TUG 2015 conference report

Stefan Kottwitz

TUG 2015 — the day before

The TUG 2015 conference took place in Darmstadt, Germany, from 20th to 22nd of July. This was the 36th annual meeting of the international TEX Users Group. DANTE e.V. (`http://www.dante.de`) sponsored the conference fee for its members with 50 Euro for each attendee. I'm sure this great sponsorship helped many TEX friends to come.

By way of introduction, I work for Lufthansa Industry Solutions (`lufthansa-industry-solutions.com`), developing and implementing networks for cruise ships. I planned to visit the conference privately. When I told my project team that I will leave for three days, they asked me why. Conference? TEX? What is this? I explained what TEX is and how I used it in my work. I use TEX as macro language for creating thousands of lines of switch configurations. And I create graphics visualizing the physical and logical structure of networks using TEX coding, specifically Ti*k*Z — what other people point and click with Visio. That convinced my boss to consider the TEX meeting as training, so the company covered my travel costs. He expects that our work in designing and documenting will benefit, and this is true.

I arrived on Sunday right before the conference. There was an informal gathering at 7 pm in a restaurant, so I walked there. The restaurant was pretty full of TEX friends, seems like almost everybody was already there. That was a great occasion to meet people again, some I have not seen since 2011, when I attended the TUG meeting in Kerala, India. One such was Kaveh Bazargan, who has been elected to be the next TUG president. His plans are very interesting, as he is interested in boosting online presence, attracting new users, and attracting publishers to get things funded. Of course I met many DANTE members, whom I saw last in Stralsund earlier this year at the DANTE meeting. I also had interesting discussions, such as about `tex4ht`, with two men from VTEX (`http://www.vtex.lt`), a LATEX-based publishing company based in Vilnius, Lithuania. I met Reinhard Kotucha again, who is a regular at TEX and DANTE meetings for a long time. When you arrive in a restaurant and don't know most of the people yet, it's good to see a familiar face. As usual, Reinhard takes photos of the conference and the surrounding world.

Editor's note: Originally posted at `latex-community.org` by the author; edited for *TUGboat*, with permission.

Finally, only a group of DANTE people remained. About 11 pm it was time to return to the hotel.

TUG 2015 — day 1

About 9 am, the current TUG president, Steve Peter, opened the conference with some introductory words.

The topics for today:

- PDF: enriching it and making it accessible
- Unicode: getting it into TEX
- Past, present and future of TEX, LATEX and fonts
- News and announcements

Let's take a look at the presentations.

Ross Moore gave the first talk. He spoke about semantic enrichment of mathematics. At first, he demonstrated the already available possibilities to add PDF tooltips to text and math. Tooltips mean text boxes popping up when you move over it with the mouse or cursor. While they won't be printed, they add value to electronic documents. The reader is able to access further information, which is otherwise hidden to stay focused. Constructing math expressions with embedded semantics requires a difficult syntax though.

Ross introduced a new package called `mathsem`. It offers a way to provide the semantic meaning separate from a math formula. Here's an example:

```
\(
%$semantics
% x $ variable
% f $ function
% \Re $ real part of
%$endsemantics
y = f(x)
\)
```

The characters in the formula now get their own tooltip. As you can see, you can define semantics for macros. This is in fact recommended: define a macro with a name explaining the meaning, rather than how it's typeset, and add semantic information to the macro. The syntax is:
`%` *token* `$` *semantics end-of-line*

It's implemented by hacking the catcodes of the comment symbol `%` and end-of line, and then hooking in. This way has the benefit that even without the package, everything just works, since the additions are hidden in comments and look like code annotations, thus still useful in themselves.

To ease the work, `mathsem` provides a command `\DeclareMathSemantics` for repeated use, setting up default tooltips for symbols and macros. The tooltips can be also be used for a screen reader, meaning for vocalization by assistive tools. Ross

showed several examples, and discussed the application to have spoken words for math, in the context of PDF/UA documents.

Following this, Olaf Drümmer explained what PDF/UA means, in the following talk. UA stands for universal accessibility, which basically means that a document provides reasonable access to the PDF content. So, a visually impaired person could use a screen reader to get vocal information. It requires that all content (including images) be available as text. Other demands are reliable, defined order; embedded semantics; a logical structure; metadata; and not using encryption. PDF/UA is related to tagged PDF: a well tagged PDF can conform to PDF/UA. Olaf demonstrated screen reader software, a PDF/UA checker, and a visualizer tool.

Another talk by Ross Moore together with Peter Selinger gave us an update to new developments of the `pdfx` package, which helps in producing documents conforming to PDF/A standards. There are various PDF/A-x standards, and the new version of `pdfx` supports most of them. Furthermore, the new version adds support for LuaTEX.

Now we got a coffee break — a few minutes to fetch a coffee and to drink it, talking a bit; I wish it could be longer as I value such conversations during conferences.

Joseph Wright gave the first talk after the break. While the first session was about PDF, we now come to the topic of Unicode. He spoke about getting Unicode input into TEX and the challenges involved. For example, he explained the difficulties in uppercasing, lowercasing, titlecasing and casefolding. Details can give a developer headaches, I feel. There are a lot of differences to consider in various languages.

Then it was Will Robertson's turn. At the stage, he changed roles, grabbing his camera and said: "Everybody wave!" and took a picture of the audience. Then, he started with a retrospective about his development of the `unicode-math` package.

`unicode-math` allows us to switch math fonts as easily as switching text fonts. There are thousands of math glyphs in various fonts, each one with a LATEX name, but you can also simply use it as an input symbol. This can be done by code-auto-completion of the LATEX command to that symbol. This may increase readability, but not in all cases, such as when we have glyphs that are too similar. Thus, `unicode-math` gives direct access to a huge collection of symbols. A font with proper Unicode support is required, of course. Luckily, there are some.

LATEX authors commonly use fonts to convey a meaning. In Unicode mathematics, you keep the same font but choose a symbol with the desired meaning. There are a lot of spacing challenges because it is done differently in math compared to text.

Next, the GUST team discussed how characters for math fonts are chosen. That led into a talk about whether we really need new fonts, or if there's not enough demand. We heard about the TEX Gyre math fonts, described with their underlying scheme, and with variants of bold, italic, sans-serif, double-stroke and more. Requirements were discussed such as scaling factors for subscripts and superscripts, math kerning, glyph links, growing glyph chains. There are over 4200 glyphs in DejaVu Math alone.

But few companies produce OpenType math fonts. So, perhaps there's no commercial pressure for math fonts, or not enough demand. So, a future task is possibly not making just another font, but font variants, such as sans-serif variants for presentations, or bold/heavy variants for headings.

Frank Mittelbach then talked about history and current development of LATEX 2ε. Now, it's 21 years old. The policy of compatibility will now change to a policy of roll back–roll forward. Fixes and enhancements will be applied directly to the kernel. You can call the `latexrelease` package with a date as option, and it will change to be compatible to the version of that date. Also packages can adjust their code to releases via an `\IncludeInRelease` macro. There will be also patch releases which will not be roll-back points, in contrast to major releases.

All the fixes of `fixltx2e` are now in the kernel. ε-TEX support is now included out-of-the-box, along with fundamental support for XETEX and LuaTEX, a regression suite for testing with all formats. ε-TEX and XETEX passed the tests, while still there are many failures in running against LuaTEX, to be examined. Some improvements as well:

- `\emph` can now produce small caps or others
- `\textsubscript` is defined
- `\DeclareMathSizes` only takes points
- fewer fragile commands

The final speaker for the third session was Hans Hagen. His talk "What if ..." was more reflective. He looked at Unicode, which is a nice way to get rid of input encodings and font encodings, with easy transition thanks to existing UTF-8 support. And there are sophisticated casing, categories, and spacing. What if we had had it earlier? A lot of time was "wasted" struggling with encodings. However, Unicode may introduce challenges due to possibilities, cultural issues of symbols, and persistent compatibility errors. And, there are exceptions due to user demands.

He reflected on TEX's design, which is nice but

may be boring sometimes. The look and feel may not fit some purpose such as schoolbooks. Generally, nice fonts help reading, but of course there are different opinions on design.

Hans explained that we still have insufficient but persistent standards. The market demands XML, in and out, and support of tagged PDF and EPUB. Then, he took a look at the development of speed and memory and how we can benefit today, such as saving time struggling with hardware constraints. What if we had this 20 years ago? What would be our position today? We compete with WYSIWYG and real-time rendering, so we should explore today's hardware benefits with more effort — though constraints can lead to better solutions.

He finished with a look at perspectives. Will TEX become a niche or go mainstream? Will it be a backend? Or more used by individuals? Requirement for quality doesn't grow; other apps can do well too. Can we focus on control, and on cultural aspects? The future is not clear.

A short break followed, with some ice cream in the hotel cafe. Joseph Wright then gave an update on the status of the UK TEX FAQ. The original Cambridge server is not available any more. So, the FAQ has moved to another server, maintained by me. The original and established domain name (`tex.ac.uk`) has been preserved. Now, while I like to help in continuing, improving accessibility and web design, the UK TUG team will continue maintaining it. Joseph Wright asked for people to help improve the FAQ content. This work load can be shared, so anybody could focus on a specific part. The FAQ sources are on `github` for further development.

Joachim Schrod then described to us the services provided by CTAN. An essential part is providing LATEX and other packages to chosen servers, which in turn sync them to about 200 mirrors. That's for us TEX users, who can then update our packages. This is a principal task of all the CTAN-related servers, noticeable by the many terabytes moving. Other services, such as manually browsing package directories, are less used, as this is done more by developers than end users. Archiving and mirroring involves challenges such as observing mirror server status and checking if they are up to date.

The heart of CTAN is the TEX catalogue. Maintenance of packages' metadata is a laborious but fundamental part of the archive. Besides incoming mirroring, there are services such as the web server, including upload management, and mailing lists. With a look at the load on the CTAN servers, Joachim Schrod confirmed Hans Hagen's words that TEX may be a niche — but it's a large one.

Finally, Barbara Beeton and Volker RW Schaa gave words in memoriam of Pierre MacKay, Richard Southall and Hermann Zapf, who have passed away.

In the evening, there was a dinner in the Dreiklang restaurant. The complete LATEX3 team, Henri Menke and I decided to go to the Ratskeller. We talked about current LATEX3 development until late into the night.

TUG 2015 — day 2

Pavneet Arora started the first session with a talk about FLUSS, a flow leak monitoring system. I was curious, how this should be related to TEX. The working title, FLUSS, is an acronym. It stands for "Flow leaks unearthed ss" where ss means 2x sigma. The latter refers to double sigma testing.

His talk had little to do with typesetting. But it has to do with TEX. Pavneet considers TEX to be a part of the core stack of embedded systems. He uses TEX as a sophisticated documentation backend, and for reporting, so in this case not for publishing. So to say, he used the "TEX of Things" to detect water leaks. Why is this important? As with fire, water damage can be limited by catching the problem early on; you don't have fire detectors at home, but smoke detectors, to get warned at an early stage. He focused on the water supply instead of all possible breaks and leaks along the whole supply way. His application suite is monitoring the flow at the source side, e.g., near the water meter. It learns water consumption patterns over time, and results in ConTEXt-generated reports. They allow the triggering of alarms thanks to pattern recognition. The hardware is an embedded system based on a Raspberry Pi. There's a bunch of tools to install — TEX was the easiest part, a great sign of its maturity and its reliable packaging. We saw ConTEXt-generated diagrams and how to detect a water leak there. This topic is highly important to insurance companies, connected to much money. Thus, Pavneet showed an interesting and unexpected use of TEX in that industry.

In the next talk, Tom Hejda spoke about preparing LATEX document classes and templates for the Czech Technical University in Prague (CTU). He spoke about differences in creating classes for journal articles compared to university theses. There, he considered the user's point of view, stated some basic facts and gave examples. He started with typical usage. The procedures are different:

- Journal article: author typesets, it's reviewed, there's a final author version, it's copyedited and typeset.
- Thesis: student typesets, supervisor comments, the final version is submitted by the student.

A journal has its style, decides which packages can be used, etc.; the journal has full control of output. In contrast, with a thesis, the university has style restrictions, but the students mostly decide how to actually typeset the thesis. Journal articles and theses also differ in sectioning depths, used packages, and in the variety of topics, which is comparatively narrow in most journals.

Tom compared these examples and discussed differences in their approach:

- `actapoly`, a class for journal articles in the *Acta Polytechnica*, written in a mixture of TEX and LATEX 2ε.
- `ctuthesis`, written using LATEX3 as much as possible, with a rich key=value interface.

Boris Veytsman followed, presenting a new multibibliography package. There's actually a package with this name. He reworked it, and his new package is called `nmbib`.

Generally, a bibliography is not merely a technical list. It describes the state of the field. So, not only an alphabetical listing, but also a chronological list shows the development and progress in the field. With `nmbib`, you can have ordering by name, by appearance, and chronological, all in the same document. Each cite command produces entries for all lists. With the old `multibibliography` package, there were some limitations, such as support for just fixed BIBTEX styles. Perl was required. With the new `nmbib`, you still get a look and feel similar to `multibibliography`: you get three lists with `hyperref` links. But now `nmbib` has compatibility with (and in fact loads) the `natbib` package and supports its commands. Any `natbib` style may be used for alphabetical or sequential bibliography lists. You don't need Perl any more. Instead of using a Perl script, BIBTEX is simply run three times for three orderings. `nmbib` is much more flexible compared to `multibibliography`, since all `natbib` customizations can be used, and citation styles can be customized. The new and more flexible `nmbib` package has also been developed with ebook usage in mind.

Leila Akhmadeeva joined Boris for a presentation about trilingual templates for an educational institute in Bashkortostan, Russia. This is a special challenge because Bashkir Cyrillic is different from Russian Cyrillic. A formal document is already a challenge for a style designer, and a consistent multilingual style is even more so. TEX is a good tool for such tasks. They chose Paratype for consistent fonts.

Paul Gessler followed with a talk about printing Git commit history graphs. Git is a popular version control system. Based on the `gitinfo2` package, Paul wrote an experimental package called `gittree`, which generates such graphs for use in LATEX, on the basis of Ti*k*Z, and provides a convenient interface. He showed use and creative abuse, such as with a `github` project MetroGit where each commit is a metro station, and a branch is a metro line, a merge is a connection between lines; together, it produces a map of the metro stations of Paris. Paul's code will be on `github` by the end of summer, and he expects to put it on CTAN in early 2016.

Steve Peter then did his final task as a president: introducing the new president, Kaveh Bazargan.

Kaveh said that it's an honor to hopefully contribute in this new way to our great community. He has worked with TEX since 1983, and his first TUG meeting was 1986 in Strasbourg. He said that TEX deserves to get far more visibility. So, he hopes we keep old friends but get more youngsters to join in. We can show there are many things TEX can do which can still hardly be done using other technology, although applications have essentially caught up in many respects. He thanked everyone for voting in the election, and announced that he will talk afterwards with the people who voted differently ... with a smile. The new TUG board then gathered before the audience. (The following report on the annual meeting includes contributions from Boris Veytsman.)

A first suggestion, quickly approved, was the founding of an accessibility working group. Klaus Höppner said we could join forces with institutes working on tools for blind people. Many people are working on things such as tagged PDF; teams should be brought together.

It was said and agreed that we should demonstrate more vigorously how TEX is used in academic work and industry. But where to show it? At TUG meetings, we are talking among ourselves. The TUG web site is visited mostly by TEX users too, probably not by not-yet-users. There is a TEX showcase there.

My silent thought was that I could adapt and expand the TEXample gallery, a tagged and categorized gallery built by Kjell Magne Fauske. It's currently focused on Ti*k*Z, but could be extended to TEX in general. There are sophisticated back-end scripts for automated workflows including compiling, adding to file shares and database, tagging, and generating output in PNG and JPG via Ghostscript for gallery view and thumbnail preview.

Frank Mittelbach said that we should support the entry level at universities. Some opinions went further: we should promote very early use of TEX, such as in schools. The potential TEX entry point today is often when people start writing their thesis.

But by that time, they have already used Word and such for 10 years. TEX comes late here. Few people feel the need to switch from Word after using it for six or ten years.

Supporting this thesis, Rajagopal CV told about his experience in teaching in India, at the B.Sc. and M.Sc. levels at the University of Kerala. (We hope a detailed article will be forthcoming.)

Regarding publishers and TEX: few publishers use TEX — they were in the room, notably River Valley Technologies and VTEX. Many publishers deny TEX because there's still an old bad reputation. More and more other applications catch up. 95 percent of files coming from authors are in Word, so the industry has developed clever things around Word, expensive tools to work with Word, such as taking out references for processing. The industry standard is conversion from Word, period. TEX is in the minority. Though, it's a big industry to tackle, if you know how.

We should promote the information that the TEX distribution is actively maintained, and will continue to be. That's an important criterion. How many people go to conferences in other fields, to tell about TEX? Not so many. Of course, other user groups are also mostly among themselves.

Boris Veytsman started a discussion in another direction. Whatever we all think, where is TEX going to be in the future? And how about TUG? In past times, there was much meaning and reason for being a user group. Without today's Internet, we promoted and helped users. But today? We may have made ourselves unneeded, because we did a good job. There's CTAN, a user doesn't need to be a TUG member any more to get all the software, online help in forums, and most of the membership benefits. What reason is left to join? Thanking and sponsoring, what else? Many members join because of sympathy. We should find a justification for the TEX Users Group to exist, as such, find a convincing reason for people to join. This was an open discussion with many people contributing. Why do we need our group? How can we tell anybody that we are relevant? Should and could we find a new identity? Why is a user group necessary?

Brought up by Matthew Skala: TEX is not the first choice even in the open source world. A user buys a new computer with Ubuntu and just clicks on "Create a new document" icon. Ten times out of ten the system will open an OpenOffice or LibreOffice clone of Word: is this inevitable?

Still, even though millions of users rely on TEX, things can easily break. As wonderful as they are, teams are small. The CTAN maintainers' group consists of four people, the LATEX team consists of five people, and so on. If a key person gets ill, everything stops. Rarely do new people pick up. It's not only about users or money — an important issue is getting users to contribute and to turn into developers. We are seriously low in developers: we need users to turn into being developers. So, user groups are essential to activate people who start contributing.

That was a serious discussion, and it's good to bring up such points; to raise questions to find answers. Nevertheless, to be sure, people here are positive and in a great mood.

Away from the TEX front, in the afternoon, there was an excursion to the Messel Pit (`https://en.wikipedia.org/wiki/Messel_pit`), a UNESCO World Heritage site because of its abundance of fossils. In the evening we met at the Herrengarten and talked until late.

TUG 2015 — day 3 — first part

The third day was opened by Kaveh Bazargan and Jagath AR. They talked about today's requirements of publishers who demand XML. Kaveh showed issues with XML. He gave examples of proper XML encoding but crazy meaning, such as embedding each letter within XML text or writing a plus-minus sign by a plus symbol with an underline tag. He reviewed the classic publishing chain, from author to publisher to peer reviewer to copy editor and finally to the typesetter with possible loops. Then he showed the cloud approach, which is not so linear but more star-like: when publishing in the cloud, the XML file is in the middle while the involved parties all work directly with that file. He showed an online editor on the River Valley Technoogies platform which allows editing, reviewing and correcting with a rich online editor. There, the file is always saved in XML and rendered into HTML or PDF on the fly. Authors are editing XML, but TEX is used in the background. Specifically, TEX is used for pagination of XML documents and producing high-quality and even enriched PDF output with different styles from the same XML base code. Jagath AR showed some examples of enriched PDF, such as PDFs with several layers for screen color mode, black and white, and CMYK coloring, all in the same PDF file.

Joachim Schrod then gave an experience report about TEX in a commercial setting. The purpose is producing all written communication for an online bank. This means usually small documents, but counted in the millions, with severe legal requirements. Such document types are letters with standardized or individual content, PIN/TAN letters, account statements, credit card statements, share

notes, and so on. Some may contain forms, some contain PDF attachments of third parties. The client actually types LaTeX, but within templates: only simple LaTeX is used by the client, no math, and there are only three special characters: backslash and braces. You can imagine that common TeX symbols can have a very different meaning in this context: think of $ and % in a bank. The client uses a web application basing on a reduced tinyMCE editor. Usage has to be simple, with low latency, and it needs to be restricted for production. There are just a few special environments, tailored to the corporate identity style to ease use. The output is generated to different channels, such as to a PDF file (with letterhead), printed letters (without letterhead, as it's already pre-printed on the paper), and it needs to be archived.

Besides manually written letters, there are jobs for automated mass production, such as producing account statements each month. The standard processing steps are: generate, format, output, and archive. The engine is plugin-based, using document parameters and templates. Different representations need to be produced, such as draft, online with letterhead, on paper without, as mentioned above. Calibration to in-house printers may be needed, and additionally inserted empty pages when sending to a printshop. Folding machine control has to be implemented, different archiving needs to be supported. Everything has to be done after formatting, since there is a legal requirement to be able to reproduce the archived file in any style on any output channel even after many years. So you need a storage strategy.

In this environment, they use the classic DVI format with `\special` commands. There are different DVI drivers for each purpose. DVI is better suited since the documents are much smaller documents than PDF files, and storage costs money. The interactive preview has to be fast, small jobs have to be processed quickly in high amounts. So they use a TeX `.fmt` file with preprocessed macros. There is not even a document class selection, it's preloaded, no standard packages are used as they are preloaded with the format as well. The packages are very short, with essentially no code, just selection. Compiling a document goes down from 1.5 s to 0.06 s per document, for example, which is a factor of about 25. This is a big thing in mass production. A sample requirement is to generate and format 400,000 documents in a determined time.

To make production even more efficient, tabular material is typeset using `\vbox` and `\hbox` instead of LaTeX tabular environments. So, TeX is very fast in this regard. Jobs can be parallelized. Codes are actually piped into a TeX process. Instead of running TeX on each small file, large container files are generated with, say, 50,000 documents for a single TeX run. The DVI file then gets split in the postprocessing. Every file, all graphics used, and all fonts used get a timestamp for storing and reproducing.

The whole process has to be robust and reliable, fast, and it should use low resources such as memory and storage. The whole setting shows that traditional TeX is still useful today, even outside of academia and publishing.

The next talk by S.K. Venkatesan presented TeX as a 3-stage rocket. The stages are:

- breaking paragraphs into lines;
- making a single long scroll page;
- cutting the scroll into pages with a cookie-cutter algorithm.

With infinitely long pages, footnotes are placed after the text paragraph.

He compared creating paragraphs with CSS in HTML for browsers and as generated by TeX, and spoke about coexistence of TeX and HTML.

After a break, Joseph Wright followed with a talk about the `\parshape` primitive command, with a live demo instead of slides. The LaTeX3 team developed a new interface to `\parshape` based on three different concepts: margins, measure, and cutouts. He demonstrated setting margins to absolute values, and to values relative to the previous paragraph. He showed indenting lines differently within a paragraph, shaping paragraphs and producing cutouts with the new interface. An open challenge is that it's still line-based, but not based on heights of objects, or lengths.

Julien Cretel gave the next talk. It was about functional data structures in TeX. At first, he explained what Haskell is: a purely functional language. He gave an example of quicksort, and said he wanted to do algorithmic things within TeX. One could delegate this to an external program, but we often like to use TeX even if it may not be theoretically the best choice. Many of us like to solve things in TeX, instead of calling Matlab or such. At least it's a pleasant intellectual pursuit. Thus, Julien wants to implement semantics like this in TeX:

```
data Tree a = Empty | Node (Tree a) a (Tree a)
```

He asked the audience for feedback. For example, if TeX should be chosen for the implementation, or it should be done with LaTeX3. He plans to focus on a subset, wants to write algorithms in Haskell and then translate to TeX or LaTeX code.

So it was more an open discussion than a presentation. Maybe we can see an implementation next

year. Some comments from the audience.

- It's easier to implement Haskell in TEX than to implement TEX in Haskell.
- Why implement it in TEX, if you have Haskell already? As above, for the challenge.
- Arthur Reutenauer suggested working with the TEX tries used for implementing hyphenation.
- TEX is Turing complete ... we know, but this doesn't help in the *how*.
- And LuaTEX? Lua is imperative, not functional.

TUG 2015 — day 3 — second part

Hans Hagen gave the next presentation. The main points were

- How far can you go with TEX?
- Do you really want to go that far?

Hans showed fascinating examples, such as TEX-rendered text fed into MetaPost for postprocessing and then re-rendered by TEX for justification.

Another example was about "profiling" lines. For instance, there are two columns, and everything has to be on the grid. TEX has paragraphs, but not a concept of a line. It's pasting hboxes together, with heights and depths. TEX doesn't natively have columns, but you can implement them. He showed an example of boxed columns, all on the grid, including such things as inline fractions. By checking the actual content, lines could stay closer in the grid when heights and depths of elements did not collide. He implemented a profiling mechanism. In the end, he did not use it ... except for this talk.

A second item: many of us know the TEX command `\ignorespaces`. ConTEXt now has commands `\removeunwantedspaces`, `\removepunctuation` and others. Content can be marked as punctuation, or tagged in any way. So, you can remove such marked content, after typesetting.

Finally he demonstrated some peculiarities of an ASCIIMATH implementation, in which, for example, writing `o` twice becomes an infinity sign, with all challenges.

Boris Veytsman continued with a talk about controlling access to information with TEX. This is not only about security, but also simply hiding unneeded information. E.g., technical people may not need to see financial information, and conversely. So a document may contain both, but shows just the relevant part to each kind of reader. Output-level access control may be sufficient.

Meanwhile, regarding security, documents may contain an open part and a part with classified information. In this case, input-level access control is needed. Existing input control in LATEX is via `\includeonly` combined with `\include`. There are disadvantages or restrictions; for example, every such part starts a new page. With a lot of parts or involved reader classes it can quickly get complicated. Separate files may be confusing. A classic approach would be:

```
\newif\ifclassified
\ifclassified\input{classified.tex}\fi
```

Another solution is provided by the `comment` package, which provides environments for information with different audiences.

For output-level control, Boris has written the `multiaudience` package. The `beamer` class offers a similar concept — presentation and a handout mode, so also visibility control. But `multiaudience` has been developed for supporting any number of such modes, a.k.a. visibility classes. This is not for security, but for hiding boring or non-relevant parts. (He later learned about the `tagging` package, which provides similar functionality.)

Regarding security, there must be source level control. Boris showed the new tool `srcredact`, which is a Perl script with an input syntax inspired by docstrip. There are two modes, one to extract text for a partial version and the other to incorporate changes from a partial version. So there's two-way communication.

Finally, Enrico Gregorio showed examples of good and bad TEX code. He talked about the spurious space syndrome, which has bitten all TEX programmers at some time. Or even worse, the "missing required space" syndrome. Missing protection of line endings is classic. TEX friends had fun visually parsing code looking for spaces.

He talked about LATEX3 and showed various `expl3` examples. He strongly recommended `expl3`: even if it adds a thousand lines to load, it's worth it — later it will be part of a format anyway. It does have some disadvantages, e.g., code is much more verbose, and still requires understanding expansion. He thanked the LATEX3 team for their great work on `expl3`. I only can join the thanks.

At the end, we had a question and answer session. One of the most important questions: where and when will be the next TUG meeting? It will be in Toronto, from July 25–27, with optional excursions before and after.

Again, thanks to TUG and the sponsors DANTE e.V. and River Valley Technologies. And especially to Klaus Höppner, who did a great job as organizer!

◇ Stefan Kottwitz
stefan (at) texblog dot net
`http://www.latex-community.org`