK, so I believe that HTML-like markup should be adopted *as the norm* for the TEX world. I say "HTML-like" because the last thing that I want to do is to throw the baby out with the bathwater. TEX is a zillion times more powerful than HTML because *the user is not constrained as to vocabulary*. Thus if I need `\latinprose`, and you need `\greekverse`, each of us (in TEX) is free to define the command and then to implement it. Not so, in HTML, and hardly so in XHTML, despite the misleading "X" in the latter's name. Thus I argue that what we need is an *extensible* markup language, superficially similar in syntax to HTML but with the essential ability to be able to define (and implement) additional tags (`<latinverse>` & `</latinverse>`, for example). Where this gets (more than a little) complex is in determining which tag can occur in which context(s). In HTML, all is simple: the grammar is pre-defined (in a DTD) and the rules may be derived therefrom. An HTML parser can then tell straight away whether or not a document is *valid* HTML. In the system I am advocating, a user would be able to add tags to the language in an (almost) *ad hoc* manner: since writing a DTD is a distinctly non-trivial task, it is clearly *much* easier to define and implement a new tag than it is to modify the rule-set to indicate in exactly which contexts it is (or is not) permitted. More work needed here: I am floating ideas, not proposing a fully worked-through solution.

With the document marked up using HTML-like notation, we next need a means to describe how that document is to appear on the printed page (or on the screen, or through a speech synthesiser). Again, I see no reason not to learn from (and copy where appropriate) the HTML world. The appearance of HTML documents is partially implicit (rules which are assumed to exist in every browser) and partially explicit (a user may augment or modify these rules using Cascading Style Sheets — CSS). The syntax of CSS is very straight-forward, and should be easily understandable by TEXies:

```
    LATINVERSE font-style:italic; margin-left:  1em
```

is pretty self-explanatory. I therefore believe that the *formatting* aspect should be expressed in a CSS-like syntax, but given the hard-learned lessons of TEX, that the vocabulary should be extensible rather than hard-wired (see preceding para. for caveats concerning extensibility).

Finally we need analysis, and here (of course) TEX comes into its own. Whilst more than a little idiosyncratic, TEX has proved itself capable of being used as a parser, calculator, constraint-solver, and everything else needed (including, of course, a typesetting engine *par excellence*) and thus is the perfect engine through which to process the HTML-like user markup and the CSS-like "document designer" markup.

Is this a pipe-dream, or reality? I sincerely hope that it is the latter. Backslashes and braces, space-gobbling and all the other minutiæ that characterise TEX may well have been acceptable for user documents in 1978, and even in 1982. By 1990, they were beginning to look a little passée. In 2005, they are surely *well* past their sell-by date. It is time to stand back, to take a good look at the program that we all love (and occasionally love to hate), to identify its strengths and to capitalise on them; but at the same time to identify its weaknesses, and to have the courage and the wisdom to excise them, leaving a lean, mean, typesetting system layered on which are a sane markup language and another (but clearly different) language in which formatting concepts can be clearly expressed.

DW: Wow! That gives me a lot to think about. I hope you will think it out more fully and write it up at some point.

Thank you, Phil, for taking the time to participate in this interview. The history you can recount and the perspective you bring are fascinating.

PT: Thank you!