David Carlisle

David Carlisle is a member of the LaTeX team and deeply involved with issues of typesetting and displaying mathematics, including being an Invited Expert on the W3C Math Working Group.
[Interview completed 20 June 2007.]

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

David Carlisle, interviewee: Born 1961 in Southwell, England, and grew up in Mansfield. At University I specialised in Mathematics at Manchester, both undergraduate and postgraduate, finishing up with a PhD in Pure Mathematics in 1985 (typeset on an IBM Golfball typewriter with large brackets drawn in by my wife, who has a steadier hand . . .). There then followed a two year post doctoral position at Cambridge before returning to Manchester in 1987. I worked as a researcher in Manchester in several projects in either Computer Science or Mathematics, until 1997. In 1998 I moved to NAG, a mathematical software company in Oxford, to work initially on OpenMath and MathML, but now more generally on their XML systems. Outside of NAG I'm an editor of the MathML specification, and quite active in the community around the XSLT XML transformation language.

I now live in Souldern, a very small village in Oxfordshire, with Joanna, my wife (who uses TeX more than I do), and Matthew who's 3 and a half and quite computer literate but not, as far as I know, a TeX user.

DW: What does your wife use TeX for?

DC: Prior to having Matthew, Joanna taught mathematics (age range 11–16 mainly) so worksheets of various sorts got typeset with TeX, but even now, letters and stuff, and the occasional poster for some village events gets typeset in TeX I believe.

DW: Does NAG develop the same sorts of products that our local Boston area company MathWorks does (e.g., MATLAB)? And why/how is XML important to math software products?

DC: Probably this isn't the place for a comparative review of commercial software, but in general MATLAB offers an interactive environment in which to solve problems of various sorts. NAG's main products are lower level libraries of mathematical functions which are often embedded into other products rather than being used directly by the end user. However, for those who don't want to program directly in C or Fortran, we do offer interfaces to the libraries from problem solving environments, including Maple and MATLAB.

DW: When and how did you first get involved with TEX?

DC: September 1987, when I returned from Cambridge I took up a position at the Computer Science Department in Manchester which had (at the time) one of the largest Sun networks in Europe. So right from the start I had the benefit of the graphical "Sunview" environment and the `dvitool` previewer which was far ahead of its time (and one of the best dvi viewers I ever used). Actually it's not quite true that I had that environment from the start; when I first got there my machine was still on order, and so I was at a bit of a loose end. I asked some of the other post docs what I should do to orient myself to life in the CS department, and one of them gave me *The TEXbook* to read. I remember I read it right through to the end, before I ever got chance to try out TEX.

The Computer Science department had by then quite a lot of experienced TEX (almost all LATEX) users, so it was a good environment in which to lean TEX, Also I maintained close contacts with colleagues in the Mathematics department who were just starting out with installing personal computers and so I also spent some time helping them get TEX set up. It was while using TEX on these early (512K) PCs that I first started looking at the internals of the LATEX system, developing some locally modified versions that had a slightly more usable startup time. A direct descendant of those efforts can still be found on CTAN as `mylatex.ltx`. Another unusual aspect of the initial environment in which I learnt LATEX was that right from the start I had access to scalable fonts. Most people of that era associated TEX with Computer (or Almost) Modern fonts in bitmap form, but Mario Wolczko had modified LATEX 2.09's `lfonts` file to use scalable fonts so I had (as I recall) `latex` (CM), `pslatex` (Times), `pslatex-n` (New Century), and `pslatex-b` (Bookman). We had good on-screen PostScript preview, with Harlequin scriptworks, and later, Ghostscript.

DW: You are well known for your involvement with the LATEX team. How did you move from learning TEX/LATEX at the university in Manchester to becoming part of the LATEX 2_ε core development team, and what is your role on that team?

DC: I'd posted a few stylesheets to (pre-CTAN) TEX archives and had answered a few (or a lot) of questions on the `comp.text.tex` (I think initially `comp.text`) newsgroup, so I suppose I got noticed somewhere. I did email Frank quite early on to ask if he minded if I distributed my `narray` style (which consisted largely of a complete copy of his array style). He said he didn't mind but that it would probably be better to instead distribute `narray` in a form such that it input (rather than copied) `array.sty`. So that's what I did. In the end of course we abandoned `narray` and merged the two back together (although anyone looking at the internal array package documentation can still see the join). I joined the LATEX 3 team by way of a trick! Frank and Chris mailed me one day out of the blue with the offer of a free trip to Hamburg for the weekend. Checking with Google suggests that would have been 1992. It was the DANTE meeting that launched the NTS project. (They kindly held the last session in English as non-German speakers were present.) The LATEX 3 project then had a meeting over the following two days. I think Frank, Rainer, Chris and Johannes, plus Phil Taylor (who was, I assume, mainly there for NTS) also sat in. Shortly afterwards I obtained a copy of the LATEX 3 kernel as it was at the time, and generally got involved in the development of LATEX 2_ε and its subsequent maintenance. In the end it seems I found that the mechanics of producing mathematics papers was taking more of my time and interest than actually producing the mathematics, and these days I work full time on (mainly mathematical) document markup and production, rather than being a research mathematician.

I'm not as active as I was in the LATEX maintenance, though I'm still on the core team

and get all team emails and any messages sent to the LaTeX bug system, but I have less TeX-related time than I used to have, with changing jobs and family circumstances.

DW: From your biography page on the NAG web site (`http://www.nag.co.uk/about/dcarlisle.asp`), I see that you are much more broadly involved in math typesetting and display (MathML2 Recommendation, co-chair of the World Wide Web Consortium Math Interest Group, editor of a draft update to the ISO entities for characters, an editor of the OpenMath Standard, and a member of the OpenMath Society), much more widely than just in TeX. How do you see TeX playing in the math typesetting and display world as the world continues to evolve?

DC: Yes, actually I should update that page. The W3C re-chartered the Math activity to work on MathML3, the first draft of which was published earlier this year. I stepped down as co-chair of the group, but will again act as co-editor. As for the future, that's always hard to predict. TeX has far more competing products these days, most notably the office suites (Microsoft, OpenOffice.org, etc). Personally I've never felt comfortable authoring in a WYSIWYG environment but I am apparently in a minority with that view. In stark contrast to the situation when I started to use TeX, people who prefer that kind of environment can gain pretty good mathematical typesetting. For example, the math layout rules in Office 2007's math layout are explicitly modelled after the TeX layout rules as documented in Appendix G of *The TeXbook*. (See `http://blogs.msdn.com/murrays/archive/2006/09/13/752206.aspx` where it says, "*The TeXbook* is a user manual that includes a detailed specification for mathematical typography. We have used many of its choices and methodology in creating our solutions, which are appropriately enhanced with the use of OpenType tables and some additional constructs.") This means that even in TeX's core constituency of the academic working in the mathematical sciences, TeX does not have the totally dominant position that it had as an authoring system. On the other hand the rise of XML (including XML output from office suites and database systems) but also DocBook, XHTML, and any number of more specific XML languages means that there is a greater than ever need for a high quality batch oriented typesetting system. (The PDF version of the MathML spec is typeset by pdfLaTeX for example.)

TeX would make an ideal basis for such a system; however, the main problems (when using traditional TeX) relate to Unicode input, and TeX-specific font output encodings. It's not that it's impossible to work around these issues, but managing the mappings between an external Unicode world and an internal 8 (or 7) bit TeX is a black art that always limits TeX acceptance in larger projects. For various reasons Omega never seemed to gain the momentum to take over as the main TeX engine and I'm glad to see XeTeX (which I must admit I haven't had chance to use yet) is gaining real acceptance, being part of the TeX Live distribution, etc. A major reason for using TeX is its portability, so (as was clearly seen with ε-TeX) people are reluctant to use experimental versions of things if they have to be downloaded/installed as there is no guarantee that the person receiving the document will have the appropriate software version. However once a system is installed by default on most TeX systems (even if most TeX users are not using it) then it becomes possible to fairly quickly "move" the community. If LaTeX font packages start working (or working more effectively) if running on a Unicode-based TeX system (whether that be XeTeX or Omega or something else yet to be produced), then the system will rapidly gain acceptance even if a large part of the existing community is English speaking mathematicians who don't immediately see any real benefit from moving to Unicode (just as they didn't really have immediate gain from the disruption of TeX3

moving to 8 bit).

DW: You mentioned (and your author bio in *The LaTeX Companion, Second edition*, also mentions) that you are involved with the XSLT language. What is that and what is your involvement with it?

DC: XSLT is a language for transforming an XML document to something else (another XML document, or HTML, or text, normally). It is probably one of the most widely installed programming languages ever, as most systems have several XSLT engines installed. (Mozilla/Firefox, Internet Explorer, Opera, and Safari each include an XSLT engine, as does the Gnome desktop in Linux, and the Java and .NET frameworks each have XSLT as standard as well.) So many people have an XSLT engine to hand even if they are unaware of it. It recently acquired a new related language, XQuery, which is more targeted at database usage. I have avoided being on the standardisation committees for XSLT, but I'm one of the more active members of the main community forum (`xsl-list`) and probably my activity on `xsl-list` accounts for the reason that I am a rather infrequent poster to `comp.text.tex` these days — I can only handle one high volume list!

DW: XeTeX (increasingly in widespread use) and LuaTeX (moving into the beta demo phase) seem to be creating a good bit of buzz in the TeX world these days, and ConTeXt seems to be increasingly used, perhaps at the expense of LaTeX. How do you feel about activities such as these and the continuing role of LaTeX as the TeX world moves forward?

DC: See above, on XeTeX. I think that a Unicode-aware TeX-like system that can use system installed fonts is an absolute essential for TeX going forward. I'm happy to see that XeTeX seems to be working towards this. LuaTeX I know less about. There have been several attempts in the past to integrate TeX with scripting languages, Perl, Python, etc. Unlike with Unicode and font support where one can say that more or less any progress is a good thing, the success of a language merger is almost all to do with the fine details, so I can't really comment until I've seen it. As for the growth of ConTeXt at the expense of LaTeX that's inevitable as essentially there is nowhere else for ConTeXt users to come from other than people who are, or would be, LaTeX users. I think, though, that LaTeX's more open package-oriented approach will always mean that it's more popular than ConTeXt, which as I understand it has a far more "monolithic" approach to system design, where the core has far more features but it's harder to add to that core. But ConTeXt's core is of course newer than LaTeX and has benefited from that experience, and Hans has made a very nice (and extensively documented) system. There has never been any sense of rivalry between the ConTeXt and LaTeX teams and we had several joint meetings as well as meeting at several more general TeX conferences around the time that LaTeX 2_ε and ConTeXt were being initially developed.

DW: I believe you developed a system called `xmltex`. Why did that project get started, how does it work, and what is its status?

DC: Most of `xmltex` got written in a couple of "weird weekends" so it was never really a long term project, although Sebastian Rahtz stress tested it while using it for his PassiveTeX system that typesets XSL-FO XML, as well as TEI XML and a few other formats. A long time earlier I'd written a small package that tried to typeset HTML (`typehtml` is still on CTAN). That was mainly concerned with typesetting the math component of the ill fated HTML 3.0 draft specification. When Sebastian told me he had been using it as a basis for some experiments in typesetting XML, I was sure that something better could be done; much of `typehtml` is complicated by the need to infer missing markup and "tag soup" that is HTML. The stricter requirements of XML parsing are supposed to make

it easier to write robust parsers. I ended up writing a fairly full XML and XML Namespace parser, with some non-conforming behaviour forced by TEX's normalisation of white space (some of which can not be controlled by TEX macros) and its inability to effectively deal with UTF-16 encoding. `xmltex` does however have table driven encoding support that in principle allows any 8-bit encoding to be used, as well as the full Unicode UTF-8 encoding. Dealing with these Unicode encodings with a classic 8-bit TEX accounts for almost all the complexity of `xmltex`, harder really than parsing XML syntax.

How does it work? Basically you change round the default catcodes so that the XML control characters (< and & mainly) are "active" and defined to macros which look ahead and parse the XML element and attribute syntax. People seem surprised that this is possible, but actually it's easier (modulo character encoding issues) to parse the highly regular XML syntax than something like a LATEX tabular column specification, which also has to be parsed character by character by the TEX macro layer. However, the fact that it is possible doesn't necessarily mean that it's a good idea. Parsing XML with TEX isn't necessarily that fast and more importantly it's quite hard to give sensible errors or recover in a graceful way from any mistakes in the input. An alternative strategy that I'd really recommend these days is to use a real XML parser to parse the XML; this can then handle error reporting, and normalising any encodings used. XSLT or similar technology can then be used to generate a TEX document in more standard TEX syntax and in ASCII format, which is easier to process with TEX. That said, `xmltex` does seem to be used in some places still and has had very few bug reports so, within its limitations, it does seem to work well enough. I recently had a request from someone asking if he could take a more active maintenance role in `xmltex`. I have no objection, so (depending if this person decides that he wants to take it on) I may be handing over `xmltex` in the near future.

DW: Please tell me about your motivation for becoming a blogger (`http://dpcarlisle.blogspot.com`).

DC: I've often been active in public forums (`c.t.t`, `xsl-list`, ...) but in a blog one is less constrained by the mailing list process. One reason for not blogging earlier (apart from lack of time) was a feeling that I ought to do as some other well known XML bloggers have done, and design my own XML-based blogging engine. That would be possible, but at some point you realise that there are some projects for which you'll never have free time. Also, and perhaps more importantly, I think one of the main jobs of the Math Working Group is to tell people how to put mathematical documents on the web, and in particular to make it as easy as possible for people to do that. If the answer to having a mathematical blog is to design your own blog engine, and use a highly customised web server, then effectively you are telling people that it can't be done. So I thought I'd try just using an off-the-shelf blogging engine and see how I get on. Google's blogger service is perhaps the archetypal free blog. It's early days yet, I've only made 13 posts, and at times I find the blogger interface infuriating; it appears to be impossible to get MathML into comments for example, and its editing interface has a habit of "helpfully" inserting
 tags at random places thus making invalid MathML (or even XHTML) that is being submitted, but it works and they seem to be actively working on it (I just joined after blogger had made a big upgrade to which they have now switched all existing bloggers). So far it's a more or less technical XML-oriented blog; I haven't tried the more discursive style with photographs, family incidents, etc that one finds in some other blogs. Nor have I mentioned TEX yet! As I say, it's early days, I'm not sure yet where the blog is going, if anywhere, we'll see ...

DW: Thank you very much for participating in our interview series. If your involvement with W3C ever brings you to its headquarters in Cambridge, Massachusetts, at some point, please let me know so we can meet in person.