Tomas Rokicki

Tom Rokicki created `dvips` and TEX's original Pascal to C converter.

[Interview completed 15 August 2008.]

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

Tomas Rokicki, interviewee: I live in Palo Alto with my wife Sue and our yellow Labrador Andy. I moved out to California from Texas to attend Stanford, and liked it so much I've felt no inclination to move away. My favorite pastime is still programming, whether it's cellular automata (Golly), solving Rubik's cube, or programming the Propeller microcontroller. When I'm not programming, I'm usually training for my next marathon.

DW: I was reading about some of those activities at `http://tomas.rokicki.com`, gathering background information for this interview.

TR: I grew up as just another math/science geek. Grammar school near Chicago during the 70s was Radio Shack 150-in-1 kits, disassembling radios and electrical appliances, and studying an electronic engineering textbook my parents purchased for me. High school in Texas was slide-rule competition, microcomputer programming, and chess.

I was fortunate to be accepted into Texas A&M University, but even more fortunate to meet Professor Norman Naugle there. Professor Naugle helped me find jobs that complemented my classwork, and always had a challenging program that needed writing.

DW: Your web site also mentions that you recently had cochlear implant surgery.

TR: My deafness is in remission: The cochlear implant is working *amazingly* well. If you saw me in person it would probably be the first thing you notice about me. I have some words on `http://radbits.blogspot.com/` about it. This is just a *touch* personal, but my life has been so greatly impacted by this amazing miracle that I'm really glad to mention it.

DW: When and how did you first come in contact with TEX?

TR: In my junior year at Texas A&M, Professor Naugle introduced me to TEX; he was trying to get it running on campus in various departments. In particular, he needed a

way to print from TEX to the newfangled QMS laser printers. These laser printers were huge, about the size of a corner mailbox, printed at an incredible 300 dpi, and controlled by a proprietary language. The first TEX code I wrote was a driver for those beasts.

Another task was to get TEX running on the Unix minicomputers that were in use at that time. Since the Unix Pascal compiler was not up to the task of compiling TEX, I wrote a script using `lex` and `yacc` to compile the TEX code to C.

Professor Naugle and I flew out to Stanford for one of the early TEX User Group meetings (I believe this was in 1984), and it was here that I met so many of the famous TEX people, including Don Knuth himself, Barbara Beeton, David Fuchs, and so many others. Once I saw the campus and the people and got a taste of the excitement in the computer science department at the time, I knew that was where I wanted to go for graduate school.

Between graduation from Texas A&M and my first semester at Stanford, during the summer of 1985, I had the unique opportunity to work on the TEX project. David Fuchs generously allowed me to stay with him for a while until I could secure lodging on campus. That summer, and the next few years at Stanford, were some of the greatest times of my life.

DW: Was your method of compiling TEX into C related to how TEX is currently compiled using Web2c?

TR: Sure; it's the same code. The current Web2c project is derived from that code I wrote back in the mid 80's. Many people have extended it over the years, such as the Kpathsearch integration, and better portability, and much, much more.

DW: What part of the TEX project did you work on that summer before graduate school at Stanford?

TR: The primary work was the design and implementation of the PK tools. At that time, disk space was relatively precious, especially on the microcomputers of the time. The PK format stored the same information as GF files, but in approximately half the disk space. So I wrote `gftopk`, `pktogf`, `pktype`, and a few other utilities.

DW: You said your years at Stanford were great times. Tell me something about what you did during those years and that made them great times.

TR: Well, ... the "Programming and Problem Solving Seminar" course was spectacular fun, both the one I took, and the one I later was a teaching assistant for — fairly open-ended problems solved (or just worked on) by groups of first-year doctoral candidates.

Learning to play go from Daniel Weise, and playing my classmates in Margaret Jacks Hall.

Knuth's Concrete Mathematics course was a blast.

Programming the Amiga, and leading a student club centered around that machine for a few years. I had a blast programming that machine.

Just being a graduate student at Stanford back then, and learning really neat things from my classmates and the classes, was just amazing.

DW: Please tell me about when and how you came to develop `dvips`.

TR: I developed `dvips` on the Amiga, based on `dvisw`, a QMS SmartWriter driver I had written earlier. The SmartWriter driver was extremely limited because that laser printer had only about 64KB or so of memory, so there wasn't much room for downloaded fonts. Even the PostScript printers available then had limited memory, however, so managing that memory and making sure a document that used a lot of characters from a lot of

fonts did not run out of memory when printing was a challenge.

In any case, converting `dvisw` to drive PostScript was not all that difficult; most of the code remained the same. I wanted to generate fairly concise PostScript (disk space being somewhat precious; I did not have a hard disk, and floppies only stored 800K each) so there was some effort there.

Getting the output fully correct turned out to be a real challenge; there were bugs in the PostScript interpreters at the time that needed to be worked around. Over the years, as PostScript showed up on different devices, many changes had to be made to support all the different interpreters and their differing characteristics.

One early example is maintaining an even baseline. For downloaded type 3 fonts, the vertical displacement of a character bounding box from the baseline is specified along with the glyph data. For some interpreters, there was a discontinuity at 0; characters with a negative vertical displacement would show up one pixel offset from their desired position. Thus an 'e', which might live entirely on or above the baseline at a particular resolution, would be placed correctly, but a 'g', with its descender going below the baseline, would be rendered one pixel too high. This made characters in a line jump up and down on the baseline in a very ragged manner — but only by a single pixel. At 300 dpi it was fairly noticeable, but at any resolution it was wrong. So `dvips` had to pull some PostScript tricks to make sure these lines rendered correctly both on interpreters with this problem, and on interpreters that did not have this problem.

Another example is rules. `dvips` had been running along well for a long time when NeXT came out with their computer and its printer. For whatever reason, rendering rules on the NeXT was just incredibly slow. Rules were rendered by `dvips` using the "imagemask" operator in order to guarantee precise placement and size, and the NeXT version of PostScript just did not like the imagemask calls that `dvips` generated.

It should be noted that Don Knuth himself performed the conversion of `dvips` to support the virtual font format, and along the way he made some significant improvements to `dvips`. I believe at least one note from him to me remains in the source.

DW: I don't yet have a solid fix on the timing of your original `dvips` work. Was it originally for PostScript level 1 in 1984 or for some other version of PostScript?

TR: I'm not quite sure on the timing myself, but I'm pretty sure I made it work during 1985 or 1986. It was indeed for PostScript level 1; the first version was developed against an original LaserWriter.

DW: Have you continued to maintain `dvips` over the years, or has that been passed on to others in the TEX community?

TR: Frankly, Karl Berry has done the great majority of the work, and for that I am very grateful. I have not been following the TEX community for some time.

The authorship of `dvips` is a lot of people at this point. The original program was entirely my own of course, but Don Knuth made a dramatic cleanup when he modified it to support virtual fonts. The Type 1 support was added by Sergey Lesenko, although at this point we use code from pdfTEX instead. Major contributions have been made by many people; here's a partial list (in alphabetical order, hopefully to prevent hurt feelings), although I'm still a bit nervous about all the great contributors that I am forgetting:

> Karl Berry, Peter Breitenlohner, Mark Doyle, Jim Hafner, Yannis Haralambous, David M. Jones, Akira Kakuto, Don Knuth, Sergey Lesenko, Dave Love, John Plaice, Fabrice Popineau, Sebastian Rahtz, and Ulrik Vieth.

DW: We can certainly add names to our interview's web site if more contributors come to mind later.

TR: In any case, I certainly intend to support `dvips` to the best of my ability and to encourage people to contact me with questions or issues, but most recent issues have about things I really don't know much about, so other people end up doing most of the work.

DW: How do *you* pronounce `dvips`?

TR: I used to always pronounce `dvips` as `d-v-i-p-s`, but a friend of mine always pronounced `d-vips` (in one syllable), and I have sort of adopted that pronunciation.

DW: Please sketch your work life after graduate school, e.g., I've heard your name mentioned in connection with NeXT. Also, is Radical Eye Software a real company?

TR: The only thing I did for NeXT was NeXTTEX, which was as close to a clone of AmigaTEX as I could make it. I did both AmigaTEX and NeXTTEX while I was in grad school.

AmigaTEX did have the previewer and TEX connected through inter-process communication, so you could immediately preview pages even while TEX was still running, and it would immediately refresh the current page when it got to it. (Remember, back then, systems were slow, especially when running from floppies.) In addition, TEX stayed resident and after processing, it was immediately ready for the next job. Over the years additional enhancements were added, like integrated bitmap and PostScript graphics in the previewer. All that work was done under the Radical Eye Software name.

The Radical Eye Software name actually goes back to programs I wrote while still at Texas A&M University; I generally copyrighted anything I wrote with that name, whether it was free software, contracted work, or commercial software.

After grad school I worked for HP Labs for a very pleasurable number of years, with some fun colleagues. I stayed at HP until 1999, when I started Instantis with a few friends, where I still work.

DW: Thank you very much, Tom. It has been exciting to talk with you. Yours was one of the first names I heard when I started using TEX about 10 years ago, and discovered that I needed to use `dvips` every few minutes.

TR: Thanks! I have benefited from TEX in so many ways, personally and professionally, and I am just so grateful to have been part of the early days and have the chance to make my small contributions.