
TUG 2015 abstracts

Editor’s note: Slides and other related information for many of the talks are posted at <http://tug.org/tug2015/program.html>.

— * —

Kaveh Bazargan and Jagath AR

TEX — After 35 years, still the best solution for modern publishing

TEX is around 35 years old. The engine has remained almost unchanged. Since about 1990, the desktop publishing revolution gradually took the focus away from tags or mark-up in text, but the ubiquitous requirement for XML by publishers has focused attention on mark-up yet again, and TEX has returned as the king of automated pagination. We will discuss and demonstrate the advantages of TEX for pagination of XML over other pagination systems, including: fully automated pagination of XML files; highest typographic quality; production of “enhanced” PDFs, not possible by other means; obtaining different PDF styles from a single source.

Julien Cretel

Functional data structures in TEX

Because TEX lacks rich data structures, implementing even simple yet useful algorithms in it can be laborious. However, TEX is, in many ways, remarkably similar to functional programming languages, which are often praised for their expressive power.

Building on Alan Jeffrey’s approach to embedding elements of the lambda calculus in TEX (see Alan’s `lazylist` package), I plan to demonstrate how to implement and use richer data structures (such as binary search trees) in TEX & friends.

Olaf Drümmer

PDF/UA — what it is, how users can benefit from it, and how to get it right

PDF/UA is the latest addition to the group of international PDF standards. Published in 2012, it defines what a tagged PDF — as defined in PDF 1.7 (per ISO 32000-1) — must look like to be considered ‘universally accessible’, and how PDF/UA conforming tools should take advantage of its features. “Accessible” is often thought of as content accessibility from the point of view of people with some disability, but is not nominally limited to that. Content in a PDF/UA conforming file can also be more easily and more meaningfully accessed by software, allowing for intelligent content extraction or flexible repurposing (think formatted text reflow on mobile devices).

This talk gives a very compact overview of the

rules defined in the PDF/UA standard, and how a PDF/UA file typically differs from an ordinary PDF file. Several sample usage scenarios will be demonstrated so attendees can get a feeling for how PDF/UA matters to users who have to rely on PDF/UA conforming documents and on suitable tools. Finally, several challenges will be discussed that document authors tend to run into.

Paul Gessler

Pretty-printing Git commit history graphs with PGF/TikZ

Increasing popularity of the distributed version control system Git has created a desire to integrate its versioning metadata into documents dynamically. An existing package, `gitinfo2`, by Brent Longborough, provides hooks and tools to access this information within L^AT_EX documents. My experimental package `gittree` adds a convenient interface for producing commit history graphs within L^AT_EX documents using the PGF/TikZ graphics language. I will present several examples of `gittree`’s use and discuss continuing development efforts.

Hans Hagen

What if ...

What TEX does and what TEX doesn’t do, the way a macro package is set up, how users use a TEX-like system, and what they expect (or demand) to a large extent depends on the circumstances in which the system was developed. In that respect a macro package is a snapshot of how at a certain moment texts ended up on the media popular at that time. How would TEX stand in a future museum? What if the developments around computer technology, the ideas about communication, the expectations of users and commerce had been slightly different? What if ...

Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski

All the characters we need

We will discuss the choice of characters for math fonts.

Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski

Six GUST e-foundry math fonts and what next?

Since the publication of the math extension of the OpenType font format in 2007, barely a dozen OpenType math fonts have been released. This probably means that new math fonts are not (urgently) needed, which does not mean that existing fonts need not be improved, nor that creating special varieties of math fonts, such as heavy or sans serif variants, is useless.