

Peter Flynn

Peter has worked on T_EX interfaces at many levels for many years, including SGML, XML, dual web and print presentations, within the context of excellent typography. He is also a past board member of TUG.

[Interview completed 19 September 2006.]



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

Peter Flynn, interviewee: My infancy and early childhood was spent moving, although I remember little detail. At that time, if your company said ‘move’, you moved, and my father’s job took him all over southern England. In 1961 the pace slowed down and we moved from rural eastern England to the Midlands for five years. Then my father was moved again, this time ‘home’ to Ireland (although I’d been born in England, we still had strong family ties to Cork) but rather than disrupt my schooling I stayed on as a boarder, returning home in the vacations.

The school was to have a major effect on my life, as it was there I discovered the big box of little bits of metal with letters on the ends: the remnants of the school printing press, which nearly got me expelled for printing a scurrilous leaflet about an unpopular teacher! Nevertheless, I eventually left school for Exeter University to read languages, and filled in as a reporter on the student newspaper. But I discovered that ‘languages’ in a British university meant 99 percent literature and 1 percent language. I loved the reading, but I wasn’t a lit. critter, and I only stuck it for a year.

The next year was spent earning a living back home, first selling paint, and then delivering laundry, until those little letters called me, and I ended up doing a business degree at the London College of Printing. 1976 was not a good year to graduate: Maggie Thatcher was crushing the print unions, and Fleet Street was decamping *en masse* to the docklands. I was given the chance to do a taught MA in computerised planning at what was then the Central London Poly, and I emerged in 1978 working for the Printing and Publishing Industry Training Board, trying to browbeat a reluctant industry into training people in the ‘new technology’.

When Mrs Thatcher summarily wound up all the ITBs and put us onto the street, I fled printing for a few years to sell dialup computing services to an equally reluctant financial clientele who spent most of their time in City wine bars. I had got married in the meantime, and for some years had been quietly looking for a job in Ireland. The university in Cork was expanding its computing support, and I joined them in 1984. Everyone said I was mad to work in a state with over 50 percent income tax, but what clinched it was that instead of a 2-hour commute which left me not seeing my baby daughter from Sunday night until the following Saturday morning, I was into work in seven minutes. It’s been an uneven ride: I thought I might get a few years off in the groves of academe, but the pressure in a university can actually be far worse than in business. However, it has given me the chance to work at the edge of some new technologies, and to pass that knowledge on to another generation, which is always rewarding.

DW: When and how did you first get involved with \TeX and friends?

PF: In the early 80s, when I was still at the PPITB, we were doing a major survey of the effect of new technologies on employment levels. We all knew the depressing news: that direct-entry systems would eventually bypass the hot-metal compositor, but the unions weren't having any of that alarmist rubbish (what was good enough for Gutenberg was good enough for them), and all the systems available required specialist dedicated computers.

A colleague came back from a visit to Stanford with news of a professor who had written a typesetting program that would run on a normal computer. We laughed. But as it happened, we had one: a DEC 20/20 leased from ADP, and with a little goodwill and some transatlantic calls we got a tape. A friend who knew more Pascal than I did spent some frustrating nights getting it to work, sitting in my loft at a TI-700 running up a huge dialup phone bill. We finally showed it to a meeting of print technologists from PIRA and the NGA. The union guys freaked: "You can't let people see that", was a typical comment.

When I moved to United Information Services (part of United Telecom of Kansas, "the second-largest non-Bell telephone company", as they liked to style themselves) my life went crazy. I was supporting a remote database from London to the money-makers, statistics processing from a basement of DEC-10s in Pittsburgh to drug companies, and stress-analysis packages from a Cray in Kansas to engineers. I tried to interest my employer in desktop typesetting, but the unions wouldn't let any printing company print stuff that didn't have an NGA sticker on it to prove it had been typeset by their members. The only cute jobs were reprogramming a daisy-wheel printer to form real typeface letters by steering the period character to print high-density dots, and the discovery that a little work on the manual for P-Stat, which we got as a Fortran print-file on tape, would make it print near book quality on the new-fangled laser-beam printer in the Xerox office down the street.

Oddly, it wasn't me that introduced \TeX to UCC. We had a VAX, but no graphical displays or printers, and I was tied up with debugging the crystallographers' Fortran. The credit goes to a colleague, the late and much-missed Mike Gordon, who bought some copies of PCTeX for an admin application. In 1986 I was in hospital for a thyroid operation, and Mike smuggled in my IBM twin-floppy luggable with a modem card and cable so that I could get my email, chat on RELAY, and possibly even do some work. He dumped the \TeX disks on my bed and said "you're the printer, see if you can make sense of this." I spent the next few days playing with macros and writing a short story which proved beyond doubt that I can't write fiction to save my life.

It wasn't until 1988 that we had enough users to justify me going to a TUG meeting to find out what was going on. Exeter had changed a lot since I had been there in 1973, but $\text{L}^{\text{A}}\TeX$ was a revelation to a dyed-in-the-wool plain \TeX hacker. It took me another year to make the connection with SGML, which I had to support for an EEC-funded project, for which we got a converter to $\text{L}^{\text{A}}\TeX$ called Daphne. This came from the DFN in Berlin⁵, and it sticks in my mind because I signed the agreement while I was there at a RARE WG3⁶

⁵Deutsches Forschungsnetz — Germany's National Research and Education Network.

⁶RARE was the Réseaux Associés pour la Recherche Européenne, the Association of European Research Networks, who were contracted to DGXIII (Directorate-General 13: Information Technology) of the European Commission to report on the future direction of networking in Europe. Working Group] 3 (Directories and Information Services), of which I was Secretary towards the end of its life, produced the blueprint for what would have become an X.500 directory service and a network of information servers. But working for a bureaucracy that size meant technology overtook us before they could react, and those of us who preferred the

meeting the day the Wall came down, and we suspended discussions to go and hack off a chunk to keep.

I then missed the 10th anniversary meeting in Stanford, but I had offered Cork as the location for a combined TUG and Euro \TeX meeting in 1990, which turned out to be fairly pivotal, not just for the organisation of the font files, but for the organisation of TUG, and for the use of \TeX in the university. This too has been a rocky road, and the effects on \TeX of the use and abuse of synchronous typographic interfaces is something I'm still exploring in my own research.

DW: No doubt many long-time \TeX people know about the impact of the Cork 1990 meeting, but having come to \TeX more recently, I don't. Please fill me in on what happened at Cork that addressed font files and affected TUG and your university.

PF: I wasn't directly involved, as I wasn't on the Font Committee, and of course I had my hands full organising the meeting, but the major change was the agreement on a 256-character font file layout. Before this, \TeX fonts had a 128-character font file, and diacritics were all composed from floating accents. The 'Cork encoding', which includes essentially all the letters needed to typeset in Western European languages, allowed \TeX fonts to have a direct representation of the characters in what is now the ISO-8859-1 character repertoire, where the diacritics are precomposed, and many additional characters included. Computer Modern is represented in this format by the EC fonts which we now use for the T1 encoding. The move brought \TeX into line with what is now conventional practice, and started to pave the way for the full use of Unicode which we now see in $X_{\text{Y}}\TeX$. The TUG organisational issue was separate, and reflected a growing dissatisfaction among non-US members at the perceived US-centric approach. Just holding the first TUG meeting outside North America was a major step, and it also had the effect of boosting the profile of \TeX within my university.

DW: You said, "It took me another year to make the connection with SGML"; please elaborate on what you mean by that.

PF: I'd been familiar with markup for a long time, back as far as RUNOFF (an early text processor), so when I encountered \TeX , it was that much easier to assimilate. When some of my users asked me to help them with SGML for project reports, I could see how it worked, and the concept of a file format that could be independently verified was attractive, but I couldn't see how to get it converted to \TeX for printing. It was at the meeting in Berlin that Gerti Foest, the DFN member of RARE WG3, told me they had a program that did exactly that—and it suddenly dawned on me that other people had had this requirement and had solved it. That meant it was possible to store information in a format that didn't predicate its appearance (and could therefore be maintained by the author without damage), but which could be converted to a typesettable format as and when required. I haven't authored anything major directly in \LaTeX ever since: it was all in SGML, converted to \LaTeX for printing (and now XML, converted to both \LaTeX and HTML).

DW: That approach sounds interesting. Will you give me a few more details of how you compose in XML (e.g., do you use an XML editor), what programs you use for the conversions to \LaTeX or HTML, and how you control the chapter, page, etc., design of the \LaTeX document (if XML allows these things to be specified)?

PF: I use Emacs because I'm used to it and it does most of what I want. I can't imagine trying to write XML without an XML editor: a bit like trying to compose a piano sonata

TCP/IP path to the OSI path voted with our feet.

without a piano. I've used dozens of SGML and XML editors, especially since I started research in that field, but although it might be aesthetically more satisfactory to use a synchronous typographical editor, the current interfaces just get in the way. They're not designed for authoring, and require far too much foreknowledge of the technology to be useful for the non-expert.

It's interesting that OpenOffice, AbiWord, WordPerfect, and Word can all save their own files in XML, but pretty much all they do is save an XML description of the visual arrangement of your document: this offers no advantage over a proprietary binary format except that it can (with difficulty) be post-processed into something more meaningful. So long as their interface makes no attempt silently to capture meaning from the author, meaningful, re-usable XML will remain out of our grasp.

To convert to L^AT_EX I used to use Omnimark, which was available free for a while in the SGML days; but once XML got off the ground I started using XSLT with Mike Kay's Saxon processor for standalone work, and Apache's Cocoon for interactive online documents. XML documents traditionally hold no information about format or layout, only structure and meaning: appearance is the responsibility of the stylesheet (an XSLT program); and of course once it's converted to L^AT_EX, you have the full range of styling and formatting available.

Put simply, it means that

```
<chapter id="intro"><title>Introduction</title>
```

can trivially be output both as

```
\chapter{Introduction}\label{intro}
```

and as

```
<h2><a name="intro">Introduction</a></h2>
```

so you get the best of both worlds: proper control over document structure while editing, and proper control over formatting when you want output.

DW: I see from the TUG web site (<http://www.tug.org/board.html>) that you were on the TUG board from 1992–1996 and editor of TTN. Please tell give me your perspective on TUG and its place in the T_EX world then versus now. Also, I don't know what TTN was: please fill me in on what it was and what happened to it.

PF: By 1992 it was clear that TUG was not the only T_EX user group, and that the falling membership was not just due to local user groups: the increasing spread of the Internet meant that individuals and organisations were questioning their need to remain members; and the lack of coordination between the TUG office and the activists and users was making T_EX look less relevant. I wanted to try and improve things, and I guess I also felt that I had got a lot from T_EX and wanted to put something back. There was a certain amount of political infighting, as always in volunteer organisations, but a huge amount of goodwill and effort went into redirecting TUG into becoming more responsive and more attuned to the changing needs of the users. TUG meetings used to be held in big, expensive hotels, with strong participation from well-funded companies and universities. As this all fell away, the organisation needed to change tack, and at the same time cope with technological change and competition from synchronous typographical editing software. Sadly, a new regime in my own institution didn't see my participation in TUG as useful, so I had to resign at short notice when the funding rug was pulled from under my feet. I think TUG is in much better shape now, organisationally, but it did take over a decade to get there.

TTN was "T_EX and TUG News", a news-sheet separate from *TUGboat* which Christina Thiele capably edited for a long time. I offered to take over from her when she became

President, but I was unable to give it the attention it deserved, and it was subsumed into *TUGboat*.

DW: What do you think about the roles TUG and the various local user groups play today and what they should be doing if they are to be useful in the future?

PF: I've been out of the inter-group politics for a long while now, so I'm not up to speed on their interrelationships. TUG clearly has a custodial role as the progenitor of T_EX user groups, as well as the one representing the interests of North American users and those users who either don't have a local user group or who just prefer to join TUG. But T_EX is so widespread that TUG can't possibly service the huge diversity of interests in other cultures alone, so the local user groups are the primary focal points for T_EX in their domain. LUGs can do things like organise local meetings, write culture-specific documentation, and represent the views of their members to TUG more effectively than individuals could do alone. Some of the LUGs are so active that they can take on major roles in driving the technology of T_EX forward.

For the future, I think the existing cooperation is an excellent model, but we do still need to publicise T_EX more widely, and crush some of the myths that have grown up around it. The hegemony of the word processor has meant that even the original core of T_EX— universities, industrial research, and the computer industry— is now largely unaware of T_EX (or only marginally aware), and computer users outside the core are wholly unaware of it.

These Unawares are prey to the FUD that surrounds the word processor, which is supported by millions of dollars of marketing. We can't compete with that level of effort, so the only way we can ensure T_EX's survival is by making sure that *when* new users encounter T_EX, they get the very best and most accurate information possible. The `comp.text.tex` newsgroup, TUG's web site, *TUGboat*, CTAN, the FAQ, the LUG web sites and their journals, and all the documentation (paper and web) need to find a way to ensure that the user can get useful and usable information. Right now, the new user is often presented with too many apparently conflicting options and unnecessary detail. We *need* that Marketing team with the tin of polish and the soft cloth to stop the 'basement enthusiast' image putting new users off (while keeping the basement enthusiasts, of course, including me!).

DW: When I surf around on the web looking for your name, I find many interesting places, for example:

- <http://silmaril.ie/>
- <http://imbolc.ucc.ie/~pflynn/>
- <http://xml.coverpages.org/faqs.html>
- <http://www.tug.org/tug99/program/node20.html>
- <http://www.ctan.org/tex-archive/info/beginlatex/html/>
- a number of instances of your Typographers' Inn column in *TUGboat*, e.g., <http://www.tug.org/TUGboat/Articles/tb25-2/tb81inn.pdf>
- etc.

You are a very active fellow. Is there an overarching goal or theme or your activities— is there some model you are trying to advance in the world? Or are you just interested in helping people in a variety of areas that you find interesting?

PF: Some of it is just stuff I got involved in which happens to have left a trace. One of the members of that RARE Working Group was Tim Berners-Lee, who demonstrated

his new hypertext project to us in Zürich in 1990. In Cork we were just starting a major new project to publish transcriptions of Irish historical and literary documents, which would all be in SGML, and this seemed like an obvious way to put them on the Internet. I downloaded Tim's *httpd* software, and UCC became the ninth web server in the world. In fact the server you mention (*imbo1c.ucc.ie*) is that very machine, now the world's longest-serving web server still in use. It's a Sun Sparc IPX running SunOS 4.1.3, and in over 15 years it has never crashed once, and only been offline three times (twice for an office move and once to add more memory).

The effect of all this was that I unwittingly became Ireland's first webmaster, and also the first person on the web to break a link (I renamed a file without thinking of the fact that it was linked from Tim's machine at CERN, and I suddenly started getting email asking what I'd done to it). The historical and literary project also got me far deeper into specialist markup than I had expected, and that brought me into the design of HTML and eventually of XML. The consultancy, Silmaril, had been a loose association of myself and some colleagues, but by this time the others had moved away or changed business, so now it concentrates on XML and L^AT_EX.

So there isn't any overall game plan, but there are lots of areas I am interested in, like typography (which reminds me I owe *TUGboat* another column), and I guess some of it is for helping people. I can remember being a clueless newbie myself (and most of my meanderings are probably still sitting on some server somewhere), and being deeply grateful to the people who gave their time and expertise to show me what I had done wrong so that I would learn to do it right. It's probably not entirely altruistic: the better we can document T_EX, the less time we have to spend constantly explaining simple things, so the more time we can spend explaining more difficult things, or finding out how to do things better.

This has culminated in the research I'm doing for a late-life PhD. Not to put too fine a point upon it, the editors which newcomers use for XML and L^AT_EX—both plaintext and typographical—suck lumps of lead. I'm not talking about experienced users (Emacs does me just fine) but about the next generation. There are dozens of unnecessary interface barriers to the editing of structured documents, and they are all usability errors which are forcing people back to using word processors and deterring the rest even from trying L^AT_EX or XML. I presented a good chunk of my argument at the Extreme Markup conference in Montréal this summer (<http://epu.ucc.ie/articles/extreme06>) and I've been delighted with the support I have had from users. Oddly, the manufacturers and vendors don't seem interested (with one exception).

DW: How do you think that T_EX has to continue to evolve to remain relevant in a world increasingly focused on XML, MS Word, and big vendor driven standards for fonts (or what should T_EX's place be in the larger world)? Which of things like the Web2C version of the basic T_EX typesetting engine, pdfT_EX, X_YT_EX, ConT_EXt, and hoped-for developments such as L^AT_EX 3 and the often desired redo of T_EX's capabilities integrated with a non-macro-based programming language seem most relevant to you for T_EX's continued usefulness, or do you believe other changes or expansions of T_EX are more important for the future?

PF: The fact that T_EX is still here after nearly 30 years is proof enough of its usefulness and reliability, but what I said earlier about fighting the FUD continues to be true. We simply don't shout loud enough about what we've got.

XML is no threat: on the contrary it's a whole new field of opportunity, especially while the XML-to-PDF language (XSL, as distinct from the XML-to-anything-else language,

XSLT) continues to be expensive and time-consuming. There was an article a long time ago called something like ‘PostScript versus T_EX’ as if PostScript were some kind of threat!

Big vendor software like Microsoft Word, and its competitors both commercial and free, online and offline, will always look more superficially attractive because they work on the basis that making it look pretty means that it’s correct. Microsoft in particular is very good at selling this line, so much so that I suspect the majority of the population now believe it to be true. There is no point in trying to peddle to the general population the benefits of open source software, data preservability, the reusability of information, and the accessibility of document structure, because they simply aren’t relevant to the kinds of documents they write. There is equally no point in trying to sell these aspects of T_EX systems to business, because most businesses aren’t looking that far ahead, and it’s cheaper to pay the Microsoft licenses than to re-train everyone to use L^AT_EX.

Individual divisions *within* businesses, however, sometimes *do* have people looking more than two quarters ahead, and they are part of our market, as are people sick and tired of losing time and data trying to make a word processor do stuff it wasn’t intended to do. And we still operate a *laissez-faire* attitude to universities, our biggest training-ground by far, where we should be in there handing out T_EX Live CDs and leaflets describing how L^AT_EX can improve output and save time. We need to tackle this one at the roots or we’ll lose the upcoming generation for ever.

By the same token we can’t compete with the big boys in funding T_EX people to sit on international committees for fonts or anything else. It costs \$6,350 for a non-profit based in the USA to join the W3C, for example, before getting onto a committee and before paying the travel to get to the meetings. But the various T_EX user groups worldwide include among their members from time to time a number of the very experts whose opinion might be consulted by these committees. What we *could* do is offer these individuals some kind of incentive — it may not even have to be financial — to ensure that T_EX’s voice gets heard. In fact this probably happens anyway if the relevant gureaux⁷ are T_EX users, but it would be nice to *know*.

This has a corollary: we have to listen to them even if the direction things are going in isn’t palatable. If the world takes on a new technology that obsoletes something T_EX does, we need to change. We did this with PostScript; we did it again with PDF; we’re doing it with Unicode. We’ll have to do it with fonts, too.

What we do with our engines is a slightly different matter. My guess is that most people are still using the ‘standard’ T_EX binaries to produce DVI, even if they end up converting it to PostScript or PDF; and that relatively few so far are using pdf_ε-T_EX as their engine. I may be wrong: we need to start collecting some of this information, and I hope the proposed T_EX Counter will help here. Other developments, including ConT_EXt and X_εT_EX are going to be just as important, and at this stage it’s difficult to pin down one way of doing things and say ‘that’s the only route for us’, except that producing PDF seems to me to be the right direction, and coping silently and sensibly with multiple font formats and arbitrary Unicode input is going to be a huge advance (but we’re going to need an editor to go with it all!).

T_EX in its original form is architecturally incapable of being integrated into the synchronous typographic interface. Various reimplementations of it have nevertheless made it possible: for example Textures, LyX, Scientific Word, and the reuse of large chunks of the T_EX engine in assorted word processors and DTP systems as well as in ‘native’ systems like Instant Preview and preview-latex. Whether or not we reimplement T_EX in a

⁷A colleague of mine recently came up with this plural of ‘guru’.

Next Generation version, we have to understand that the world is moving towards digital seamlessness, and that in a few years, a majority of people will expect everything to ‘just work’ — and if it doesn’t, to drop it on the floor. Right now, a majority of people are still accustomed to incompetently-written operating systems and applications software failing too frequently, but the expectations are rising, and we’re going to have to rise with them. There will always be a need for T_EX Wizards who can work behind the control surface, but up to now, the T_EX community has tended to concentrate on them to the exclusion of everything else (and done a damn fine job in the process, too, of course, but things change). If we want the use of T_EX systems to expand, we’re going to have to crack the interface problem: it’s not going to go away. Alternatively, we return T_EX to the computer scientists with grateful thanks for all the fish, and look for something else.

DW: Thank you, Peter, for participating in this interview. It was wonderful to hear about your background and current projects.