

## Abstracts without papers

### Fundamental

*Hans Hagen*

#### parshapes

Paragraph shapes are powerful on the one hand but limited on the other. In LMTX we can now use a high level interface that also picks up where a par break destroys the specified shape. This presentation will discuss the current implementation but users can come up with additional demands.

#### hyphenation

We have more control over hyphenation. Although this is work in progress, it is still good to know what can be done already, in ConT<sub>E</sub>Xt as well as in the engine. Better handling of compound words in combinations with ligature prevention is an example.

#### localcontrol

The local control mechanisms permits sidestepping the so called main loop which makes it possible to make macros that behave like vardef in Metapost. Although one can end up in confusing situations due to the inner workings of T<sub>E</sub>X with respect to scanning and expansion, it has some potential.

### Handy

*Hans Hagen*

#### box lines

Boxes are the building blocks of any T<sub>E</sub>X engine. Although ConT<sub>E</sub>Xt hides most of that behind commands and constructors like `\framed`, it makes sense to know a bit what can be done at a lower level. Box lines are an example of an interface that sits between high level commands and primitives. In this talk I will discuss the low level ones as well as box lines.

#### scope

The difference between global and local (grouped) in T<sub>E</sub>X is rather fundamental. Often counters are defined global but in LMTX we can also temporarily define local ones. From its ancestors we inherit pushing and popping. Doing things after a group has been extended and we can also do things at a par break. I'll try to cover this here.

#### matching

Occasionally a new (low level) command is introduced that doesn't fall into the often used category but is still interesting because of how it is implemented. Running over text using a match is an example of this. We can discuss wishes for similar additions.

### Math

*Hans Hagen*

#### getting there

In order to deal with the inconsistencies between math fonts as well as user demands, there has been some extensions to the LuaMetaT<sub>E</sub>X math engine. This year nearly all hard wired parameters have been made configurable and more detailed control over the rendering has been added. I will show some of that. A few new tricks might be discussed as well.

### LMTX

#### where we stand

This is the usual update on LuaMetaT<sub>E</sub>X and LMTX. It takes a while to get where I want to go, but most loose ends are tight up. I will mention what has been done with the T<sub>E</sub>X engine, the Metapost machinery and the Lua subsystem (including libraries). I will show the impact of

the new argument scanners on the ConT<sub>E</sub>Xt code base. I will also spend some words on memory management.

### **overload protection**

This is also a discussion. I will show what overload protection means and how it is implemented in T<sub>E</sub>X and Metapost. I will also show how it can be kept consistent. The question is to what extent protection has to be enabled by default (there are levels).

### **Fonts**

*Hans Hagen*

#### **compact mode**

Compact fonts have been mentioned on the mailing list but what are they? I will show the difference with the traditional font mechanism and try to convince users that it might be a good default. I will explain some details of the impact it has on text as well as math, at the T<sub>E</sub>X end as well as in the engine.

#### **missing glyphs**

This is a short summary of how we (can) deal with missing characters: visualization as well as quick and dirty drop-ins.

#### **\definefontfamily (previously known as simplefonts)**

*Wolfgang Schuster*

How you can use `\definefontfamily` to simplify your font setup.

#### **An Overview of Language Support in ConT<sub>E</sub>Xt**

*Tomáš Hála, Tamara Kocurová, Adriana Kašparová et al.*

This lecture summarises the long-term work of the members of the Club of Modern Computer Typesetting Technologies at Mendel University in Brno (CZ), which resulted in a publication providing an overview of the current state of language support in ConT<sub>E</sub>Xt.

### **Diary**

*Tomáš Hála*

At the beginning of the year, one needs to buy a diary. But how to get it when the stationery was closed on the basis of government regulations? This lecture describes how to prepare your own diary in ConT<sub>E</sub>Xt (and with Lua).

### **The PDF mess**

*Hans Hagen*

Or “What to do with the somewhat useless and/or unreliable PDF features?” Buying competitors, embracing some technique and then ditching it has become popular with large companies. The same is true for PDF. Native simple support for video and audio, something that browsers can do easily, was replaced by a mechanism depending in Flash that now has been dropped so we’re now stuck with some complex media subsystem. Over decades support for forms, attachments, notes, and media demanded adaptation to viewers. Where do we go? And why don’t open source viewers not catch up? (Maybe we should compare a few.)

### **Embedding interactive elements and multimedia in PDF files**

*Michal Vlasák (remote presenter)*

Recently, as part of my bachelor thesis, I looked into the state of multimedia (audio, video, 3D) and other relatively obscure PDF features, with the connection to T<sub>E</sub>X.

### **PDF tools on Linux**

*Henning Hraban Ramm*

There is no Adobe Acrobat for Linux, but there are free options as well as a few commercial solutions.

## MetaPost

*Hans Hagen*

### scanners and injectors

We can extend the Metapost subsystem with commands that look much like primitives by using scanners. The communication between T<sub>E</sub>X and Metapost as well as between Metapost and Lua is influenced by this feature. Pushing back something to Metapost has also been improved. Both are responsible for recent low level changes in Metafun.

### local positions

This is more a demonstration of a relative new variant on positioning information as for instance used for anchoring Metapost graphics. Users are invited to come up with ideas for usage.

### A cross-stitch pattern (de)construction

*Taco Hoekwater*

The road-trip I went through to make a cross-stitch pattern do what I want it to do, which involved breaking a PDF down to tiny pieces and then constructing a wholly new PDF.

### Rendering OpenStreetMaps

*Hans Hagen*

Triggered by a question on the mailing list I decided to see if we can use Metapost to render openstreetmaps. It is indeed possible but not without taking some aspects into account. What colors to use? What is the right stacking order? How to deal with inconsistencies (a side effect of arbitrary contributions)? How to deal with a somewhat bloated encoding (the no way back problem)? How do formats like that evolve (when do bugs become features)? This all makes an interesting discussion.

## Wiki updates

*Taco Hoekwater*

What has happened on the wiki over the past year?

### Reading a mail log, or: LPEG for intermediate T<sub>E</sub>X users

*Taco Hoekwater*

I use Lua(T<sub>E</sub>X) to parse and display information from server mail logs.

### Aha! and Gotchas – Our statement migration journey from proprietary software to ConT<sub>E</sub>Xt and other Linux tools (GNU Parallel and mutool)

*Ramkumar KB (remote presenter)*

I work in a large global financial organisation in Singapore and we are currently evaluating using ConT<sub>E</sub>Xt for our PDF Statement Generation system. This was largely possible due to the new Statistical Charts module becoming available in 2020. Mostly we are in the last stages - mainly in batch mode - where data systems generate 20,000 data inputs (JSON / XML) and the statement system needs to spit out 20,000 PDF statements - we are trying out different options (including containers), as we need this PDF generation to be reasonably fast.