
A rally in Bonn: TUG 2023

Eileen Wagner

Thursday, July 13

TUG 2023 took place in Bonn, Germany, after three years of online-only meetings due to the COVID-19 pandemic. Our local hosts, Ulrike and Gert Fischer, invited us to the Hotel Leoninum, a retirement home and hotel that provided a sense of calm and conviviality.

Beyond the usual updates on kernel, key packages, and favorite editors, this year’s topics focused on continued efforts in accessibility, related research around hyphenation, and various experiments in Unicode and more. It wouldn’t be TUG without the riveting demos, be it specific publishing house setups, private practice workflows, or yes, even writing *The L^AT_EX Companion* itself. There was a stream of live tweets and toots about the conference on #TUG2023.

Friday, July 14

The conference took place at the Old Church of the hotel, a repurposed seminary from the 19th century. This was not only befitting the sanctity of the TUG, but also enabled speakers to preach to the choir, from the quire.

The president, Boris Veytsman, opened the conference at 8:30 sharp. He noted that this is the first in-person conference since the beginning of the COVID-19 pandemic, which is special for him and many attendees.

Ulrike and Gert Fischer, Carla Maggi, Paulo Cereda and samcarter kicked off the talks with a lighthearted peek behind the scenes of generating the annual Great TikZlings Christmas Extravaganza video.¹ samcarter went into detail on how each scene was created in Beamer and then stitched together from PDF to PNG to MP4. The talk ended with a compilation of various duck-themed animations.

Next, Oliver Kopp urged the audience to (re)consider JabRef as BIB_TE_X-based literature management software. Their recent work rewired the GUI and switched to BIB_TE_X as its internal data model, which means no conversions are needed. New features such as integrated web search, drag-and-dropping PDFs with metadata, grouping options, and data quality control are truly exciting. Almost as importantly, Oliver described their efforts to have JabRef included as a potential student project at university computer science programs. This meant a lot more contributions for JabRef, and a lot more computer science students who worked on a real-world package with

their code actually being used. Their contribution framework on GitHub² is a great place to start.

Jan Šustek followed with his work on generating documented sources by blocks. The goal is to write source code as well as its documentation in a single file. This was made possible by modifying certain Op_TE_X macros and using nested blocks. The audience was equally awed by the capability of this macro and the fact that they met a real-life plain T_EX user — on macOS, no less.

Barbara Beeton reviewed frequently asked questions on `tex.stackexchange` so you don’t have to! She turned these questions into an introduction to key concepts in T_EX, titled “what every newbie should know”. The article is highly recommended for those frequently introducing T_EX to new users.

After a coffee break, Martin Ruckert came with news from the HINT project. HINT is a file format, viewer, and engine (Hi_TE_X) that renders T_EX documents in a resizable and searchable viewer. (See TUG 2019 and 2020 for more details.) He introduced the issue of optimizing for small font sizes (<10 pt). Outline fonts, along with some tricks in rounding and interpolation, can help glyphs become better defined. Support for links, labels, and outlines were added in HINT as well. Martin showed that some commands are better than others when supporting variable window sizes. For that, HINT needs to design more macros. Did you know that you need to type ‘La T_EX’ in Unicode to find ‘L^AT_EX’ because between ‘a’ and ‘T’ we use a glue and not a kern? Martin’s next projects might include bi-directional typesetting and subpixel rendering. Thankfully he’s going into retirement from teaching this fall, so we can expect even more updates soon.

Dennis Müller then introduced an HTML/CSS Schema for T_EX primitives. But first he talked about Rus_TE_X, which is an engine in Rust that is close to pdf_TE_X. Right now there is a difficult tool chain for producing HTML involving L^AT_EX_{ML}, OMDDoc, MMT, and more, which “might only work on Martin Kohlhasse’s computer”. Dennis’ goal was to combine everything in one package with the same representation format for both humans and machines. This way you can inject additional services such as JavaScript. He used an AI — “a good old-fashioned symbolic AI” — to help with the conversion, and so far the results look promising.

Patrick Gundlach makes a lot of catalogues, and is maintaining a project called ‘boxes and glue’ that brings T_EX’s typesetting engine to more folks (implemented in Go). His motivation is mainly that there

¹ github.com/TikZlings

² github.com/orgs/JabRef/projects/3

are some trivial limitations for Lua \TeX : no https requests, for example, which he really needs! He has his work cut out for him: on the backend with fonts, language, nodes, PDF library, and on the frontend with font families, colors, CSS/HTML, accessibility, page layout, and so on. (And then an application on top!) His boxes and glue engine is “pretty fast” outputting 300 pages per second, which is much better than the 50 pages per second he had with his old catalogue software. He also described some lessons learned, starting with “PDF looks like an innocent file format...” He confirmed the audience’s suspicion that, once set up properly, the typesetting is much faster than the previous InDesign process.

With that, we headed to lunch, with tables sprawling from the hotel restaurant to the sunny terrace. Delegates gathered shortly before the afternoon sessions to take a group picture. One of the sponsors distributed a number of rubber ducks, brightening the post-lunch spirits.

samcarter (and Joseph Wright) gave updates on **beamer**. While their efforts still focused on maintaining backwards compatibility, they also made sure to ship some usability improvements. For example, transparent backgrounds now work as expected; there is a flexible interpretation of the **aspectratio** option; title pages are more modular; the geometry calculation was renewed.

This transitioned to another talk by samcarter on **tcolorbox**, a new theme for the **beamer** class. It replaces normal beamer blocks with **tcolorboxes** of the same look and feel. **tcolorbox** makes it easy to modify the appearance of blocks, accommodating all kinds of user requests (“Can I have the rectangular block with rounded shadows?”). Keep them coming!

Boris Veytsman took the floor to describe his **nostarch** class. No Starch Press is a publishing house that switched their tooling from InDesign to \LaTeX , very much thanks to Overleaf’s wider adoption. The \TeX nic problems he faces there can be unusual. For example, url splitting on hyphens is ambiguous between hyphen (linebreak) or hyphen (url). No Starch has a convention to split *before* hyphens. Another example: captions should be the same width as the corresponding figure. Boris insisted that he can’t fix a bug with **amsmath** (Bad mathchar 32768), but despite his claim that “the magic of category codes is sometimes black magic”, a delegate, David Carlisle, found the issue (including line reference) by the end of the presentation, and the fix is in the current release.

Overleaf’s Ben Davies followed with a presentation of their new Visual Editor. This development comes from well-known issues: many people — col-

laborators or proofreaders or just “young people who are intimidated by any code” — might not be familiar with \LaTeX , so Overleaf wanted to offer a way for them to contribute without using \TeX syntax. This was made possible by the switch from CodeMirror5 to CodeMirror6. The new Visual Editor is a WYSIWYG editor that hides code and adds previews of maths and figures. Track changes and comments work with both the Code Editor and Visual Editor. Ben discussed many user experience decisions, such as introducing a visual “undo” button that gives people the confidence to make mistakes and experiment. The crucial issue is to balance how much code to hide, and refraining from too much WYSIWYG conditioning which can feel limiting. So far, 2% of Overleaf users are making use of the Visual Editor.

The visual theme was continued by the next speaker, Didier Verna, who demonstrated his visualisation software for interactive and real-time typesetting. \TeX engines are production systems that are not meant for playing around, and so his tool focuses on experimentation and demonstration instead. His program ETAP³ showcases kerning, ligatures, baselines, etc., in typesetting algorithms. (It of course includes the Knuth-Plass algorithm.) After a brief detour on programming in multi-paradigm languages that allow inheritance and polymorphism, he went on to demonstrate the power of ETAP to produce statistical reports on the behavior of the various algorithms. He showed a number of benchmarking tests for efficiencies of various algorithms. He hopes to add more parameters in the future — **microtype** and a tolerance threshold (sloppiness) are clear candidates.

Eberhard Lisse took on a review of Typst, a recent addition to the typesetting ecosystem. Eberhard started using \LaTeX early on and, like so many of us, never wanted anything else. He gave an in-depth view into the use of \TeX in his medical practice, and asked “Is \TeX ready for prime time?” Even though his staff has begrudgingly learned \LaTeX and he’s gotten very far with customising his tool chain, his answer is a sad “no”. “For casual use it’s much too complicated,” he concluded. More reason, then, to be excited about a new alternative: Typst. Typst is both a typesetting engine, a collaborative editor, and a platform built by two computer science students from Berlin, who may have shared similar feelings about the production-readiness of \LaTeX . In Eberhard’s view, Typst fills the gap between advanced tools (\TeX) and simpler tools (Word/Google Docs): it is highly capable, blazingly fast, and uses a simple markup language. It is also less powerful than

³ github.com/didierverna/etap

TeX. He proceeded to point to missing features — namely floats, indices, interactions with the environment, and labels — but remains hopeful that sensible decisions will be made. *Eberhard is also going into retirement this fall, so we can expect more experiments in Typst!*

Next, a research group from Masaryk University consisting of Ondřej Sojka, Jakub Máca, and Petr Sojka presented a roadmap for universal syllabic segmentation. The tagline: give patterns a chance! They are studying the similarity of languages when it comes to hyphenation patterns, starting with Czech and Slovak, and following up to nine languages (cz, sk, ka, el, pl, ru, tr, tk, ua). Their results: syllabic segmentation across languages is possible using Judy arrays; and their hypothesis: universal patterns are feasible, too, with high impact on virtually all typesetting engines.

The conversation then seamlessly transitioned to the TUG Annual General Meeting (see separate report in this issue). Delegates left the hotel in time to eat out on this warm summer evening.

Saturday, July 15

Henning Hraban Ramm showcased the architectural guides for Bonn that his publishing house⁴ made, using first InDesign and then ConTeXt in production. Some of his customizations in ConTeXt include OpenStreetMap integrations, beautiful bleed and trim boxes, and shadow captions for white text on a monochrome background. When asked how long it took to produce the book, he said that the basic typesetting was very quick, but it took hours to set it up; in InDesign it was the other way around.

The morning continued with beautiful examples from Thomas Schmitz, who teaches ancient Greek at the University of Bonn. He switched to ConTeXt years ago with the ambitious goal of having all lecture content and notes in the same source document. For him this meant XML, with its advantage of easy code reuse and outputs in different formats. Examples of these formats are lecture slides, course schedules, translations (philology), printout for students, bibliography for conferences. This amounted to over 30k lines over one term. Thomas demonstrated his setup with an integrated Lua interpreter. His presentation module is available on the ConTeXt Garden.⁵ Upon the question of any preprocessing setups (linting and checks), he responded that that won't be possible because of his editor (Emacs) — cheers in the room.

⁴ www.dreiviertelhaus.de

⁵ modules.contextgarden.net/cgi-bin/module.cgi/ruid=5038752680/action=view/id=31

Vít Novotný came next with a Markdown 3 update. He observed that L^ATeX is easy to write, but hard to read! This presents a challenge, since most of the time writers need to read what they are writing. Markdown is not just a great option for those wanting a non-distracting syntax; it is also a great option for TeXnicians who need to work with clients and publishers who do not use TeX syntax. Vít went on to discuss the Markdown syntax of choice, CommonMark by Jeff Atwood and John MacFarlane.⁶ Tables, footnotes, and citations are not included though, and for structured metadata, a YAML file is still required. The exciting feature is the “hybrid” mode that allows authors to switch back and forth between Markdown and TeX syntax. Until now, Markdown-enabled TeX files were not interoperable with other Markdown clients. Markdown 3's hybrid syntax does not break interoperability. It also introduces some neat features like tables, task lists, superscripts, and more. It added TeX4ht for websites, OpTeX support, and also updated ‘pandoc-to-markdown’; `lua-tinyyaml` is now on CTAN. Perhaps more importantly, Markdown 3 is now en route to have better governance and community management, adding rooms in Discord and Matrix to welcome developers and *authors* alike. The stable release is coming soon. The session ended with an admission of guilt: Vít created his presentation with Google Slides.

Next, Rishi T from STM Document Engineering announced a new editor, Primo. As a company that works closely with publishers, they have built many tools that ease the authoring-to-publishing workflow. With Primo, they wanted to create an all-in-one solution for many known issues in the submission process. These issues include: authors can't understand journal's requirements, authors cannot collaborate easily, missing material during submission, the back and forth querying, and general technical constraints in the submission systems. Primo is a cloud-based authoring, submission, proofing framework that hopes to be self-explanatory and user-friendly. It will offer a WYSIWYG and non-WYSIWYG mode as well as collaboration, which the audience saw in screenshots. It is XML-based and a DTD-compliant tool. There was great disappointment in the room when the timeline was announced: the platform will be released between April 2024 and April 2025 in three phases.

After the coffee break, Ross Moore gave an overview of accessibility principles and practices via a pre-prepared video. He gave a thorough tour of TeXed documents in various PDF readers, evaluating

⁶ commonmark.org

them against the PDF/UA standard⁷ set by the W3C and WCAG. The display of bookmarks vs. tables of contents vs. hyperlinks is not always compatible with navigation, for example, and a lot of information is hidden in fields that are not read by screen readers (hover-over). The best practice is to include alt-descriptions for information blocks, and adding navigation shortcuts in sensible places. Ross also referenced best practices in accessibility design in web pages, which have more established standards and tools for evaluation. A well-tagged document can have its semantics carried over to web pages as well.

Following the general introduction to PDF accessibility, Ulrike Fischer gave an update on the `tagpdf` project. The goal is to add structure to PDFs to improve accessibility and reuse of data. Adding tags (alt-descriptions and other useful metadata) is already possible in the current release, but the challenge for this project is to make tagging automatic and easy to use. Some first problems (again echoing Ross’s talk): free-of-charge PDF viewers don’t show tags in the visual interface (“If people can’t see it in their resulting PDF, they often don’t want it!”); this also makes testing difficult. Moreover, many PDF viewers don’t support PDF 2.0, which is important for proper tagging. Ulrike described tagging standard “Leslie Lamport documents” as tests — this is possible now as of the summer 2023 release! Some things are not yet supported, notably footnotes, sectioning in `memoir`, etc. More work ahead!

Tagging is not the only core construction happening now. Joseph Wright gave an update on `expl3`, very much motivated by the question “what does \TeX NOT do out of the box?” \TeX is “blind” to core document elements such as color and graphics — the boxes to be glued, so to say. Joseph took a closer look at these backend support files, such as `drivers.dtx` which includes a lot of clutter accommodating different developments of backends over the years. `expl3` tries to create a single source for `.def` files that would be compatible with core backends (`pdf \TeX` , etc.). This principle applies not only to colors and graphics, but also hyperlinks, PDF constructs, tagging of course, drawings, and box transformations. As with most rewrites, the difficulty lies in finding the right level of abstraction. Given the immense undertaking, the audience wondered if it would be easier to create a specification or a set of primitives. This way, engine developers can carry the burden of implementing it respectively.

More news from the \LaTeX Project: Frank Mittelbach was ready to present *The \LaTeX Companion*,

third edition. The “monster” weighs 3.5 kg and took five years to complete. A primary goal was to classify CTAN packages according to their functionality, usability, and correctness — and how they relate to other packages. After an initial review, Frank decided to focus on 500 (about 10% of all packages). For those who might compile a 1600-page book with over 100 fonts in the future, Frank shared his lessons learned:

- Test all examples included — better even to have them in production.
- Take care of pagination *after* you copyedit and finalise font sizes.
- Keep layout code as separate as possible, so you can distinguish between things you changed for the layout and things you changed “to make the page look right”.
- Never underestimate the power of automated checking.

Frank welcomes scrutiny and contributions of course. Now off to lunch!

In another remote presentation, Jim Hefferon made a case for using Asymptote,⁸ a descriptive vector graphics language for technical drawings. It is in part based on Metafont and MetaPost, but extends it from 2D to 3D. Asymptote uses a single source file for related graphics, meaning (unlike `TikZ`) all graphics are *outside of the document*. Jim showcased a number of Asymptote examples.

Linus Romer, in designing fonts, threw himself in at the deep end of curvature combs. Most font editors offer curvature-related tools, and curvature combs are one of them. He described his implementation of curvature combs in MetaPost, along with a number of harmonization algorithms that helped smooth out the paths. As expected, there is lots of math behind curvy characters!

More font adventures: Rajeesh KV talked about shape-shifting Indic scripts. In Malayalam, any consonant followed by the vowel sign of u, ū, or ̣ are represented by a cursive consonant-vowel ligature. The glyph of each consonant has its own way of ligating with these vowel signs. We may call this “ligatures on steroids”. Rajeesh developed a reusable component-based design for these Malayalam fonts using Metafont/MetaPost to assemble the characters. This shifts the paradigm from visual tools to code-based tools. His assembly line: MetaPost, SVG, FontForge, scripts, and finally OTF/TTF/WOFF2. It is even possible to specify width and angle of the pen for the shape library!

Victor Sannier representing the French TUG (Le Groupe francophone des Utilisateurs de \TeX),

⁷ www.iso.org/standard/64599.html

⁸ asymptote.sourceforge.io

GUT) showcased his Metafont for rustic capitals. Rustic capitals as a type started in the 1st century, and had been regularized in the 4th and 5th century. He measured and analyzed many of them until eventually tracing them with a ‘`draw_serif`’ macro he devised. He said that he would continue to design more rustic characters and welcomes feedback.

Meanwhile, Bonn is getting warm and humid, and the organizers have opened the doors and turned on the fans. Delegates opened bottles of sparkling water and cleverly mixed them with apple juice to create the inimitable *Apfelschorle*.

Another font talk followed. Ulrik Vieth had the unenviable task of reviewing all OpenType math fonts—so you don’t have to. He both evaluated their completeness and design choice. He started with a history of OpenType math fonts which were first ready for use around 2010. The list is now over 20 official fonts, and more than 30 if you count not yet released or not properly licensed fonts. Some fonts have around 500 symbols, some have 1200. This largely depends on the availability of additional series such as script, fraktur, or blackboard bold.

Building on the previous talk, Mikael Sundqvist dug deeper into OpenType math. The root of many issues here is that there is no good standard! While Microsoft offers some instructions, it’s unclear what should be included in a math font. Several OpenType math fonts were created by converting and extending older T_EX fonts. Together with Hans Hagen, Mikael works on math in ConT_EXt. He observed many frustrating inconsistencies: italic corrections are transformed to corner kerns, staircase kerns are inconsistent, accents from different fonts look completely different. Depending on the engine, the glyphs are also differently rendered. Extensibles and rules often don’t work. Their efforts have corrected for many of these issues in ConT_EXt. Moreover, his observations often feed back to the font designer, many of them responsive. He promised to publish an article outlining best practices for math font design.

For the final talk of the day, Tom Hejda proposed closer collaboration between T_EX Live development and Overleaf. Tom reiterated the Overleaf view that T_EX Live should be “all you need to use L^AT_EX”. Due to their uptime requirements, Overleaf can only afford an annual deployment in Q3 every year; that version of T_EX Live is then locked for a year for their users—roughly two million at this point. Their schedule, along with the pressure to have a version of T_EX Live that’s compatible with the Overleaf ecosystem (most importantly: the templates that are offered), prompted Tom to start a discussion on whether and how T_EX Live developers and Overleaf

can work together. This would also mean T_EX Live could benefit from early deployments and extensive testing. The group recommended that Overleaf contact the core team as well as package maintainers as soon as possible when deployments fail. (It turns out that, just like Overleaf, the community is also not thrilled about hard deadlines!) It is also possible for Overleaf to test the early (beta) version of the newest T_EX Live release. It was noted that more frequent deployments (continuous integration) would directly benefit their users.

With that, the group moved on to a banquet at *Konrad*’s on top of the Marriott Hotel. We spent much time on the terrace overlooking the Rhine, enjoying the view of the Seven Hills and the old parliament of West Germany. A three-course meal followed. Some delegates chose to walk back to the city center along the river, a calm and inspiring end to Day 2.

Sunday, July 16

For those quick to recover from the banquet, Day 3 began with Ben Frank’s presentation on Docker containers,⁹ a community service offered by the Island of T_EX. The brilliant idea behind Docker: it is self-contained, meaning each T_EX Live image ships with literally everything required to run it. Ben went on to describe two use cases that benefit from Docker. First, a small German maths journal that needs to be able to run older versions of T_EX for historical issues. Second, student representatives who compile their weekly meeting minutes with a completely CI-based and ConT_EXt-heavy setup. Future directions for dockerized T_EX could be including more operating systems (such as Raspberry Pis) and layer-friendliness (only pulling changes). He added a public service announcement: do not use ‘latest’, it is not image-friendly.

Joseph Wright took the audience on further adventures in Unicode-land. The code for the Unicode case changing algorithm is now in the kernel. So far, the project largely involved updating case mappings for Unicode engines, automatic locale switching via `\BCPdata`, and full Unicode support for pdfT_EX. Joseph showed more recent work and challenges. One key take-away: Greek, with all its cases and accents and exceptions, is a great language to stress test your Unicode engine! Further improvements are coming, of course, such as true titlecasing and a closer look at graphemes (human-perceived characters).

Oliver Kopp introduced his library of L^AT_EX templates¹⁰ motivated by providing a simple IEEE

⁹ islandoftex.gitlab.io/community/projects/docker

¹⁰ latextemplates.github.io

template that works out of the box. Many existing templates do not include key packages correctly, such as `microtype` or `hyperref`, and don't include minimal examples as guidance. His templates contain a bit of templating language, and the tool includes a CLI for basic design choices. The setup via “micro-templates” (modules, so to say) can reduce the overhead of maintaining \LaTeX templates. Importantly, there are automated CI checks to ensure that the templates are always compatible with the newest stable release.

Paulo Cereda, Overleaf community coordinator and Island of \TeX core member, gave a run-down of the latest developments on the Island (there are many!). First off, `albatross`, a CLI that helps you find a font based on particular glyphs, was redesigned with border styles, font lookup, and graphemes. Users can use glyph, hexadecimal, or multiset union to search. The Island's community website was also updated, with an index pulling from all the READMEs of the projects. There are exciting ideas for the future, such as providing binary code for packages. The Islanders will also continue their quest to improve user experience. Finally, the Island is ready for a new visual identity. Help on all of the above is explicitly wanted: “visas to the Island are easy to attain!”

To close off TUG 2023, Frank Mittelbach shared the highlights of his last 38 years with \LaTeX through a personal picture story with a mere 90 slides. It started with an invitation in 1989 to attend TUG at Stanford, and appropriately ended with finishing the new *\LaTeX Companion* in 2023, documenting many friends, ideas, and past TUG convenings across multiple continents.

Brushing aside the sentimentalities, Boris (now outgoing president) shared that he was glad that the group was able to bring back TUG in full glory. He thanked the organizers for the great line-up and cultural program — which would be continuing onwards to an outing to the Seven Hills (Drachenfels and Königswinter). The group lingered on the terrace after lunch, bidding their farewells and exchanging last comments. Until next year!

Acknowledgment

I'd like to thank the TUG bursary for funding to support me in attending this conference.

- ◇ Eileen Wagner
Berlin, Germany
`hello (at) bumble dot blue`