Will Robertson

Like many graduate students, Will Robertson has gotten involved with TEX; he is also representative of a minority of such TEX-using graduate students who take the next step of contributing significantly to the TEX community.

[Interview completed 25 March 2007.]

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

Will Robertson, interviewee: Turning 26 this year, I'm not old enough yet to have a personal history. I can tell you who I am, however. I am currently in the second phase of my PhD in Mechanical/Mechatronic Engineering, for which I'm (approximately) trying to build a table that floats on magnets. My current plans are to finish up in about a year, but we'll see how that goes.

I'm living and have always lived (but most probably won't *always* live) in Adelaide, in South Australia. We have apparently the greatest number of serial killers (or is it murders?) per capita in the world. We have the second largest Fringe Festival (arts, theatre, dance, and so on) in the world, a statistic I can't really comprehend given our size. It certainly transforms the city for almost a month of every year, which is just starting up as we speak. Outside the city, we've got a fantastic wine industry. Hopefully it remains sustainable while we run out of water.

I also work at a Chocolate Cafe, which keeps me busy when I'm not researching. While it helps pay the bills, I mostly do it for the side benefits: access to lots of great chocolate and people. Although I'm biased, we serve the best hot chocolate I've ever had: we use actual melted Belgian chocolate, instead of chocolate powder or syrup, frothed up with steamed milk. And we can also make it with soy milk, to cater for the vegans. Real chocolate doesn't have dairy in it, you see.

DW: How did you first get involved with TEX?

WR: My friend introduced me to TEX at the end of my third year of uni (Mechatronic Engineering), which was around the end of 2001, I suppose. I started out using LyX, and relatively quickly migrated to LATEX proper. LyX is mostly fine if you don't know LATEX and can live within its restrictions. Knowing LATEX, I now find LyX rather clumsy when I'm forced to help people in the department customise their documents. Some of its features are quite convenient, though.

At first I was using LATEX for technical purposes: reports and so on. It was the little things that we liked about it — ligatures, automatic numbering, sensible float and referencing behaviour, and so on. All this (except the better output quality) we could do with

Word at the time (provided you were very finicky when writing the document: "styles" provided the content/formatting separation we laud in LaTeX) but it was less effort in LaTeX. I don't know if that is still the case, but I'm very glad that Word is getting a proper maths renderer in Office 2007 (we'll probably return to this point later in the interview). At last, mathematical documents produced in Word will be readable!

Anyway, after I started my PhD, I begun exploring the LaTeX world in depth. Contrary to all of the recommendations, I actually found that I didn't need a book to learn anything. Long hours exploring packages and reading mailing lists sufficed. It took me a couple of years, I guess, before starting to program with it proper, and now I can't really give it up. I no longer take my laptop to uni, in order to concentrate more on my work, so I don't have the opportunity to spend more than a few hours a week on LaTeX at the moment. That's good, else I'd never get anything done.

DW: You mention helping people in your department. Please tell me a little more about this; for example, is there "official" support for TeX in your university?

WR: As for the usage of TeX that I see, I can only speak for what I know in my department, specifically, but I suspect it's probably a similar situation for many other universities. In general, it seems that many people aren't very computer literate. This is a broad problem; people will do data analysis in Excel rather that a decent program like Matlab; mathematics is still done by hand on paper rather than with Mathematica (well, I guess this is more understandable; I think you need to "click" with Mathematica before it's very useful); Word is used instead of LaTeX.

However, not everyone uses Word. My PhD supervisor used LyX for his thesis, and he recommends it to all of the new PhD students. Anyone I can influence uses LaTeX, because I tell them I can't (or won't) help them much if they use LyX. This is kind of true: some things I simply don't think are possible within the LyX environment; my main excuse is generally that it's a pain to debug LyX when things go wrong from a LaTeX point of view.

Michael Murray, who is (or has been, when I was) active on the Mac OS X TeX mailing list, is actually the Head of the maths department of our university. Over there, LaTeX is just what everyone uses. When you're writing maths day-in, day-out, I guess it only makes sense. Mechanical Engineering doesn't follow their lead, but maybe Electrical Engineering does.

As far as TeX support goes, as might be obvious to an observer, there is none. The university runs training courses in various Microsoft products, but that's as far as it goes. We have a very strong "pick it up as you go" culture in the software world (as I see things generally), which means that software needs to be designed with this in mind — able to be grasped non-committedly and coaxed to produce something without too much effort. Bearing this in mind, LaTeX can be a little difficult to approach (to say the least) for the casual user.

Furthermore, where would TeX support come from? It would be unfeasible to suggest that TUG could provide any, and there aren't any Australian "LaTeX User Groups", as far as I know. So people rely on the local expert, which is me — and eventually I teach them how to use Google Groups to answer their questions, as they are usually trivial to solve.

DW: Do you have any contact with or awareness of TeX users in other universities such as Baden Hughes in Melbourne (who was a founding member of the TPJ editorial board) or Ross Moore in Sydney (who is a member of the TUG board)?

WR: Like I said above, there is no Australian TeX user group that I know of, and so communication between "well-known" Australian TeX celebrities only happens through

happenstance. I didn't know that Baden Hughes was from Melbourne, but Ross Moore and I have been working on some XeTeX things together for a little while now (not that we've met — we're 1400 km apart). We started collaborating through XeTeX connections, however, so it's just coincidence that we're compatriots.

DW: You and I first became acquainted through working on *The PracTeX Journal* (TPJ). How did that come about and what is your motivation for putting effort into this TUG activity?

WR: That's right. To be honest, I can't remember how I got roped into all that! The email conversation that initiated it seems to be temporarily lost in time (I'm sure it's somewhere in my mail), but I vaguely remember offering a bunch of suggestions early on, and assumedly Lance Carnes (Editor) thought that my opinions were along lines that weren't well-represented with their then-members of the board. (That is, if I was just re-iterating opinions of the board, my help would have been less valuable.) Somehow he also recognised the opportunity to share some of the production burden, and I started "production editing" a couple of papers per issue almost immediately.

These days, we've got more helpers, and I've been less able to spend as much time. To be honest, I see the production work I do (sending off articles for review, reviewing articles myself, and copy-editing others) as an enjoyable chore with the side-benefit of learning a good amount about how (online, at least) journals work from the inside. As a hobby, there are much worse ways I could spend some of my time.

DW: Also, I see from the author index that you have written three articles for TPJ.

WR: I'm much happier *writing* the articles, but that takes a lot more time. Like you say, I've written a few now, on a somewhat diverse range of topics. My motivation for them comes from a few sources. Firstly, I need to practise writing. I wouldn't say that I love writing, nor that I am "a writer" (although I wouldn't mind trying it one day), but when I write well I enjoy it, and I can only write better with practise. (Whether I write "well" ever, anyway, is certainly up for debate.) Most important, probably, is the aspect of sharing knowledge around. The things I write about for *The PracTeX Journal* aren't new or novel, they're just reporting facts and opinions — the journalism side of technical writing, I guess you could say. When I spend time looking into, to use an example, the features of the Latin Modern fonts, and many people still seem to be ignorant of them, it only makes sense to write about it and make the time I originally put in better spent. The LaTeX world is a bit of mess due to its age: there's a lot of useless stuff lying around in CTAN that should be retired but can't be for backwards compatibility. Trying to teach people to use the current "best practise" seems a worthy cause to me. That's the type of thing I can't really quantify, though. Lastly, to be egotistical, it's nice to see your name in print.

DW: My memory is that in addition to serving as a production editor, you took over maintenance of the LaTeX class for TPJ.

WR: The pracjourn class was the first document class that I published for people to actually use, I think, and really got the ball rolling for me in terms of getting used to the issues involved with that. Formatting the frontmatter (name, affiliation, email address, etc.) was an interesting typesetting job, and I (still) think the design I settled on works quite well. At least, I haven't tired of it yet. I've never actually heard feedback from anyone whether they like or even notice it, though.

DW: I think it functions well for its purpose.

WR: I've gone on to write fairly basic classes for a few conferences that I (or colleagues) have attended. I've been quite disappointed with the quality of some of the ones I've come across that I haven't written; in some cases, the conference organisers have just coerced a poor graduate student who might know how to write a LaTeX document into writing the class. The results aren't pretty; the last one I saw had instructions to write sections something like

```
\section*{\centerline{1.1 THIS IS A SUBSECTION}}
```

That's right. Manual section numbering. Manual formatting. Not exactly what we're used to.

Going a little off-topic, I think organising some sort of TUG-organised, community-driven service for conference organisers to create one-off classes for conferences would be a decent thing to have to ensure that this sort of thing doesn't happen in the future. After all, once you've written one conference class, there's not too much that needs changing to adapt it for new typesetting guidelines.

DW: Since you reached the point where you could "program with it proper", you have obviously gone a lot deeper into it than many TeX users with only a few years experience. Can you give me a brief sketch of the purpose or function of fontspec, and how your development of it came about? I noticed you wrote an article for *TUGboat* about it: "Advanced font features with XeTeX — the fontspec package" (`http://tug.org/TUGboat/Articles/tb26-3/tb84robertson.pdf`).

WR: This answer needs a little bit of background. Bear with me while I try to be brief. As we probably all know, TeX was invented before computers were powerful and even before the notion of a "computer font" was well established or at least standardised. So its methods for dealing with fonts have been very quaint for some time now.

In order to overcome the many limitations that TeX imposes on the fonts it can use (as well as the input of multilingual text), Jonathan Kew wrote the XeTeX program, an extension of TeX that accepts Unicode input and supports OpenType fonts. Unicode input is important to be able to *write* in any desired language, and OpenType fonts are important to be able to *typeset* them.

To cut a long story short, the way XeTeX interfaces with OpenType fonts is rather low-level. Font features such as using lowercase numbers instead of uppercase numbers are activated with hard to remember strings such as `+onum`. The fontspec package provides a more readable (and hopefully memorable) interface to such things with keyval-type options such as `Numbers=Lowercase`.

Secondly, LaTeX itself wasn't designed to accommodate the idea of "font features", whereby a broad range of minor typographical details could be varied between instances of any one font. (Only "macroscopic" variations, such as boldness or shape, are directly provided for.) The fontspec package allows users to select fonts with any combination of font feature, and to change the font features mid-document.

But most importantly, fontspec simply allows users to select fonts *easily*. No need to mess around with extra files that need to be written, or swathes of font definition code. In its simplest form, `\setmainfont{Baskerville}` is all that is required to select Baskerville as the main document font.

DW: Do you have prior background/interest with fonts, typography, etc., or did your interest develop in parallel with your experience with TeX?

WR: I'd always been finicky about layouts and fonts, but it was TeX that really brought out my major interest. Before XeTeX, I experimented with various fonts that could be

installed in TEX, and before that I was experimenting with GUI fonts to a small degree. Mac OS X is bundled with a nice selection of interesting fonts, and I was fascinated with two in particular: Frere-Jones' Hoefler Text, with its wide range of font features such as swash caps, engraved caps, dedicated subscript/superscript characters, and so on (and so on!) — all within a single font — epitomised the abilities of modern fonts to provide for every last detail (in the limiting sense). Secondly, Hermann Zapf's Zapfino was the first script font that I'd used with a large number of contextual ligatures and alternate characters. XETEX allowed us access to these fonts within a LATEX environment, the best of both worlds.

DW: How did you get connected with the XETEX system? Is your fontspec work coordinated in any way with a larger group of people working on expanding the capabilities of XETEX?

WR: The Mac OS X TEX mailing list is certainly what got me started in my TEX hobby. It was there I began asking my naive beginners questions, and slowly learned enough to start answering others'. So for a couple of years, I was quite active there. But I realised eventually that I wasn't gaining anything by spending so much time reading and replying there, and chose to leave. Only so much time in the day, and all that. But these days I browse the `comp.text.tex` newsgroup, which takes a similar amount of time. There's no escape, it seems.

Via the Mac OS X TEX mailing list, we learned about this new XETEX project in April 2004. I started using it a few months after that, but not for any serious work. Even back then, my knowledge of TEX and LATEX was that of a user; I didn't have any knowledge, less experience, of any actual programming in TEX. Over time, it became obvious to me that something needed to be done to make fonts easy to set up in XELATEX (that is, LATEX when using the XETEX engine). Bruno Voisin had a template for setting up NFSS families in LATEX with XETEX-native fonts, but that method was a little inaccessible.

I played around for a while with various interfaces before creating fontspec. You only learn from doing, and this was my first TEX programming project. At first, fontspec simply let you select a font to use immediately. Then to select the font for the whole document. And then to select font features as well. My old versions are lost in time (I didn't know version control back then; I've only started learning it within the last year!) but I remember the slow understanding of how TEX programming worked. Now, I'm a little more confident of my ability to put a package together, and I'm working on a new package for dealing with Unicode maths. Jonathan has been putting in features to follow Microsoft Word 2007's lead in OpenType maths font features, and I've taken up the project to provide the user interface for it all, much in the same way as fontspec is a user interface for XETEX's font access in general. (It's a little bit embarrassing that the TEX community has to follow the lead of Microsoft Word, of all programs, but oh well.)

Chris Rowley and Ross Moore are helping me out with various parts of this "Unicode maths" project, but there isn't a group of us working together to improve XETEX in general. Jonathan works on whatever he feels motivated by, which rather recently has been to improve the maths support of XETEX after Microsoft gave him a model to follow. Similarly, I do what I can with fontspec and a couple of minor packages, and Ross has a "Unicode text" package that's been essential for new users of XETEX to have compatibility with their old LATEX documents. But we've never met, and we communicate mainly via the XETEX mailing list.

It seems to work best this way, in the open source software world — too many people working together gets everyone bogged down trying to work collaboratively, rather than

just heading off in their own directions and seeing what happens.

DW: I assume you will use TEX to write your thesis. Do you think you will stay with TEX after you graduate or drop it as so many other graduate students do after having used it for the few years needed to write a thesis?

WR: I am quite surprised by this question. Of course I will stay with TEX! My interests have extended far beyond what I need to typeset my thesis. I'm reasonably confident, however, that my office mates will have a hard time sticking with LATEX, even though they like it, because they only have me to ask when they run into problems. They don't know where else to look.

This leads me into a few words about where I think the big problems of LATEX lie. Because it's obvious that it is much harder to learn than it could be. And the crux of the problem, the long and the short of it, is that the LATEX kernel has been intentionally restricted to the functionality originally designed for it. Now, this is a good thing, by and large. Standards need to be frozen in order for them to be adhered to. But it's past time to move on and create a new standard. There are fast approaching competitors that will eventually overtake LATEX if we continue to stagnate. (Not to mention ConTEXt, which has surpassed LATEX in essentially every area. Maybe the solution is just for everyone to switch to that.)

As far as a document preparation system goes, LATEX has a lot of work to catch up to what's happened over the last five years in HTML. Introductions to LATEX often laud the separation of form and content in the LATEX syntax, but the implementation is only skin deep when compared to the things that HTML+CSS can do.

Most of LATEX's shortcomings are solved with various packages, and many of those packages are so widespread now that their use is essentially mandatory. The memoir class attempts to unify many aspects of document preparation, but even that noble effort cannot hope to accommodate everything.

The LATEX 3 project is supposed to be working to update LATEX. But they face another problem: they've invented a wonderful extension to the LATEX programming language in order to do all of their coding. But now they're stuck with a system that no-one knows how to use, and no-one will learn it to write their packages because no-one's using LATEX 3 yet. On top of all that, LuaTEX is set to be released in a year or so, perhaps obviating all their hard work on the low-level design.

If I magically were able to make this all happen, here's a rough sketch of things I expect to see over the next few years to get LATEX back on its feet. Who knows, this might be their plan anyway. In any case, after I write my thesis I wouldn't mind helping out.

- With the next release of LATEX 2_ε, the LATEX 3 team releases a formal LATEX 2_ε package that contains the designs of their LATEX 3 syntax. This allows LATEX 2_ε package writers to update their thinking and their packages and, importantly, to learn the new system. Wrinkles in the syntax can be ironed out with (hopefully) widespread use.
- At around about the same time, "someone" (a single person or small group better than a committee) writes a high-level specification of the LATEX 3 document preparation interface. In short, a formal definition of every document element from chapter heading to footnote that could possibly be used within the large majority of documents written in LATEX. Gather as much material from existing packages and classes with the intention of unifying all of the best ideas of the LATEX 2_ε ecosystem into LATEX 3. For example, built-in support for multi-page tables, multiple levels of footnotes, named referencing of document parts, and so on. This is just a design

document, not the low-level code.

- Finally, the actual code is written. By now, we've got a syntax to write code in, and a specification of what needs doing, plus enough people that know the system who can help out. Port over as much as possible from LaTeX 2_ε, and LaTeX 2.5 is released, the interim unstable version of LaTeX that is backwards incompatible with LaTeX 2_ε and that will converge eventually to LaTeX 3, which will remain fixed for another ten years.
- In 2030, repeat the process :).

Of course, talking is easier than doing. But the main theme here is "get people involved". Very few people are going to learn the LaTeX 3 syntax without something to gain from the process. As an academic exercise its appeal is limited, but if the syntax can be used to write LaTeX 2_ε packages then I think its popularity will be assured. (Coming from a novice package writer, the new syntax is a breath of fresh air.)

So back to your question: yes, I'm looking forward to sticking with LaTeX and I'm enthusiastic about its long term future.

DW: I'll close this interview by saying: Thank you for participating in our interview series; it has been good to hear the point of view of a relatively short term user of TeX. Also, best wishes for finishing your graduate work and your life and career after that. I hope we will be able to meet in person at some point.