Ross Moore

Ross Moore is a long-time TUG board member and TEX contributor, especially in the areas of mathematics and Unicode support.

[Interview completed 12 December 2007.]

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

Ross Moore, interviewee: I'm an academic mathematician, lecturing at Macquarie University in Sydney, Australia. My undergraduate studies were at the University of Melbourne followed by a few years of postgraduate work at the University of Oxford, before returning to Australia. There were several years spent in Canberra before moving to Sydney, where I've lived now for more than 20 years. Much of that time was spent with Penny (now deceased), both as a bridge partner and partner in life. Although we had no children together, there are four girls from her previous marriage, and now five grandchildren. The 2007 year was a very busy one for me. For five months I was on sabbatical in Switzerland, working at ETH Zürich helping to prepare for the largest-ever meeting of (applied) mathematical scientists. My partner Robyn accompanied me. We travelled a lot, including a quick trip back to Australia when my mother died, and during this time we decided to get married. The knot was tied at the end of September, after we had been back in Sydney for a couple of months.

DW: Congratulations to you and Robyn.

RM: Most of my research work, since leaving Oxford in 1981, has concerned developing software techniques that should be of use to mathematicians; e.g., helping to present their work in the best possible ways, using electronic software tools. Much, but not all, of this has been TEX-related. Also, programming languages such as PostScript, Mathematica and Perl have been, and remain, very important to my work.

DW: Looking at your web site (`http://www.maths.mq.edu.au/~ross/`), I detect that you are holding back some personal information, for example, about the Bush Band and what kind of music it plays.

RM: I was not part of the band. Mostly they played folk music. In Australia this tends to be derived from Irish, English and Scottish folk tunes, perhaps with lyrics reworded to describe a local historical event.

DW: Having a university math faculty position that allows you to do research on software to help do math rather than doing research in math itself sounds a little unusual. The last several math professors I interviewed said they got little encouragement and no support for their TEX activities. The Macquarie University web site (`http:`

`//www.mq.edu.au/`) calls it “The Innovative University”. Is the university non-traditional in some way? Should I presume that you do have to do some traditional math teaching and research as well as your software work? And what is the set of software tools to help mathematicians you have worked on?

RM: The “Innovative” title refers more to research and development, with quite strong links to commercial companies located close to the campus, than to teaching practices. Certainly we do traditional teaching of mathematics, though I prefer to use computer software in my teaching, rather than writing down everything on a blackboard, whiteboard or onto overhead-projector slides.

As for my software efforts, this is not so much writing completely new software tools, but more about realising the ability of existing tools to become much more useful in areas where previously they had not been sufficiently-well applied. This includes writing macro packages for TEX, programming directly in PostScript, and coding in general-purpose software applications such as Mathematica and Maple. And then there is LATEX2HTML, written in Perl, and coding in PHP for the production of web-pages containing mathematical content.

I have been lucky in finding professors, both at my university and elsewhere, who have recognised the value of this kind of work and have provided support, including travel. In return I have helped them produce “camera-ready” copy for four books published in hard-cover, mostly Proceedings-like volumes, and one monograph in soft-cover. Each of these projects has involved developing some special TEX techniques that were not hitherto available. It has also led to my involvement in the organisation of large conferences, including the one in Zürich, where I develop the web site for collecting and processing abstracts both for online access and for printing in the Program and/or Abstract Book(s).

As a specific example, the original version of Xy-pic showed great promise to be useful for commutative diagrams and such-like. Category theory is a particular strong point in my Department, and their work requires more than just the straight-line kind of diagram that was supported to some extent by existing macro packages. So I added support and drawing methods for spline curves, which then extend to methods for specifying knots, braids and 2-cells, as well as general curved paths and arrows. Also I added color and driver-specific output support for different TEX engines. In particular, the PostScript backend which I developed, greatly improves the quality of output that can be produced by Xy-pic, and allows for some graphic effects that are not obtainable in any other way. A direct result of this is that Xy-pic has become a vital part in publishing research work in category theory world-wide, not just at my university.

DW: How did you first get involved with TEX?

RM: Aaah, that goes back to the early 1980s. I’d been dabbling in symbolic manipulation packages to do mathematical calculations. Back then it was REDUCE, Macsyma and muPad (where now we have Mathematica, Maple, and MatLab). I needed a way to present the machine-generated results and a colleague showed me the LATEX manual. I got an account on some VAX machines, which could be used to run the mathematics and also typeset it. There was a visitor, from Germany I think, using the same laboratory who used plain TEX rather than LATEX. This prompted me to read *The TEXbook*, and find out how to get around all those annoying aspects of LATEX layout that were so hard to change. Thus I became a TEX programmer, rather than just a (LA)TEX user.

After a while Macintoshes started appearing at the university. This changed the computing paradigm to a WYSIWYG kind of interface, for word-processing. I experimented with these for awhile, and (dare I say it) rather liked the abilities of the earliest versions of Microsoft Word on the Mac, at least when compared to MacWrite. But then they bloated the interface and kept changing how to do things. This turned me right off of it. By now there was a decent TEX application, which soon became Textures. It cost a bit of money, but was fun to use. OzTEX also came onto the scene, and was more like the interface that I'd been using under VAX/VMS and Unix. I started to understand the issues concerning different drivers and printer resolutions.

It wasn't until LATEX was rewritten in the 1990s, with much improved support for packages and document classes, that I began to embrace this instead of continuing to use Plain TEX. I needed a package for general commutative diagrams that could be used with LATEX. AMSTEX and LAMSTEX were candidates, but not sufficiently flexible. Then I discovered XYpic, and helped Kris Rose extend it to be much more useful for mathematics. The first Proceedings volume that I edited used both LATEX and Xy-pic (note the name change); this appeared in 1995.

This is about the time that the World-wide Web was born, and the first HTML recommendations. But that's the start of another story.

DW: And your part of that story is . . .

RM: . . . extending the mathematics support within LATEX2HTML. This conversion software, written in Perl, was originally developed at the Computer-Based Learning Unit of the Education Department at Leeds University, primarily by Nikos Drakos. This was when web-browsers were not yet very sophisticated. Support for mathematics was minimal, based upon simply creating a (LATEX-generated) image. This could not work properly with regard to equation-numbering, cross-references, or hyperlinking to sub-parts of a set of displayed equations, and such-like, and had difficulties getting displays properly sized and aligned. So I set about developing different levels of mathematics support, based upon the structure of the layouts used within different environments. There are now options that allow mathematics coding to be fully parsed down to the level of individual characters and symbols, or to lesser levels at which images can be generated and aligned. The resulting HTML coding can refer to mathematical symbols in any of various different ways, according to what a browser can show; named entities, parametrised entities, UTF-8 strings, or as images.

The way that indexes and bibliographies were handled by LATEX2HTML also needed an overhaul, as well as extending the parsing of tabular material by doing more detailed processing of the column-specifier argument. In order to be able to align images in a web-browser, there is delicate TEX programming in the preamble of the LATEX job that is used to automatically generate the images.

Also, I did some work to support old TEX-based pre-processor methods for typesetting Indic languages. This results in LATEX-generated images, such as with the earliest ways to handle mathematics. As Unicode has become more widespread, these methods will become obsolete; nevertheless, they continue to work with legacy compuscripts.

DW: Will Robertson mentioned in his interview that you are doing some development work related to XƎTEX. Please tell me about that and other TEX development work you are doing (or have done), in addition to just using TEX.

RM: The first part of this work was to provide backward-compatibility with existing LATEX documents, written to use packages that are now quite redundant for processing by XƎTEX; for example, the `inputenc` and `fontenc` packages. XƎTEX requires Unicode-compatible input (ASCII, UTF-8 or UTF-16), whereas support for different languages in LATEX has been to refer to characters in special fonts, each having its own customised encoding. We are all familiar with using macros such as `\'`, `\"`, `\^`, etc. for putting accents over letters, and `\textcopyright`, `\textsterling`, `\textdegree`, etc. for other characters.

To work with XeTeX, all such macros needed to be redefined to produce references to Unicode code-points. I wrote a new LaTeX package, now called `xunicode`, that encodes the correct Unicode codepoint for all the symbol-producing macros that occur within the standard font packages in a usual LaTeX distribution. Along with Will's `fontspec` package for accessing OpenType fonts, the `xunicode` package is recommended to be loaded whenever processing LaTeX source that has not been prepared entirely in UTF-8 or UTF-16. It actually does a bit more than this, since it was written to be fully compatible with LaTeX's NFSS font-selection scheme. This means that, by simply changing the value of `\fontencoding`, a document can use the older LaTeX method of accessing legacy fonts, as well as the new (XeTeX-only) direct method.

More recently, Will, Chris Rowley and myself have been working on developing full XeTeX support for mathematics using Unicode-encoded fonts; in particular, the new STIX fonts, and other fonts that include mathematical symbols at the proper Unicode code-points.

DW: How do the various tools you are working on as part of your research (Mathematica, TeX, etc.) fit together?

RM: This is a good opportunity to talk about labelling graphic images. Software such as Mathematica is great for producing graphs of mathematical functions and scientific data. However, typically the labelling features of such programs are rather poor, using just ASCII strings to label axes and tick-marks. Typically the graphs need touching-up in a sophisticated graphic editing tool such as Adobe's Illustrator software. But if you want properly typeset mathematics in labels, even this is not enough.

One approach is to pre-typeset labels using TeX or LaTeX, then include these using Illustrator. Because of the non-standard encodings of the fonts that TeX has traditionally used for mathematics, this method may not always work. Also, changes with different versions of Illustrator has meant that graphics files produced this way have stopped working properly with later updated software.

A good solution to this problem is to keep the graphics and labels separated; that is, import the graphic into a TeX document as an image, then overlay the image with the desired labels, which then have the mathematics correctly typeset. This has the added advantage of keeping the style and fonts used in the labels consistent with what appears within the surrounding text content of the document. LaTeX's "picture" environment is one way to achieve this, using a coordinate system that needs to be set up for the environment containing the image. Another way is to use the `xyimport` extension, which I wrote for use with Xy-pic diagrams, which then gives a natural way to use the full drawing capabilities of Xy-pic to annotate graphics with symbols, lines, paths and arrows as well as typeset labels.

Taking this a step further, the `warmreader` package implements the idea of having symbolic labels to indicate places of interest within an imported graphic. (This is like using LaTeX's `\label` command to attach a name to a location within a document.) For this to work, there needs to be a kind of auxiliary file that contains information about the size of the image and coordinates for the named points of interest, called "marked points". Wendy McKay had a real need for this kind of labelling technique for a collection of images that had been originally created years earlier, using the then-current versions of Adobe Illustrator. So we enlisted the help of a programmer at Adobe Systems Inc., Thomas Ruark, who wrote a “Marked Objects” plug-in tool that gave a point-&-click interface for selecting places of interest and recording their coordinates and a symbolic name within a file, using the appropriate data-format.

DW: Your math department page (`http://www.math.mq.edu.au/staff/ross.html`) tells me that part of your post graduate work in the UK included a PhD from Oxford. The math genealogy web site (`http://genealogy.math.ndsu.nodak.edu/html/id.phtml?id=14166`) lists a Ross Moore whose advisor was Roger Penrose, who is well known outside the math world. Is that you?

RM: Yes, it is. I wasn’t aware of this entry until you mentioned it. Probably someone at Oxford or Cambridge has submitted a list of Roger Penrose’s students. My entry should now have been updated with a bit more information.

DW: The typesetting world has largely moved to InDesign, QuarkExpress, etc., the word processing world has largely move to Word, and the [claims for] typesetting capabilities of these other systems keep getting better. Will there be reasons for mathematicians to continue using TEX into the indefinite future?

RM: Yes, indeed. Apart from the results being aesthetically more pleasing, the “language” that TEX uses to record a mathematical expression is just so much easier than with any of these other tools. This is particularly true for displayed equations, matrices and tables; not to mention commutative diagrams, expressed with Xy-pic say.

Agreed, these other systems are getting better; in particular Unicode and the new STIX fonts will help close the gap even further, by moving a lot of the layout aspects of mathematics presentation into the fonts themselves. This will mean that mathematics will be represented using strings of UTF-8 or UTF-16 characters, rather than as ASCII strings as in traditional TEX (or LATEX) source. But this requires appropriate, easy to use, tools to construct such character strings, or generate them from something else (such as TEX coding). These tools do not yet exist, or those which do are not close to having widespread use within the mathematics community.

For example, systems such as LyX and Scientific Workplace use LATEX as the underlying typesetting engine and as the export format for publishing and interchange with colleagues who may be using other tools. MathML has not yet attained general acceptance within the mathematics community; it is used by some publishers, but the best typesetting of it requires conversion back into TEX or LATEX anyway.

DW: You have been a member of the TUG board for a number of years. Please tell me how that came about.

RM: Back in 1997, at the annual TUG meeting in San Francisco — the first that I had attended — Kris Rose and I were asked to sit in as “observers” at the Board meeting. There were several vacancies becoming available on the Board, and I was naïve in not realising that we were in effect volunteering to fill these. I’ve been there ever since, which has been quite fun and has allowed me to do some things that otherwise wouldn’t have happened.

DW: What is your personal or general view of the work done by individuals such as Kew, Koch, Hagen, et al., to keep TEX development moving?

RM: I jumped in quite early with Jonathan Kew’s work on XƎTEX, which I see as having the potential to become the platform that will most likely support use of the STIX fonts for mathematics. Either this, or it will evolve by merging with pdfTEX (or LuaTEX) to create a bigger, better, more flexible system employing the “best of all worlds”.

The Macintosh computer has always been very popular with mathematicians. (I first used one at work in 1985, and got my own in 1986.) The change to a Unix-based operating system, Mac OS X, meant that existing software for TEX would become obsolete. This included the Textures application, which had been very popular in parts of the mathematics community; because of some "ease-of-use" features that it had, which were not available in other TEX applications. Gerben Wierda and Richard Koch were already doing the work that unleashed all the power of a Unix-based TEX system, but with a "Mac-like" human interface. With more than a little urging by Wendy McKay, other people joined-in and their work has became the MacTeX project, which is now the free Macintosh variant distributed with TEX Live.

As for Hans Hagen, he just does so much that it is impossible to keep up with all his work. Some TUG members may remember TUG2001 (`http://www.tug.org/tug2001/bulletin/preprints/`) in Delaware; where I gave three talks, and Hans gave four. I'll not repeat that performance, but Hans does so frequently.

DW: Your math department page also says you are webmaster for a couple of math societies, you mentioned the first conference proceedings you did with TEX, and you recently told me you are involved with another conference proceedings. You obviously do a lot of pro-bono work; what is your motivation for that? Also, what improvement to the world of TEX would help you do such work easier?

RM: Software for data presentation and publishing is moving ahead rapidly, quite irrespective of the special needs of mathematics. This is perfectly natural; but it can be leaving the average mathematician far behind, using older tools which may become incompatible with the newer developments and techniques. It takes someone with sufficient mathematical expertise and training to be motivated enough to even test new combinations of software techniques applied to mathematical content.

The main motivation is about seeing needs arising, or an opportunity developing, where I have appropriate experience, knowledge and ability to make a useful contribution. Whilst this kind of work is not really scientific research *in* mathematics, it certainly involves a kind of social research *for* science and mathematics. My work can, and does, lead to techniques and time-saving improvements to the way other researchers conduct, record and publish the results of their own research work. Greater recognition of this aspect, by research-funding agencies, would make my university life easier, by improving chances for promotion and research grants.

DW: Thank you very much for taking the time to participate in this interview series. It has been a pleasure for me to learn a bit about your life and work.