MkII MkIV LMTX

Where does it end?

Hans Hagen

The Covid situation made this meeting special. It forced us to adapt and consider how we should deal with these kind of situations. Despite the difficulties, we had a very nice meeting as usual.
The first talk gave a summary of the evolution of ConTEXt from its inception to today. The following talks covered more specialized topics.

MkII

In the 1980's I bought the TEXbook, but it was pretty abstract to me.

In the beginning of the 1990's when we had to get some math on paper, we bought(!) a copy of LATEX.

Right from the start we had to make it look better than other out-of-the-box solutions so we developed a shell around it. Soon, however, we scrapped this and started from scratch.

First we built on top of LAMS-TEX, then we switched to inrsTEX. After learning from trial and error, we soon found that we only needed a few of these components.

We joined the ntg, met Taco and friends, and slowly got some attention.

And it all went on till we had what we later called MkII. But we always had ideas about what more we wanted.

We went from TEX to ε-TEX to pdfTEX to pdfeTEX. We played with the idea of `eetex`, different backends etc.

Mk . .

ConTEXt has been keyword-driven and class-based from the start. This came with a performance hit, so the reputation was that it was slow: Features like inheritance, flexibility and user control all come at a price.

ConTEXt always had an abstract driver model (dvips, dvipsone, dviwindo, Acrobat, pdfTEX etc). It also had an adaptive front-end so we could support successive engines: TEX, ε-TEX, pdfTEX, Aleph, XETEX.

There had to be color and graphics support from the beginning.

The interfaces permitted extensions without breaking compatibility. The user interface was multilingual; we started building it for Dutch and German users.

It came with management tools (e. g. TEXexec, TEXutil, TEXfont, texmfstart etc.) for job control, dealing with (user) fonts, image manipulations etc. And of course MetaPost, XML, combining font setups, mixing encodings, and UTF patterns evolved with the system. Educational usage was often the reason for adding new features.

MkIV

At some point we started playing with Lua (in SciTE). And then Hartmut started adding some basic Lua support to a clone of pdfTEX that soon became LuaTEX. Next the Oriental TEX project provided means for Taco to transition to C.

And for years we slowly built up the system. A LuajitTEX version showed up and Luigi took over integration in TEXLive (like compilation within the infrastructure and updating libraries).

In parallel we tested features and explored what we needed with ConTEXt: MkIV evolved. And soon, further development in MkII was frozen, and all new development occurred in MkIV.

Wolfgang checked and completed all setups while we upgraded the interface subsystem. Obsolete mechanisms (font, language, input) were removed, and a lot more happened: some more TEX, lots of Lua, better MetaPost integration, and more advanced XML.

To some extent a project like this came too late because the glory days of TEX had already passed. Publishing had changed but just as with pdfTEX, we felt a conceptual upgrade was needed.

MkXL (aka LMTX)

When we had to stabilize and freeze LuaTEX for the LATEX community, we switched our focus to LuaMetaTEX. Its name reflects the importance of each core component. The idea was to create a lean-and-mean engine that would become very stable and not depend on the issues of the day.

It's for all those dedicated users who like quality and playing around but also want guarantees that the tools keep working years from now: it's about independence.

Of course we tested and explored with ConTEXt, and this time LMTX evolved. Here the X reflects that we consider XML to be part of the picture.

Although there will be (and already is) new functionality, the changes will be less dramatic because this time we don't have the change in font, encoding and regime subsystems (that made some MkII commands go away).

Hopefully, we can now improve some of the more tricky (hard-to-do-in-good-old-TEX) mechanisms.

And at some point we will freeze MkIV and development will happen in LMTX only.

This meeting

My talks in this meeting were mostly about LuaMetaTEX and the ConTEXt version LMTX that targets it: How it is done, which concepts show up, and where we want to go.

You will be surprised by the plenty of new features relating to LuaMetaTEX unless you have kept a close eye on last year's developments. There is more to tell, but most of that is already known from the previous meetings.

And, as usual, a ConTEXt meeting is not only a deadline, but also a starting point. It's you who keep it all going. And, even more than that, it is about us meeting.