TUG 2018 abstracts

Editor's note: Videos are available for nearly all of the talks; links and other information at `https://tug.org/tug2018/program.html`.

– – ∗ – –

Doris Behrendt
The General Data Protection Regulation (GDPR) in the European Union

On 25 May 2018 the GDPR was applied in the EU. In my position as treasurer of the German TEX user group DANTE e.V. I studied this regulation from the DANTE perspective and will talk about some aspects of this regulation, which are concerning us.

As some of you probably know, a lot of Europeans — including myself — are somewhat delicate about data processing and privacy. While the industry complains about the GDPR being a monster of bureaucracy, there are also some quite interesting legal bearings that come with it, e.g., it will also apply "to the processing of personal data of data subjects who are in the Union by a controller or processor not established in the Union, where the processing activities are related to ... the offering of ... services, irrespective of whether a payment of the data subject is required, to such data subjects in the Union ...".

This should be interesting especially to companies that are not based in the EU but are handling data of EU citizens, and by GDPR Article 83 (5) not complying could become expensive: "Infringements ... shall ... be subject to administrative fines up to 20,000,000 EUR, or in the case of an undertaking, up to 4% of the total worldwide annual turnover of the preceding financial year, whichever is higher ...".

You can imagine that this could become very interesting when the next Facebook or similar data scandal comes up.

S. Coriasco, D. Ahmetovic, T. Armano, C. Bernareggi, M. Berra, A. Capietto, N. Murru, A. Ruighi, E. Taranto
An automated method based on LATEX for the realization of accessible PDF documents containing formulae

Mathematical formulae contained in PDF documents generated using LATEX are usually not accessible with assistive technologies for visually impaired people, such as screen readers and braille displays.

To address this issue, we developed *Axessibility*, a LATEX package which allows creation of PDF documents in which the formulae can be read by these assistive technologies. *Axessibility* automatically generates hidden comments inside PDF documents corresponding to each formula (by means of the `/ActualText` PDF attribute). This actual text contains the LATEX code of the formula, and it is read by screen readers (JAWS, NVDA and VoiceOver). Moreover, we have created NVDA and JAWS dictionaries (in English and in Italian) that provide natural language reading for users that do not know LATEX.

While this package enables accessibility of mathematical formulae contained in PDF documents, it does not generate PDF/UA compatible documents.

Joachim Heinze
The unchanged changing world of mathematical publishing

1. A very short overview of the history of mathematical publishing with some Springer examples is given. *Numerische Mathematik* was the first of all SpringerNature journals ever, over all disciplines, to go online in 1994.

2. The change of the world of publishing: generating (scientists), composing (publishers and scientists) and disseminating (librarians and publishers) mathematical content in electronically form. TEX and "online visibility" are the buzzwords here.

3. Open access for all mathematical content? "New" initiatives like "Overlay Journals", based on arXiv, are briefly discussed, as well as the more recent Sci-Hub and ResearchGate initiatives.

4. Keep track of what has been published and cited. MathSciNet and zbMATH, the two big math review journals, in comparison to other initiatives, like Google Scholar, Scopus, and Web of Science. A new initative from China? MathSciDoc.

5. Recent developments in the dissemination of scientific information are discussed. Social media (Scholarly Collaboration Networks (SCN)) in scientific communication and some new initiatives such as "Sharedit" and "SciGraph" are briefly reflected upon. Artificial intelligence and some hope for the future will close the presentation.

Tom Hejda
`yoin` — Yet another package for automation of journal typesetting

A new LATEX package will be presented that allows combining journal, conference and similar papers into issues. The most important premises the package are built upon are (1) the papers themselves are independent documents to the extent that even different compilers can be used for different papers, and (2) the papers' page numbering is automated and there are tools for communicating metadata between the whole issue and the papers.

Please note that a preliminary version of the package will be presented and help from the community will very likely be sought at the conference.

Mico Loretan
Selective ligature suppression with the `selnolig` *package*
TEX has long provided straightforward methods for creating typographic ligatures. Until recently, though, suppressing inappropriate ligatures selectively could only be achieved by applying mark-up by hand to a document. `selnolig`, a LuaLATEX package, provides the machinery to perform selective ligature suppression in an automated way that requires minimal user involvement. The package also provides sets of ligature suppression rules for English and German language documents. The talk provides an overview of the package's design philosophy and main features, discusses some of its current limitations, and gives the outlook for further developments.

Frank Mittelbach
A quarter century of `doc`
In this talk I will re-examine my poor attempts at Literate Programming and how they have shaped (for better or worse) the LATEX world in the past decades.

It's about time to rethink some of the concepts invented back then — but can we still evolve?

Ross Moore
Authoring accessible 'Tagged PDF' documents using LATEX
Several ISO standards have emerged for what should be contained in PDF documents, to support applications such as 'archivability' (PDF/A) and 'accessibility' (PDF/UA). These involve the concept of 'tagging', both of content and structure, so that smart reader/browser-like software can adjust the view presented to a human reader, perhaps afflicted with some physical disability. In this talk we will look at a range of documents which are fully conformant with these modern standards, mostly containing at least some mathematical content, created directly in LATEX. The examples are available on the author's website, `web.science.mq.edu.au/~ross/TaggedPDF`.

The desirability of producing documents this way will be discussed, along with aspects of how much extra work is required of authors. Also on the above website, and published elsewhere in this issue (pp. 131–135), is a 'five-year plan' on how to modify the production of LATEX-based scientific publications to adopt such methods. This will involve cooperation between academic publishers and a TUG working group.

[Editor's note: Since the talk worked mostly from examples, showing non-printing aspects of what can be stored in, and extracted from PDF files, the printed description is not entirely sufficient; see the video at `youtube.com/watch?v=mPBtkCsChJw`.]

Eduardo Ochs
Dednat6: An extensible (semi-)preprocessor for LuaLATEX that understands diagrams in ASCII art
(LA)TEX treats lines starting with `%` as comments, and ignores them. This means that we can put anything we want in these `%` lines, even code to be processed by other programs besides TEX.

In this talk we describe a "semi-preprocessor", called `dednat6`, that makes blocks of lines starting with `%L` be executed as Lua code, treats blocks of lines starting with `%:` as 2D representations of derivation trees, and treats blocks of lines starting with `%D` as diagrams in which a 2D representation specifies where the nodes are to be placed and a stack-based language inspired by Forth is used to connect these nodes with arrows.

A predecessor of `dednat6`, called `dednat4`, was a preprocessor in the more usual sense: running `dednat4.lua foo.tex` on a shell would convert the trees and diagrams in `%:-` and `%D`-blocks in `foo.tex` to `\def`s that LATEX can understand, and would put these `\def`s in a file `foo.dnt`; we had to put in `foo.tex` an `\input "foo.dnt"` that would load those definitions.

Dednat6 does something almost equivalent to that, but using LuaLATEX to avoid the needs for an external preprocessor and for an auxiliary `.dnt` file. Here is how; the workflow is unusual, so let's see it in detail.

Put a line
```
\directlua{dofile("loaddednat6.lua")}
```
in a file `bar.tex`. When we run "`lualatex bar.tex`" that line loads the `dednat6` library, initializes the global variable `tf` in the Lua interpreter with a `TexFile` object, and sets `tf.nline=1` to indicate that nothing in `bar.tex` has been processed with Dednat6 yet.

A (low-level) command like
```
\directlua{processlines(200, 300)}
```
in `bar.tex` would "process the lines 200 to 300 in `bar.tex` with `dednat6`", which means to take all the blocks of `%L`-lines, `%:`-lines, and `%D`-lines between the lines 200 to 300 in `bar.tex`, run them in the necessary interpreters, and then send the resulting LATEX code — usually `\def`s — to the `latex` interpreter.

The high-level macro `\pu` runs
```
\directlua(processuntil{tex.inputlineno})
```

which runs `processlines` on the source lines between `tf.nline` and the line where the current `\pu` is, and advances `tf.nline`. That is, it processes with `dednat6` the lines in the current file between the previous `\pu` and the current one.

The strings `%L`, `%:`, and `%D` are called "heads" in `dednat6`, and it's easy to add support for new heads; this can even be done in a `%L` block.

With `dednat4`, all the `\def`s had to be loaded at once; in `dednat6` idioms like `{\pu ...}`, `$\pu ...$`, and `$$\pu ...$$` can be used to make the `\def`s between the last `\pu` and the current one be local.

Boris Veytsman
Stubborn leaders six years later
After six years the journal *Res Philosophica* changed the style of its table of contents. The new design requires the dotted line with the page number to follow the last line of the article title rather than the first one. The old design was described in a *TUGboat* article (33:3, pp. 316–318, 2012, `tug.org/TUGboat/tb33-3/tb105veytsman-leaders.pdf`).

We use this occasion to revisit the old code, discuss the new one and the fact that deceptively similar designs require completely different code.

Boris Veytsman
R+knitr+LaTeX workshop
The work of a research scientist involves keeping daily notebooks. Such a working notebook is a document with text, equations, calculations, figures, tables, code snippets which reflects the current state of the lab research. This workshop teaches how to maintain such notebooks in a TeX/R environment.

Prerequisites: please install the following on your computer — a TeX distribution (preferably either TeX Live or MiKTeX), R, with packages knitr, tikzDevice and Hmisc. For a front end, we can use either Rstudio or Emacs+AUCTeX+ESS.

After you have installed R, you can install knitr, tikzDevice and Hmisc using the R packaging system. Also, Vincent Goulet has constructed convenient Emacs distributions for Windows and Mac systems which include ESS+AUCTeX, available at `vigou3.github.io/emacs-modified-windows`.

Joseph Wright
Fly me to the moon: (La)TeX testing (and more) using Lua
Testing has been important to the LaTeX team since its inception, and over the years a sophisticated set of test files have been created for the kernel. Methods for running the tests have varied over the past quarter-century, following changes in the way the team work.

In recent years, the availability of Lua as a scripting language in all TeX systems has meant it has become the natural choice to support this work. With this as a driver, the team have developed the `l3build` package (`ctan.org/pkg/l3build`) for running tests automatically. Building on the core work, `l3build` has grown to provide a powerful approach to releasing packages (and the LaTeX kernel) reliably.

Here, I'll look at the background of our testing approach, before showing how and why Lua works for us here.

MAPS 48 (2018)

MAPS is the publication of NTG, the Dutch language TeX user group (`http://www.ntg.nl`).