

# Victor Eijkhout

Victor Eijkhout is the author of the book *TeX by Topic* (<http://www.eijkhout.net/tbt/>) and several L<sup>A</sup>T<sub>E</sub>X packages, associate *TUGboat* editor for macros, and a long-time participant on `comp.text.tex` and various mailing lists.

[Interview completed 11 March 2005.]

*Dave Walden, interviewer:* Please tell me a bit about your personal history independent of T<sub>E</sub>X.

**Victor Eijkhout, interviewee:** I was born in the Netherlands and lived there until age 30. By then I had a Ph.D. in mathematics and decided to see more of the world. I have been living in the United States ever since. Right now, I'm in Knoxville, Tennessee, doing computer science at the University of Tennessee.

*DW:* When and how did you first get involved with T<sub>E</sub>X and its friends?

**VE:** I started using T<sub>E</sub>X somewhere in the mid-1980s when I was a graduate student in mathematics at the University of Nijmegen in the Netherlands. I was starting to write papers and had used troff/nroff for a while. Then I discovered T<sub>E</sub>X and started using that, without knowing much about the deeper aspects of it. It was mostly a matter that personal computers started to replace the mainframes and minis at that time, and there was no nroff on PCs or Apple Macs. I also found an article that compared the output of the two packages, and T<sub>E</sub>X was far superior in terms of plain typography, both in text and in mathematics. The PC that I first used T<sub>E</sub>X on was an early AT. To be able to run T<sub>E</sub>X we had to extend the memory from 512 k all the way to 640 k, at quite a cost.

Shortly after, the department bought a few Macs on which we ran Textures from Blue Sky Research, which was a very nice environment. Unfortunately, Macs were expensive in Europe, so most people used Atari 1040s, which were even faster than the Mac (8 MHz instead of 5-point-something), and they had more memory and a bigger screen.

*DW:* You have been a significant contributor to the T<sub>E</sub>X community. Will you please tell me how this came about (and when), why did you do it, and what you see as your significant contributions.

**VE:** When I started with T<sub>E</sub>X we didn't have L<sup>A</sup>T<sub>E</sub>X, so everyone wrote their own macros. Doing that, with Knuth's *The T<sub>E</sub>Xbook* as only resource, was somewhat frustrating. The T<sub>E</sub>X language is and remains weird! So I started taking notes every time I finally figured out something. My notes started growing, and at 80 pages I thought they looked quite nice; I gave a copy to a T<sub>E</sub>X buddy (Nico Poppelier) when he got his Ph.D. He suggested that it might make a nice supplement for an issue of *TUGboat*, or so. The notes kept growing, and at 160 pages I decided that I should go all the way, so with 200-something pages I went to the Dutch office of Addison-Wesley. They saw that this book would be more suitable for a global market than just the Netherlands, so they sent it around, and the British office of A-W accepted it for publication. I had a limit of 320 pages; I think more than half of the chapters have their last page three quarters full. There is a lot to say about T<sub>E</sub>X.

I tried making one joke in the book, but it caused a bad page break, so I took it out. I guess I'm no Knuth . . . .

*T<sub>E</sub>X by Topic* was published in 1991, sold a few thousand copies, was translated

into Japanese, went out of print, and right now you can download it from my website (<http://www.eijkhout.net/tbt/>). PayPal contributions encouraged!

Also in the late 1980s, the Dutch T<sub>E</sub>X Users Group (NTG) got started, and one of the issues that came up immediately was how much we disliked the L<sup>A</sup>T<sub>E</sub>X styles, so I took it upon myself to write some plug-compatible replacement styles (the “ntg styles”, with names `artikel1-3`, `rapport1-3`, `boek`, and `brief`—the latter is a not-compatible letter style that very faithfully implements guidelines of the Dutch bureau of norms). I believe they are part of the tetex distribution. Writing these styles was an educational experience. I learned a lot about the details of how T<sub>E</sub>X places material on the page, which was definitely good for my book. In fact, participating on the Dutch and UK T<sub>E</sub>X mailing lists taught me all sorts of obscure facts that are in the book. In retrospect it’s interesting to see how I went from asking silly questions to answering silly questions.

By the way, at the time I was no L<sup>A</sup>T<sub>E</sub>X user, so my only documents were tests for the styles I was writing. Since I needed only one document for three different article styles, et cetera, I was probably the only person to have written more document styles than actual documents. Since then, of course, I’ve been a complete convert to L<sup>A</sup>T<sub>E</sub>X. The number of add-on packages is simply amazing, so even though L<sup>A</sup>T<sub>E</sub>X is not a very good standard, it is good that we have this standard. I use L<sup>A</sup>T<sub>E</sub>X for all my writing, and lately have started using the beamer package for presentations.

Oh, for my thesis I wrote my own macro package, Lollipop, with which I was going to conquer the world. Right. I believe the manual still states that a new version will come out in 1993.

Apart from a course that I taught last year, I have been mostly a L<sup>A</sup>T<sub>E</sub>X user for the last couple of years. I’ve only written a couple of interesting macros, although one, my comment style, is somewhat successful. I found that it’s even part of the L<sup>A</sup>T<sub>E</sub>X2HTML translator.

*DW:* I use a little macro based on one of yours I found in `comp.text.tex` in many of my papers.

But you didn’t mention that for some years you regularly wrote a section on macros in *TUGboat*. How did that come about?

*VE:* My involvement with *TUGboat* came about around 1990 when I was drafted by Barbara Beeton, eh, I mean: barbara beeton. We had corresponded about a conference report that I wrote with Johannes Braams and Nico Poppelier about the first Dutch T<sub>E</sub>X days. Later I met Barbara in real life, and she suggested the idea of a regular macro column to me. A number of times I wrote about macros of my own devising, and other times I explained clever macros by other people.

The column has slowed down considerably in recent years. That is partly my diminishing involvement with T<sub>E</sub>X, but also the fact that *TUGboat* is not as necessary any more as it was before the World Wide Web. Back in the days (‘grandfather recalls’ alert!), it was hard to find stored information. There were ftp sites, bulletin boards, archie sites, but it was hard to find out about them, so a printed resource was of great value to collect information. With CTAN, the UK-T<sub>E</sub>X FAQ, and in general good search possibilities over the web, this is no longer so much the case.

That said, information on the web is often very sketchy. “Just google for it” is an easy way of dismissing a question, but not necessarily a surefire route to an actual answer. Articles in *TUGboat* are still longer and more detailed than much of what you can find online. There is something about being in print that makes people try a bit harder to write a coherent story, so *TUGboat* is still worth having.

*DW:* You said that  $\TeX$  is weird. Are you familiar with Knuth's informal essay on his involvement in the history of attribute grammars where he says, "... I haven't been able to figure out any good way to define  $\TeX$  precisely, except by exhibiting its lengthy implementation in Pascal ... surely an unsatisfying way to define semantics ... I don't know any way to define any other language for general-purpose typesetting that would have an easily defined semantics, without making it a lot less useful than  $\TeX$ . Macros are the main difficulty: As far as I know, macros are indispensable but inherently tricky to define. And when we also add a capability to change the interpretation of input characters, dynamically, we get into a realm for which clean formalisms seem impossible." What is your reaction to that?

*VE:* I didn't know about this article. Reading that quote, my first thought about Knuth being unable to define  $\TeX$  precisely was 'Well, you brought that on yourself'. Which of course raises the question whether  $\TeX$  has to be the way it is. Can something as powerful as  $\TeX$  be made without having a macro expansion language—which I would agree is inherently tricky to formalize—and in particular without having the dynamic changes to the meaning of characters?

While I'm no expert on programming languages, I'm not convinced that a macro language is necessary. And there are definitely some big disadvantages to the current approach: no matter how sophisticated the macros, they still fail to shield the user from  $\TeX$ 's lowest levels. Give a slightly wrong input, get a completely uninformative error message.

What would a non-macro version of  $\TeX$  look like? I have no idea.

But consider this: most people use  $\LaTeX$ , which is very simple in structure. A competent student can write the beginnings of a  $\LaTeX$  parser, or a  $\LaTeX$  to HTML translator, in a matter of days, a few weeks tops. In other words, what suffices for 99% of the people is actually very simple to parse. Maybe a powerful text processor should start with a simple structure that covers most bases easily, and then have a way of breaking out of that to have the infinite power of  $\TeX$  with more complicated, but less used, mechanisms.

*DW:* That leads me to a slightly related topic. What do you see as the significant components of the  $\TeX$  community? What do you see as the strengths and weaknesses of the  $\TeX$  community?

*VE:* The greatest strength of the  $\TeX$  community is also its weakness: the fact that we have this rock-solid, completely portable program to work with. Unfortunately it is that way because it is essentially frozen, with all its weaknesses. Another strength is the community of developers around  $\LaTeX$ . As I already indicated, I am very impressed with the quality and the variety of packages that can be added to  $\LaTeX$  for just about any functionality. That also means that people are finding ways around  $\TeX$ 's weaknesses.

*DW:* Do you have an image of how  $\TeX$  and the  $\TeX$  world will or should evolve going forward? Or are you thinking more about non- $\TeX$  things now?

*VE:* Well, I can try to see in my crystal ball, but for my opinion there are a hundred others. I'm too far out of the  $\TeX$  world these days to say anything sensible about Con- $\TeX$ t, Omega, NTS, and whether they can take over from  $\TeX$ .

*DW:* From your remarks, you seem to be saying that you are not as involved with  $\TeX$  as you once were. Yet you recently circulated draft notes on a course you teach on the computer science of  $\TeX$ . I also think I remember some fairly recent answers by you to questions on `comp.text.tex`. What is your level of interest and involvement in  $\TeX$  these days other than using  $\LaTeX$  and beamer for papers and presentations?

VE: My involvement in `comp.text.tex` these days seems to be limited to pointing people to my book, when it's in danger of being overlooked. Really, there are many people there who very helpfully and extremely competently answer questions. I rarely have anything to add.

My  $\TeX$  course, titled “The computer science of  $\TeX$  and  $\LaTeX$ ”, was an interesting project. One of the faculty members here at UT suggested that I teach such a course. The more I thought about it, the more I found that  $\TeX$  can be used as an easy excuse for teaching all sorts of cool computer science and mathematical topics. Here's in a nutshell what I wound up teaching.

I gave the students an introduction in  $\LaTeX$  and  $\TeX$ , following that up with a segment on how compilers parse languages. They then had to use the Unix tools `lex` and `yacc` to write a simple  $\LaTeX$  parser. Most people know about  $\TeX$ 's paragraph breaking algorithm, and that it uses dynamic programming to reduce an exponential problem (‘given  $n$  words in a paragraph, which of the  $2^n$  possible ways of breaking it into lines looks best’) into something that is less than quadratic in cost. I explained dynamic programming. It is not so well known that a student of Knuth's (Michael Plass) did a thesis showing that page breaking can be NP-complete, because of the figure placement problem. I explained about NP-completeness and showed the outlines of Plass' proof, which I thought was fascinating.

Fonts, and of course Metafont, lead to splines, approximation theory, and various topics that are more usually associated with computer graphics. Certainly raster graphics is in that corner. Researching this topic I came across an interesting fact: all literature I had ever seen talks mostly about cubic splines, but one of the most common font technologies (FreeType) uses quadratic splines, which I had never seen before, and to this day haven't seen outside of the FreeType reference manual. Once I figured out the math of them, I gave this to my poor students as homework.

Character encoding, as exemplified by the `fontenc` and `inputenc`  $\LaTeX$  packages, is another interesting topic, and I had fun reading up on all the intricacies of Unicode and Internet characters protocols. The one chapter I didn't finish was about  $\TeX$ 's macro mechanism and how it can be used to implement lambda calculus. There is an old article by Alan Jeffrey in *TUGboat* about this. I pursued this further, and got as far as implementing a prime number sieve, but didn't have time to complete the lecture notes.

Preparing for this course was a monstrous amount of work, but very fascinating. I'm going to try to turn my lecture notes into another book.

DW: I look forward to buying a copy of your book when it is available. For now, please tell me a little bit about your other work or activities outside the  $\TeX$  world.

VE: In my work I'm doing numerical analysis, which I have a degree in, but gradually I'm creeping more and more into real computer science topics. I've become interested in performance optimization, and in particular automatic ways of generating optimized versions of an algorithm. Related is a recent project where I'm applying statistical decision methods to the question of picking the best numerical algorithm.

Outside work I do ballroom dancing and I make a lot of music. I play the recorder, have just bought a bass guitar, and have a number of completely electronically generated compositions online.

DW: Thank you very much, Victor, for taking the time to participate in this interview. I'll think of you more personally next time I look something up in your  *$\TeX$  by Topic* book, and I greatly look forward to your future writings.