

Werner Lemberg

Werner Lemberg developed the CJK (Chinese, Japanese, Korean) package for L^AT_EX and has been involved in a number of other computer-based typesetting projects.

[Interview completed 28 April 2006.]



Dave Walden, interviewer: Please tell me about your personal history independent of your involvement with computer based typesetting, etc.

Werner Lemberg, interviewee: I'm Viennese. I went to school in Vienna and studied music at the Hochschule für Musik (now Vienna University for Music) — I finished with five diplomas (composition, conducting, master class for piano, chorus conducting, and singers' coaching). At the same time I started studying mathematics and Chinese, both of which I had to abandon two years later because of lack of time. Since both my parents and brothers are musicians it was quite natural for me to do the same, and I started at my parents' working place, the Vienna State Opera. My parents are now retired, but my brother still works there, being a member of the Vienna Philharmonic Orchestra.

In the following years I worked at various theatres in Austria and Germany; since five years ago I have been engaged at the Municipal Theatre in Koblenz, Germany — the town where the Mosel flows into the Rhine — as a conductor head of the music department (the exact term is “Studienleiter”).

DW: How and when did you first get involved in computer-based typesetting?

WL: Well, around 1989 I wanted to write a “Diplomarbeit” (a thesis) about the Chinese mouth organ, the shēng (xx+7B19), and I also wanted to have Chinese characters in the German text written by the computer and not inserted by hand. At that time there were just a few, quite expensive programs for MS-DOS which could do that. And there was T_EX: A good guy from DANTE told me that L^AT_EX could do what I wanted using Thomas Ridgeway's “poor man's Chinese” (pmC) package. This was the start of everything. By the way, this master's thesis was never written, while the CJK package for L^AT_EX actually came into existence . . .

DW: You said you studied Chinese for a couple of years. Was that enough for you speak or write Chinese?

WL: Not really :-). I now have an idea of the language and the script, and my colloquial Chinese is sufficient for daily life, but anything more specialized makes me desperate. My written Chinese is not sufficient for any purpose.

DW: Can you say a bit about the difficulty of developing these fonts; did you design the characters from scratch, using some kind of software?

WL: I didn't develop any font! My package just provides the necessary infrastructure to use the CJK fonts with L^AT_EX.

DW: What is the current status of the CJK package and your involvement with it?

WL: Basically, it is stable. I plan a new version in the near future that will extend the Unicode support beyond the BMP (Basic Multilingual Plane, covering the character range U+0000 to U+FFFF) — the stuff is already in the development repository.

DW: So your first experience in computer-based typesetting was with L^AT_EX, but today you are a lead contributor to Groff. How did that come about?

WL: This happened by pure coincidence. If I remember correctly, I reported a Groff problem, and Ted Harding and I got into some public discussion, and finally we were asked whether if we were interested in further maintaining Groff because it was orphaned. Today I'm still doing this... It's quite fun to compare the different paradigms of Groff and T_EX: Some problems are much more easier to handle with the one program and vice versa.

DW: By easier to “handle problems”, I assume you mean as a user rather than developer, correct? Will you please give me examples of the type of thing that is easier to handle in Groff and in T_EX?

WL: Groff has the concept of “traps” to catch vertical events. For example, it's rather simple to make the first line of a paragraph be printed in small caps, something which is quite difficult in T_EX, as far as I know. Another advantage of Groff is its capability to move easily to absolute positions on a page.

The perhaps biggest disadvantage of Groff is the lack of shrinkable space, both vertically and horizontally (vertically stretchable space is also missing). Thus formatting of columns with a small line width works far better in T_EX. Groff's algorithm to format a paragraph is also much less sophisticated.

DW: I presume you use both L^AT_EX and Groff: which do you use for what and why?

WL: For me it is a must to have good man pages! And this is what I am using Groff for. For longer documents I use L^AT_EX.

DW: What makes Groff better for good man pages?

WL: For me a good man page displays well on various output formats, at least as plain text, PostScript, and HTML. Groff works fine for all those formats (HTML output is still somewhat experimental, though), while support for (formatted) plain text output is completely missing from T_EX.

DW: At the (troff.org) web site, it said you did the style files for ESSCIRC and ESSDERC, which have to do with solid-state electronics. Do you work with solid-state electronics, too, as well as with music and computer-based typesetting?

WL: No, not at all. Developing those two style files was a paid job.

DW: You've also written c2cweb. What is the purpose and reason for that?

WL: This program is a kind of preprocessor to cweb. It takes C source and header files and pretty-prints them, together with handy variable and function indices. Unfortunately, I don't have time to maintain and improve it, and it seems that other, similar

programs do a better job.

For perfect C (and C++) source code formatting a complete language parser and preprocessor is needed, I think. I can imagine that someone adds a special output mode to `gcc` which doesn't emit compiled code but language tokens (comments, identifiers, operators, etc.) which can then be printed out.

DW: Is it correct to assume that you yourself sometimes use computers to typeset music? What program(s) do you use? Also, please tell me your view of the current state and future of music typesetting in comparison with the state of typesetting other things that can be done, for instance, with \LaTeX .

WL: Indeed, I not only use GNU LilyPond for typesetting my music, I also actively participate in its development. It works very similarly to \LaTeX ; that is, you write a text file which is then compiled to get a PostScript (and consequently a PDF) document. Personally, I like this most since I try to avoid the mouse as much as possible. Up to now I know only a single program, Score (and its recent descendants), which produces output comparable to manually engraved documents. There are still a lot of things to do in LilyPond (volunteers welcome), but the future looks quite promising.

DW: Can you show me an example of LilyPond input markup, e.g., for a measure or two of a tune?

WL: Here it is. The following input

```
\relative c' { g'4 e e2 | f4 d d2 | c4 d e f | g g g2 }
```

gives the following output which I believe is almost self-explanatory:



DW: You are a pianist, so I might assume there is a problem with keyboard input as at least the preliminary input to a music typesetting program. Will you please sketch the set of issues that leads you to do input via a text file rather than either a mouse or from a musical instrument with digital output?

WL: I don't object to inputting data with the mouse or with a musical instrument—the latter is definitely the natural input method for most of Western music. The Score program offers both, and I've indeed used those methods (you can do that with LilyPond too, using a front-end, but I've never tried it). Interestingly, my experience has shown that neither mouse nor musical instrument input is actually faster than input with a computer keyboard if you need more than the most basic music data. The movement forth and back between the computer keyboard and the mouse or the musical instrument outweighs any time savings you would have by using the “natural” input method. Of course, this holds for a trained user only.

My particular reason for using a keyboard-only solution is that I do most of my work on a laptop sitting in a train, and using the laptop mousepad for serious mouse-based input is perverse in my mind. While I use GNU/Linux exclusively, I'm quite happy that Microsoft has forced the hardware manufacturers to add the Windows and Menu keys to the keyboard. This gives me free Super and Hyper modifiers in addition to Shift, Alt, and Control which I use for switching windows and virtual screens :-).

DW: You mentioned the Score program. What is different about the Score approach from that of LilyPond that allows Score to produce better output?

WL: Score is an old MS-DOS based program which was written by Leland Smith, then a professor at Stanford (my memory is very vague about this; it might be a different university). It is quite expensive (about \$1000 and more, I think), but it produces marvelous output. This is old information since I haven't used it for years, and I haven't seen the port of this program to Windows. Big music publishing houses like Schott (Mainz, Germany) and Universal Edition (Vienna) use or have used it, and the results are quite convincing if you look at scores from contemporary composers.

LilyPond is still under development, and it can actually produce even better output than Score. It tries to mimic manually engraved music (based on German tradition) as closely as possible. The amazing thing about LilyPond is that the number of tweaks necessary to get good looking output is very small under normal circumstances, far less than are necessary for Score—it can deal with most of the object collisions. Similar to \TeX , LilyPond's default formatting capabilities are excellent.

DW: What is an object collision?

WL: When two musical objects like a note head and the “f” sign (which indicates “forte”) overlap or nearly overlap.

DW: LilyPond used to use \TeX , but doesn't any more. Why did the project switch away from \TeX ?

WL: There were several reasons:

1. LilyPond needs much control over the layout process; especially, it needs to know the font metrics. The development process has revealed that delaying text layout to \TeX causes more problems than it solves. (There still exists a \TeX back end which provides a two-step implementation: LilyPond first emits a file containing all the strings \TeX should render. Then you have to run \TeX to create a log file which holds the strings' dimensions (width, height, and depth). A second run of LilyPond reads in this log file and does the final typesetting. However, this might not work currently because there isn't much interest in maintaining it.)
2. A logical consequence of point 1 is to directly access a rendering engine (Pango). This decision gave us three additional benefits: (a) support for complex scripts, (b) support for complex fonts, and (c) native UTF-8 input encoding.
3. LilyPond's music fonts can now hold any number of glyphs since the limitation of 255 glyphs per font is gone.
4. Depending on a \TeX distribution was a pain, especially since LilyPond runs on both Windows and Macs. LilyPond itself is already a big package; not relying on \TeX both removes a package dependency and reduces the necessary downloads for potential Windows and Mac users—most Windows and Mac boxes don't have a pre-installed \TeX system.

DW: It sounds like you are pretty deeply into computer programming and you said you were two years into a math program before you had to give it up for lack of time. Having a person from arts and music be deeply interested in computers and math seems sort of surprising at first blush. On the other hand, in my professional world of computers and math, many people are amateur musicians. Is the reverse true? Do you find many people in your world of professional music who are amateur computer and math people, or do

you see yourself as an exception?

WL: Honestly, I don't know. The famous example is William Herschel, discoverer of Uranus, who was both a composer and astronomer. The fact is that I don't think that any of my colleagues here in the theatre do something similar to me. But there are a lot of computer freaks actively participating on the LilyPond mailing list who are professional musicians and composers — since LilyPond uses Scheme as its extension language a certain computer background doesn't hurt . . .

DW: Thank you, Werner, for taking the time to participate in this interview. I found it fascinating.