TUG 2022 abstracts

Editor's note: Links to videos and other information posted at `tug.org/tug2022`.

— — ∗ — —

Looking outside the cockpit: An in-depth look at airport signage

Oliver Austin

If you take a quick glance at an airport and its signage, you'll see many different situations where text is used to enhance and streamline processes for both pilot and ground crew alike. Thus, this exploration will take a closer look at such variations along the taxiway and apron at major airports, also discussing how the onset of autonomous aircraft can factor into it.

The residual concepts of production vs. the emergent cultures of distribution in publishing

David Blakesley

Who wins? The base or the superstructure? I'm not a Marxist per se, but I've lived this struggle for some time as a writer and publisher. In this keynote presentation, I describe my efforts to change or adapt the democratized tools of production to produce new forms of writing, which ultimately led to an ongoing battle with the dominant cultures of production in the world of publishing. I'll narrate two case studies. One focuses on the writing and production of an innovative, if not disruptive, textbook in the ultra-conservative textbook industry. The second tells the ongoing story of an interloping publishing company (Parlor Press) that reveals the central challenge of *distribution* for both writers and publishers, from typesetting (print) to transformation (digital). LaTeX developers and users, take note! The return of the nonbreaking space and soft return is nigh!

Fonts and formats of constitutions

Sarai Castañeda

Through the different constitutions from different countries we'll look at, France, Canada, the United States, Mexico, and Argentina it is clear that the fonts range from cursive to typewriter-like. The fonts and format of each country's constitution are based on the time period it was written and other countries' influence. The countries have developed different iterations in order for the constitution to best represent their country's values.

Comparing TeX engines and formats

Max Chernoff

Initially, TeX was a single engine and a single format. However, over the past 40 years, the number of engines and formats has significantly grown, meaning that there are multiple ways of implementing similar solutions depending on the TeX variant used. In this talk, I'll introduce and compare each engine and format, focusing on both history and practical tips.

Revamping a youth chess workbook using LaTeX packages

Jennifer Claudio

Playing chess can range from a casual pastime to a highly competitive event. Several local organizations offer chess as enrichment programs in K–12 schools, often having their own workbooks to supplement their instruction. One drawback is that these workbooks are often created using screen captures of online sources, resulting in low-quality outputs when used for print. This exploration tours a few packages used for typesetting diagrams for chess problems and puzzles and presents comparisons of one enrichment program's original workbook to equivalent pages produced using LaTeX.

Access and accessibility

Jonathan Fine

The Chafee Amendment (`www.loc.gov/nls/about/organization/laws-regulations/copyright-law-amendment-1996-pl-104-197`) to US copyright law "allows authorized entities to reproduce or distribute copies or phonorecords of previously published literary or musical works in accessible formats exclusively for use by print-disabled persons."

This wonderful legal exemption to copyright nicely illustrates the relation between access (here to print works) and accessibility (here production of phonorecords, i.e., audiobooks). Here's another illustration.

Jonathan Godfrey, a blind Senior Lecturer in Statistics in New Zealand wrote to the Blind Math list "I used to use TeX4ht as my main tool for getting HTML from LaTeX source. This was and probably still is, an excellent tool. How much traction does it get though? Not much. Why? I don't know, but my current theory is that tools that aren't right under people's noses or automatically applied in the background just don't get as much traction." (`nfbnet.org/pipermail/blindmath_nfbnet.org/2021-January/009641.html`)

Jonathan Godfrey also wrote to the BlindMath list "Something has to change in the very way people use LaTeX if we are ever to get truly accessible pdf documents. I've laboured the point that we need access to information much more than we need access to a specific file format, and I'll keep doing so. [...] I do think a fundamental shift in thinking about how we get access to information is required across most

doi.org/10.47397/tb/43-2/tb134abstracts

STEM disciplines. (`nfbnet.org/pipermail/blindmath_nfbnet.org/2021-March/009778.html`)

This talk looks at the experience of visually impaired STEM students and professionals, from both the point of view of easy access to suitable inputs and tools and also the generation of accessible outputs, as pioneered and enabled by the Chafee Amendment.

The UK TEX Users Group — a personal history

Jonathan Fine

UK TUG was established in the early 1990s. I've been a member of UK TUG almost from its start through to its dissolution earlier this year. Much has changed both in the TEX community and in the wider world over that time.

UK TUG was a significant part of the TEX community. Besides myself (Jonathan Fine), former members of UK TUG include Peter Abbott, Kaveh Bazargan, David Carlisle, Paulo Cereda, Malcolm Clark, David Crossland, Robin Fairbairns, Alan Jeffrey, Sebastian Rahtz, Arthur Rosendahl, Chris Rowley, Philip Taylor and Joseph Wright.

This list includes two past Presidents of TUG, the current Vice President and a past Secretary. Ten people on the list served on the TUG Board, for a total of over 30 years.

Five are or were members of the LATEX3 project. One was the founder and for 8 years editor of TEX Live, and another the Technical coordinator of the NTS project. One is a Lead Program Manager for Google Fonts.

This talk provides a personal history from `\begin{uktug}` to `\end{uktug}`, with a short '`\aftergroup`' appendix.

New in stock — a walk though recent LATEX improvements (that you may have missed)

Ulrike Fischer

In this talk I present a selection of improvements we made in the recent LATEX releases. The changes are not discussed in depth; the goal is to give some interesting examples and make you curious enough to explore the documentation and learn more. (See the LATEX news installment in this issue, and previously, for details: `latex-project.org/news`.)

Boxes and glue: TEX algorithms reimplemented

Patrick Gundlach

TEX (and therefore LATEX) have enjoyed great popularity over the years as an extremely flexible, versatile, and robust text typesetting system. The flexibility comes not least from the ability to modify the behavior of TEX through programming and from Knuth's foresight in recognizing the individual elements on the page as small, rectangular building blocks that can be combined into larger units and also manipulated (box).

The development of LuaTEX made modern applications possible for the first time in the long history of TEX via some extensions:

- The number of characters in fonts is no longer limited to 256. This eliminates crutches like output encoding.
- Through the integration of HarfBuzz a solid "shaper" is available. This allows OpenType features and complicated writing systems (e.g., Arabic) to be output without any problems.
- The system can be programmed with Lua instead of the built-in macro language.
- Due to the clever PDF support, almost all PDF properties and standards can be supported.

I use these extensions for the program 'speedata Publisher", which is mainly made for the fully automatic creation of product catalogs and data sheets from XML.

Despite all the achievements of TEX and LuaTEX, there are still serious disadvantages:

- TEX and LuaTEX are anything but modular. Changing single areas is especially difficult, because TEX is not designed for that.
- Some things cannot be achieved with LuaTEX's on-board tools. For example, HTTPS requests require an external library. Documents in our catalog area often get their images from image databases that are accessed via HTTPS.
- For other tasks, too, it is better to use an external library than to reinvent the wheel. For example, an XML parser or a library for bidirectional text typesetting.
- Parallelization of tasks: modern processors usually have several processor cores, which lie idle with TEX. Several tasks in TEX could be executed in parallel. Paragraphs could be wrapped with different parameters and then the best one selected. Loading font files and preparing them for subsetting in PDF does not have to be done sequentially. TEX does not provide such facilities.
- Distributing LuaTEX binaries across platforms is difficult due to external dependencies. For single applications you don't want to ship or require a whole TEX Live installation.

The restrictions mentioned have troubled me considerably. Regarding the output quality of TEX, there are hardly comparable alternatives — especially not in the open source realm. Therefore, there seemed no alternative left but to re-implement TeX in a “modern” programming language. Some years ago there was already such an attempt ($\mathcal{N}\mathcal{T}\mathcal{S}$), but it failed. After long pondering, respectively to meet my requirements for a text typesetting system for catalogs and datasheets, I came to the conclusion that I “only” take over the algorithms and the logic of TeX, but not the input language.

Boxes and glue

“Boxes and glue” is a library written in the Go programming language. The name is based on the model of TeX with the stretchable spaces between the rectangular units. The development of boxes and glue is quite advanced and includes among other things:

- TeX’s smallest units (node) with ways to nest them inside each other (vbox, hbox).
- TeX’s paragraph breaking algorithm.
- The pattern-based hyphenation.
- The inclusion of TrueType and OpenType fonts and PNG, JPEG, and PDF images.
- Text shaping with HarfBuzz.

Besides these basic parts, there is yet another library that builds on `boxesandglue`. It offers:

- Reading XML files
- Interpretation of HTML and CSS
- grouping of font files into families with easy font selection
- Handling of colors of all kinds (RGB, CMYK, spot colors)
- Tagged PDF

The application programming interface (API) is not yet fixed. The development of boxes and glue is being carried out in parallel with the further development of the speedata Publisher (`github.com/speedata/xts`) and the requirements here largely determine the programming interface of `boxesandglue`. Since it is a library, there is no fixed input language as with TeX. In this respect also, `boxesandglue` is also yet suitable for and (end) user.

References

- $\mathcal{N}\mathcal{T}\mathcal{S}$: `en.wikipedia.org/wiki/New_Typesetting_System`
- Boxes and glue: `github.com/speedata/boxesandglue`
- speedata Publisher: `github.com/speedata/publisher`
- XTS XML: `github.com/speedata/xts`

Using LaTeX deployed in AWS as a PDF report generation tool for a cancer clinical trial search engine

Hubert Hickman, Matthew Mariano, Haibin Wu, Hong Dat Cheung

Matching cancer patients with clinical trials is a complex process. One of the outputs of that process is the production of a PDF report containing relevant information about a set of trials. In this paper we present strategies, challenges, and conclusions regarding our use of LaTeX deployed in AWS to generate PDF reports.

Bridging the gap between LaTeX/PDFs and the modern web

Nicolas Jimenez

In this talk we explore the history of LaTeX and PDFs in scientific communication, the roles these tools play, and how those roles may evolve over time. We discuss the rise of Markdown for web publishing, its limitations, and opportunities. We also touch on some recent developments by Mathpix to facilitate document interoperability and accessibility for researchers and the broader STEM community.

Right to left `beamer` documents in XeTeX

Vafa Khalighi

I will discuss the recent changes to the `bidi` package allowing users to produce right to left `beamer` documents describing the challenges and what needs to be done. I will also discuss other recent changes of the `bidi` package.

Bidirectional multi-columns and paragraph footnotes in TeX

Vafa Khalighi

Appendix D (Dirty Tricks) of *The TeXbook* describes algorithms for multi-column typesetting and paragraph footnotes, among much more. The described algorithms are used in various TeX packages such as `footmisc`, `fnpara`, `manyfoot`, and many others.

When the package `multicol` is used, things get more complicated. Another level of complication arises when you want to mix these with both right-to-left and left-to-right typesetting.

The `bidi` package provides both right-to-left and left-to-right multi-columns and paragraph footnotes. This talk will describe my own experience learning about how other packages implement multi-columns and paragraph footnotes, and also the approach I took in the bidi package for these features.

Typesetting mathematics in Persian

Vafa Khalighi

I will discuss how mathematics is typeset in Persian and what is required. I will also talk about how the XƎPersian package implements these features and show some examples. I will then discuss recent changes to the `xepersian` package allowing users to change between English and Persian digits mid-math mode.

Observations and analysis of Vietnamese text

Tia Luc

Having Vietnamese as my first language and English as my dominant language has inspired exploration of the history and applications of the former. Considering how Vietnamese and English both use the Latin alphabet, this presentation will explore the similarities and differences between the two using a collection of instances in which Vietnamese text is displayed in our world.

Accessible tables using 'Tagged PDF'

Ross Moore

Some basic requirements for accessibility of tabular material are:

- each cell, whether header or content, must have an attribute providing a unique ID for that cell;
- each data cell must specify the corresponding row and column headers that most directly provide the meaning of the information contained within the cell. This is done via a Headers attribute using the unique IDs for the header cells.

Header cells themselves may have other row or column headers; e.g., as a common header for a block of rows or columns.

Tagged PDF has the tagging and mechanisms to provide such attributes. When the PDF is translated into HTML (using the ngPDF online converter, say) this information is recorded in the web pages, to be available to Assistive Technologies. In this talk we show several examples of tables specified using various packages, as in *The LaTeX Companion*, both in PDF and HTML web pages.

A novel coding idea that allows this to be achieved was presented. This involves two aspects:

- turning the '&' character into an active token while the tabular material is being processed;
- use of 'look-ahead' to see what kind of material is coming at or before the start of each tabular cell.

The example documents shown can be found on the author's website, at `science.mq.edu.au/~ross/TaggedPDF/TUG2022/`. This HTML page was itself created using the same methods, from a PDF file which is available at `science.mq.edu.au/~ross/TaggedPDF/TUG2022/TableSite.pdf`.

Some rationale concerning how header cells are determined in real-world documents is explained in one of the examples: `science.mq.edu.au/~ross/TaggedPDF/TUG2022/FishTables-only.pdf`

Machine translation of mathematical text

Aditya Ohri, Tanya Schmah

We present a machine translation system, the PolyMath Translator, for LaTeX documents containing mathematical text. The system combines a LaTeX parser, tokenization of math and labels, a deep learning Transformer model trained on mathematical and other text, and the Google Translate API with a custom glossary. Ablation testing shows that math tokenization and the Transformer model each significantly improve translation quality, while Google Translate is used as a backup when the Transformer does not have confidence in its translation. For LaTeX parsing, we have used the pandoc document converter, while our latest development version instead uses the TexSoup package. We will describe the system, show examples, and discuss future directions.

Musical composition typesetting

Christopher Park, Emily Park

We will explain the typesetting of a musical composition using the LaTeX markup.

TeX Live 2022 status update

Norbert Preining

This talk reports on changes within the TeX Live project and distribution over the last year, as well as looking at further development directions and challenges we are facing.

Bricks and pieces

samcarter

Real world bricks and jigsaw puzzles are a fun pastime for many people. The tikzbricks and jigsaw packages bring them to the LaTeX world. This short talk will give an overview of both packages and show examples how they can be used.

Detailed descriptions of usage and options can be found in the respective package documentations, linked from the CTAN package pages: `ctan.org/pkg/jigsaw` and `ctan.org/pkg/tikzbricks`.

A gentle introduction to Markdown for writers

Tereza Vrabcová

TeX is great for producing beautiful documents, but not the easiest to read and write. At this workshop, you will learn about Markdown and how you can use it to produce different types of beautiful documents from beautiful source texts that don't distract you from your writing.