

Taco Hoekwater

Taco Hoekwater is heavily involved in implementing many significant upgrades and improvements to various \TeX -based systems.

[Interview completed 13 October 2006.]



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

Taco Hoekwater, interviewee: I grew up in a small town in the Netherlands, and after some moving around, I now live in the nearby city of Dordrecht. My wife and I live in a small house that stands on the side of a dike. We have been married since 1993, and we have three cats and two dogs.

Ever since I was a small kid, I have had an interest in history, and especially the history of the applied arts. I read lots of fantasy and SF novels (that is a very nice way to spend the many rainy nights we have here in Holland).

I have started studies in art history and philosophy, but never finished those because I got side-tracked when I was drafted for the army in 1992 and came in touch with computers & \TeX .

DW: When and how did you first get involved with \TeX and its friends?

TH: In the military, I was trained as a system administrator for a communications mini-computer that ran Unix System V. Because the actual machine was not a very suitable teaching environment we were encouraged to install Linux on the PS workstations that came with it.

The distribution was one of the earlier Slackwares that came on a few dozen floppy disks, with a sub-1.0 linux kernel, and a bunch of the floppy disks were labelled “n \TeX ”. I was immediately hooked, mostly because of the ‘arts and crafts’ feel of \TeX compared to the ‘industrialized’ approach of line-printers and telexes that were prevalent.

When I was released from the army, I discovered that thanks to changed laws, I could not financially afford to finish my studies, so I searched for a job, and soon found one as a \LaTeX expert for Kluwer Academic Publishers.

DW: How “expert” were you with computers and \TeX by the time you got out of the army and went to work for Kluwer?

TH: The nice thing about being drafted for the Dutch army at that time was that there

wasn't much work to do, leaving lots of time for study. After two years of doing that, I was pretty good with PCs (MS-DOS) as well as Unix systems.

However, my knowledge of \TeX and friends was basically limited to the plain and manmac macros. The first document I have ever compiled was Michael Doob's 'A Gentle introduction to \TeX '. I was definitely not a ' \TeX wizard' when I started working for Kluwer, just a more or less competent ' \TeX nician'.

DW: Do you still work for Kluwer, and what did you or do you do at Kluwer — are you still a “(L) \TeX expert” or do you do something else for work now?

TH: Working at the prepress department of a large publisher proved to be a very fast way to improve my \TeX knowledge. In my first year at Kluwer, I wore out one copy of *The \TeX book* and one copy of Lamport's book completely. The job was two-sided: provide support for the \TeX -based authors, and help the internal \LaTeX typesetters with the fine details of typesetting documents according to the typesetters' instructions.

This meant creating a modularized $\LaTeX 2_{\epsilon}$ class file to replace the two dozen or so separate 2.09 styles, and the compilation, distribution and maintenance of an internal \TeX distribution based on $\em\TeX$ so that all typesetters were guaranteed to use the same \TeX macro and font files.

Later on, Kluwer's interest shifted away from \TeX and toward SGML documents and workflows. I worked roughly two years on the development of the SGML article DTD and a backend system that could typeset those SGML documents — using \TeX , of course. These SGML-related activities took up almost all of my office hours, but my \TeX support work continued on a freelance basis. I had a private company for that: it was called Bittex and it did typesetting, macro programming, and Metafont developments.

In 2000, I left Kluwer and abandoned Bittex to join a three-person company that my brother-in-law had started. The company is called 'Elvenkind', and I still work there today. We focus mostly on database text format conversions and creating web applications to access that data, but we also still do a small amount of style file (I should say class file) design and even a some actual typesetting. Currently, I am working nearly full-time on the successor of $\pdf\TeX$, thanks to a grant from Colorado State University.

DW: I saw your name listed on the announcement of the upcoming \ConTeXt users meeting; please tell me how you became involved and about your involvement with \ConTeXt and, I assume, Hans Hagen.

TH: I ran into \ConTeXt first when I was working on that SGML backend system for Kluwer, around 1996. For that, I needed a macro toolkit that was more reliable and predictable than \LaTeX could offer at that time. As it happens, Hans Hagen had just given some lectures about \ConTeXt during one of the Dutch Language \TeX User Group (NTG) meetings.

\ConTeXt was still commercial software then, so we invited Hans over to the Kluwer office for talks about licensing and eventually reached an agreement. So, Kluwer ended up being one of the very first users of \ConTeXt , and I have been in touch with Hans ever since.

DW: But that doesn't describe your current involvement with the world of \ConTeXt .

TH: I have to go back in history a bit first. In the beginning, \ConTeXt was lacking in a few areas that were a must-have for academic publishing. Most notable of those were mathematics and bibliographic referencing, so I wrote the initial support for that. Also, Kluwer needed an SGML parser on top of the core macros, so I developed one. (That SGML parser is not the one used in \ConTeXt right now, by the way; the current code is an

independent development by Hans himself.)

All of that increased my knowledge of low-level ConTeXt enormously, and being one of the first users of ConTeXt outside of Hans's company Pragma ADE quickly turned me into a 'resident T_EXnician' on the ConTeXt mailing list. I've been answering quite a lot of questions there over the past years.

Right now, I manage a few of the details related to releasing new versions: I compose the release notes for every official released version of ConTeXt, I run a mirror of Pragma ADE's web site (<http://context.aanhet.net>), I take care of uploading the distribution to CTAN, and I am the maintainer of the 'museum', a repository of more than a hundred old versions dating back almost a decade: <http://foundry.supelec.fr/projects/contextrev>.

DW: Please tell me how you became involved and about your involvement with MetaPost and your ambitions for that project.

TH: You may be aware that ConTeXt uses MetaPost as an in-line drawing engine for T_EX. Hans being Hans, he always has feature requests for every tool he uses, and MetaPost was no exception. Bogusław Jackowski, also a heavy MetaPost user, also had some wishes. Somehow they got together, and because Hans knew I had some experience in WEB programming, he dragged me into it, mostly to verify the feasibility of some of their requests.

As it turned out, John Hobby himself was no longer interested in doing development on MetaPost, and he generously agreed with a group from the T_EX user groups taking over. Hans and Boguslaw do quality assurance and testing, Karl Berry takes care of the web site, Troy Henderson has just volunteered to keep the manual up-to-date (he takes over from Karl), there is a mailing list with people contributing requests and patches, and I do actual web coding, release new versions, and travel around giving talks and gathering bugs and feature requests.

I do not have much of a mathematical background, so my personal goals for MetaPost are mostly related to software engineering. Like, I want to make it possible that MetaPost be built as a library that can easily be embedded within pdfT_EX. And we have a pending patch from Guiseppe Bilotta that increases the precision of MetaPost's internal calculations quite a lot, but it needs quite some extra web source code that is not written yet. It would also be nice if one day MetaPost could output OpenType fonts directly.

All of these are fairly modest goals; larger projects will have to wait until later. Hopefully, a steady stream of news will convince some people to give MetaPost a try, and then perhaps the core group will grow enough so that we can tackle larger tasks like three-dimensional drawing or real world modelling.

DW: You mentioned earlier that you are now working on a successor to pdfT_EX. Scouting around a bit, I find that you are (also?) involved in supporting OpenType fonts in pdfT_EX, providing better support for Arabic typesetting in T_EX, and with LuaT_EX. Are these different projects really one and the same or, if not, how do these various projects (and any others that may be related) fit together? Also, does this mean you are working closely with people like Hàn Thế Thành and Martin Schröder? I guess my meta question is, "Are you in fact working (with others?) on a plan to create a consolidated successor to several separate projects that have existed in the past and thus, to some extent, provide the long awaited production-quality successor to T_EX?"

TH: All those T_EX-related projects are facets of a single big project: making a worthy successor for pdfT_EX. pdfT_EX is lacking in some areas; for instance, it does not have the multi-lingual functionality of Aleph or X_YT_EX, it does not handle OpenType fonts well,

is hard to extend, and it still has many hard-coded limits. This is a team effort, with everybody from the core pdfTeX team contributing. The big difference between me and the rest of the team is that I am now working on it almost full-time.

Thanks to a grant from Colorado State University I can spend a large amount of my office hours working on improving the Arabic typesetting capabilities of pdfTeX, adding Unicode and OpenType font support and assimilating bits of Aleph into pdfTeX—or rather, into LuaTeX.

The LuaTeX project was started a while back as a means to extend pdfTeX in an extensible way, by adding the Lua script interpreter to the executable and giving Lua scripts access to the internals of the typesetting engine. Also, there are plans to add MetaPost to the executable as an embedded library.

Eventually, all of this work will result in a completely new program called MetaTeX. This will be the successor of the current pdfTeX (and probably Aleph), and the final non-MetaTeX pdfTeX will become frozen and only updated for bug maintenance, for people that do not want to move forward with us. We hope to have this all done by the summer of next year.

DW: At the recent PracTeX Conference in New Jersey, I saw Jonathan Kew demonstrate using the existing fonts within his operating system with XeTeX—no big process of converting to the TeX internal formats for font descriptions. Is what you are doing with pdfTeX going to result in a system that will make working with fonts that easy, or will it still require the conversion to the traditional TeX formats for fonts? Also, is there communication between what you are doing with pdfTeX and Jonathan's work on XeTeX? Is some further consolidation there likely or possible?

TH: The goal is that for simple use of a font, like in XeTeX, OpenType fonts can be used as they are, without any conversion. For harder stuff, like hanging punctuation, pseudo-hz font expansion, and top-quality Arabic typesetting, extra information is needed that is normally not included in the font. Font loading and instantiation will be under the complete control of the Lua script language, so it will be possible to add that extra information using augmentation files instead of actual font conversion.

There is some communication between us and XeTeX, but the two projects have very different approaches to the problems of typesetting engines, so neither of us is working towards a merge right now.

pdfTeX aims for the absolute best quality and as much configurability as possible. Therefore, it does all typesetting tasks, everything itself, often at the expense of more source code and therefore longer development cycles. XeTeX aims for the highest ease of use for users and economy of implementation. Therefore, it uses external libraries for some of its tasks. The result is that XeTeX is not able to do micro-typography like pdfTeX, and pdfTeX is way behind XeTeX when it comes to Oriental script support.

Perhaps the projects will grow closer together in the future, but I am not aware of any plans in that direction that are active right now.

DW: Scouting around, I also find your work on the koeieletters and koeielogos fonts and your conversion of the Metafont logo font and a couple of the other miscellaneous original TeX system fonts into Type 1. Can you say a word about the motivations for these?

TH: The older fonts I needed for internal use at Kluwer. They were converted using Richard Kinch's `metafog`. In fact, I was also commissioned to create an extended math font set for Kluwer, using the same production process. That font set never got to produc-

tion quality, but I still have a thousand plus Metafont glyphs in an archive somewhere.

I feel fonts are a very interesting subject, so I've been playing with them now and again for quite some time. The koeieletters font is the most fun (and funniest) thing I've worked on in years. Because of their graphical nature, fonts are closer to art than to programming, and I would love to earn my money creating fonts. Sadly, that does not seem feasible, so I am stuck with T_EX and web application programming for now.

DW: Your original interest in art history and philosophy seems pretty distant from the set of skills you need for the work you are doing today with T_EX: Pascal, WEB, C, T_EX, typesetting, programming languages, PDF, etc. Or is there some connection with your earlier interests or some different history and philosophy with which you are now enthralled?

TH: Programming itself is quite distant. It turned out I am halfway decent in it, and it pays the bills. Luckily, working on and with T_EX and MetaPost is closer to the arts than most other IT jobs. After all, the goal is to create something that is as beautiful as possible. When I see a medieval illuminated manuscript, I am always wondering how nice it would be if we could make T_EX produce documents that are just as pretty.

DW: This has been fascinating, Taco. Let me ask you a couple of somewhat philosophical questions and then we will be done. First, a question I ask of most of my interviewees: What is your view of TUG and the other user groups and how they can best, and practically, continue to support T_EX and T_EX users and perhaps push the world of T_EX ahead?

TH: The way I see it, the most important task of the user groups today is facilitating developments that are taking place, stimulating projects where needed. T_EX is free software and the active user base is fairly small, and as result there is not much of an external drive for new developments. Forward movement has to come from within our own community, and the user groups have a very important role in that community. Since this is pretty much what the user groups are already doing, I can only say: you are doing a fine job, keep at it!

DW: How does open-source/"free" software fit into your view of the world? You are developing a lot of software and you say it is "paying the bills", but . . .

TH: This is a tricky question. It is very easy to slip into a very long monologue about the pros and cons of communism versus capitalism, and I really do not want to do that. I prefer to think of programming as providing a service, as opposed to the production of goods. Let's leave it at that.

DW: Thank you very much for taking the time to participate in this interview. Hearing about all the work you are doing has been inspiring to me.