



# TUG 2015

Workshops and Presentations on  
L<sup>A</sup>T<sub>E</sub>X, T<sub>E</sub>X, PDF, Unicode, and more

*Welcome Hotel — Darmstadt, Germany*

July 20–22, 2015



dante<sub>e.v.</sub>



## Sponsors

**T<sub>E</sub>X Users Group ■ DANTE e.V.** River Valley Technologies—UK ■  
with special assistance from individual contributors. *Thanks to all!*

## Conference committee

Karl Berry ■ Klaus H $\ddot{o}$ ppner ■ Robin Laakso ■ Volker RW Schaa ■ Joachim Schrod

## Participants

*Leila Akhmadeeva*, Bashkir State Medical University

*Pavneet Arora*, Bolton, ON

*Kaveh Bazargan*, River Valley Technologies, UK

*Stefan Bedacht*, TU-Darmstadt

*Barbara Beeton*, AMS

*Nelson Beebe*, University of Utah

*Denis Bitouzé*, Université du Littoral Côte D'opale

*Johannes Braams*, Zoetermeer, Netherlands

*David Carlisle*, L<sup>A</sup>T<sub>E</sub>X3 Project

*Jennifer Claudio*, Synopsys Outreach Foundation

*Julien Cretel*, University College Cork

*Rajagopal CV*, River Valley Technologies, India

*Christine Detig*, Net & Publication Consultance GmbH

*Michael Doob*, University of Manitoba

*Olaf Drümmer*, callas software GmbH

*Graeme Duffin*, Huddersfield, UK

*Ulrike Fischer*, Mönchengladbach, Germany

*Yukitoshi Fujimura*, Ichikawa-shi, Japan

*Deimantas Galčius*, V<sub>T</sub>E<sub>X</sub>

*Roland Geiger*, Leipzig, Germany

*Paul Gessler*, Milwaukee, WI

*Steve Grathwohl*, Duke University Press

*Gary Gray*, State College, PA

*Enrico Gregorio*, Università de Verona

*Hans Hagen*, Pragma ADE

*Tom Hejda*, Czech Technical University in Prague

*Klaus H $\ddot{o}$ ppner*, TUG and DANTE e.V

*Bogusław Jackowski*, Gdańsk, Poland

*Timm Knape*, Wehrheim, Germany

*Harald König*, Balingen, Germany

*Jonathan Komar*, ITH icoserve

*Stefan Kottwitz*, Lufthansa Industry Solutions

*Reinhard Kotucha*, Hannover, Germany

*Siep Kroonenberg*, Groningen, Netherlands

*Sebastian Krüger*, Berlin, Germany

*Yusuke Kuroki*, Yokohama, Japan

*Dag Langmyhr*, University of Oslo

*Bruno Le Floch*, L<sup>A</sup>T<sub>E</sub>X3 Project

*Manfred Lotz*, DANTE e.V.

*Jerzy Ludwichowski*, Toruń, Poland

*Henri Menke*, Leinfelden-Echterdingen, Germany

*Lothar Meyer-Lerbs*, Bremen, Germany

*Frank Mittelbach*, L<sup>A</sup>T<sub>E</sub>X3 Project

*Ross Moore*, Macquarie University

*Gerd Neugebauer*, Groß-Gerau, Germany

*Heiko Oberdiek*, Sasbach, Germany

*Petr Olsak*, Czech Technical University in Prague

*Steve Peter*, TUG

*Susanne Raab*, ECAP, Uni Erlangen

*Arthur Reutenauer*, Royal Opera House, London

*Will Robertson*, University of Adelaide

*Petra R $\ddot{u}$ be-Pugliese*, CTAN

*Volker RW Schaa*, Darmstadt, Germany

*Joachim Schrod*, Net & Publication Consultance  
GmbH

*Martin Schröder*, Duisburg, Germany

*Torsten Schuetze*, Talheim, Germany

*Herbert Schulz*, Naperville, IL

*Keiichiro Shikano*, Tokyo, Japan

*Martin Sievers*, DANTE e.V.

*Matthew Skala*, Copenhagen, Denmark

*Linas Stonys*, V<sub>T</sub>E<sub>X</sub>

*Piotr Strzelczyk*, Gdańsk, Poland

*S.K. Venkatesan*, TNQ Software

*Boris Veytsman*, George Mason University

*Ulrik Vieth*, Stuttgart, Germany

*Herbert Voß*, DANTE e.V.

*Alan Wetmore*, US Army Research Laboratory

*Joseph Wright*, Northampton, UK

# TUG 2015 — program and information

Sunday, July 19, 3 pm: **walking tour** of Mathildenhoehe and Rosengarten, duration about 2 hr, with our Klaus Höppner as guide.

Sunday, July 19, 7 pm: **informal opening gathering**, LaLucha, Schleiermacher Str. 10-12 (self-pay basis).

Monday, July 20: **post-session TeXShop workshop**, Herb Schulz.

---

## Monday July 20

8:00 am	<i>registration</i>	
8:50 am	Steve Peter, TUG	<i>Opening</i>
9:00 am	Ross Moore, Macquarie Univ.	<i>Semantic enrichment of mathematics using ‘active comments’</i>
9:35 am	Olaf Drümmer, callas software GmbH	<i>PDF/UA — what it is, how users can benefit from it, and how to get it right</i>
10:10 am	Ross Moore and Peter Selinger	<i>Using pdfx.sty for producing validating PDF documents</i>
10:45 am	<i>break</i>	
11:00 am	Joseph Wright, Northampton, UK	<i>X<sub>Y</sub>TeX and LuaTeX: Getting Unicode data into the right places</i>
11:35 am	Will Robertson, Univ. of Adelaide	<i>Reconciling unicode-math with L<sup>A</sup>T<sub>E</sub>X<sub>2</sub><math>\epsilon</math> mathematics</i>
12:10 pm	Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski, Gdańsk, Poland	<i>All the characters we need</i>
12:45 pm	<i>lunch</i>	
2:00 pm	Bogusław Jackowski, Piotr Strzelczyk, and Piotr Pianowski	<i>Six GUST e-foundry math fonts and what next?</i>
2:35 pm	Frank Mittelbach, L <sup>A</sup> T <sub>E</sub> X <sub>3</sub>	<i>Twenty one is only half the truth</i>
3:10 pm	Hans Hagen, Pragma ADE	<i>What if ...</i>
3:45 pm	<i>break</i>	
4:00 pm	Joseph Wright, UK-TUG	<i>State of the (UK-)TeX FAQ</i>
4:35 pm	CTAN team	<i>State of CTAN</i>
4:45 pm	Barbara Beeton, Volker RW Schaa, TUG, DANTE e.V.	<i>In memoriam: Richard Southall, Pierre MacKay, Hermann Zapf</i>

---

## Tuesday July 21

8:55 am	<i>announcements</i>	
9:00 pm	Pavneet Arora, Bolton, ON	<i>Fluss: A flow leak monitoring system</i>
9:35 am	Tom Hejda, Czech Technical University in Prague	<i>Preparing L<sup>A</sup>T<sub>E</sub>X classes/templates for journal articles and university theses</i>
10:10 am	Boris Veytsman and Michael Cohen, George Mason U. and U. of Aizu	<i>A new multibibliography package: nmbib</i>
10:45 am	<i>break</i>	
11:00 am	Boris Veytsman and Leila Akhmadeeva, GMU and Bashkir State Medical Univ.	<i>Trilingual templates for an educational institute in Bashkortostan, Russia</i>
11:35 am	Paul Gessler, Milwaukee, WI	<i>Pretty-printing Git commit history graphs with PGF/TikZ</i>
12:10 am	q&a, TUG meeting	
≈ 12:30 pm	<i>lunch</i>	
2:30 pm	<i>Messel Pit excursion</i>	

---

## Wednesday July 22

8:55 am	<i>announcements</i>	
9:00 am	Kaveh Bazargan and Jagath AR, River Valley Technologies — UK	<i>TeX — After 35 years, still the best solution for modern publishing</i>
9:35 am	Joachim Schrod, DANTE e.V.	<i>DocCenter: T<sub>E</sub>Xing 11 million documents a year</i>
10:10 am	S.K. Venkatesan, TNQ Software	<i>A proposal to construct pagination as a three-step cookie-cutter process</i>
10:45 am	<i>break</i>	
11:00 am	Joseph Wright	<i>Through the \parshape, and what Joseph found there</i>
11:35 am	Julien Cretel, Univ. College Cork	<i>Functional data structures in T<sub>E</sub>X</i>
12:10 pm	Hans Hagen	<i>When to stop ...</i>
12:45 pm	<i>lunch</i>	
1:45 pm	<i>group photo</i>	
2:00 pm	Boris Veytsman	<i>TeX and controlled access to information</i>
2:35 pm	Enrico Gregorio, Univ. de Verona	<i>Recollections of a spurious space catcher</i>
3:10 pm	q&a	
7:00 pm	<i>banquet</i>	at Cafe Rodenstein, in the museum building.

## 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 talks 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<sup>A</sup>T<sub>E</sub>X documents. My experimental package `gittree` adds a convenient interface for producing commit history graphs within L<sup>A</sup>T<sub>E</sub>X documents using the PGF/TikZ graphics language. I will present several examples of `gittree`’s use and discuss continuing development efforts.

## Enrico Gregorio

*Recollections of a spurious space catcher*

Programming in TeX has always posed several hurdles towards pure and error free programs. One of the most often seen errors is related to spurious spaces, missing %s at end of lines that cause head scratching to the user of the macros. But there’s also the opposite problem of missing spaces, which is even subtler, because it can cause unscrutable errors far away. Other typical errors will be presented, related to order of expansion or misused conditionals.

We’ll do an *excursus* through these issues, but this is mostly an excuse for presenting the new paradigm of TeX programming provided by `expl3`, now not restricted to L<sup>A</sup>T<sub>E</sub>X only. Programming TeX with `expl3` can be easy or difficult, depending on the approach to it. Some important examples will be presented, along with solutions for generic tasks which have always taken many lines of code, with duplication of efforts, or short and nice, but unscrutable, code.

## 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 . . .

## Hans Hagen

*When to stop . . .*

A flexible system like TeX permits all kind of solutions for (weird) problems, so as soon you run into a special case it is tempting to come up with a solution. When many such solutions are built into a macro package at some point they start to compete. How far should one go in being nice for users and customers, especially when demands are mostly based on tradition, expectations, and of course subjective, non-rational and disputable artistic considerations? An example is a new optimization available in some ConTeXt mechanisms: profiling. Do we really need it and, if so where and when does it need to be available?

## Tom Hejda

*Preparing L<sup>A</sup>T<sub>E</sub>X classes/templates for journal articles and university theses*

There is a substantial difference between the requirements on a L<sup>A</sup>T<sub>E</sub>X class/template for a scientific journal

and for university theses. The main difference is that a journal class is by definition *restrictive*—the journal has to be very keen on precise look and structure of the articles, whereas the thesis class is by definition *modular*—different theses ask for a slightly different layout and structure, some have appendices and some do not, etc. This contribution discusses the differences and their implications on the class design. Two classes developed by the author (one in collaboration with P. Olšák) will be used as examples.

**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.

**Ross Moore**

*Semantic enrichment of mathematics using ‘active comments’*

A package `mathsem` for pdfL<sup>A</sup>T<sub>E</sub>X implements, on the *input* side, a way to provide semantic meaning for symbols, without adding a large syntactical burden to the specification of a mathematical expression.

It uses a concept of ‘active comment’, allowing the ‘%’ character at the beginning of a new line to become an active token under highly-controlled circumstances. With a strictly defined syntax, words to express the semantic meaning of a variable (‘*x*’ say) can be associated with each occurrence of ‘*x*’ in the expression following.

On the *output* side the words can become content for a tooltip, that ‘pops-up’ near the symbol within a PDF document. Or the words can be used for vocalisation by assistive technology of the meaning of the symbol, thereby enhancing accessibility for the PDF document. This idea extends to:

1. allowing multiple instances of the same symbol have distinct meanings;
2. attaching semantics to macro names as well as character symbols;
3. allowing nested tooltips, for subexpressions;
4. assigning defaults to be attached to symbols and macros, at either global or local levels, to maintain consistency of meaning within extended portions of a given document.

I will present examples of use of this package, and discuss implications and issues that arise, both from the kind of syntax being employed and in the internal coding that was required. A particular application is to provide words for spoken ‘alternative text’, in the context of fully-tagged, accessible, mathematical content within PDF/UA documents.

**Ross Moore and Peter Selinger**

*Using pdfx.sty for producing validating PDF documents*

The `pdfx.sty` package was originally written by River Valley Technologies and used to help produce PDF documents conforming to the PDF/A-1b or PDF/X-1a standards. This involved inclusion of information elements, color profiles, and allowance for XMP metadata elements as required in the corresponding published standards.

We have revised and updated this package so that it can now be used when producing documents to conform with PDF/A-2 and PDF/A-3 (in the ‘b’ and ‘u’ variants), PDF/X-3, PDF/X-4, PDF/X-5 (in all variants), PDF/X-VT and PDF/E. With an appropriate driver engine, it is also able to produce “Tagged PDF”, and some requirements for the stricter ‘a’ variants of PDF/A are also handled.

**Will Robertson**

*Reconciling unicode-math with L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub> mathematics*

The `unicode-math` package was developed before Unicode was widely used, and some of its original features and design decisions have needed to be re-considered as X<sub>Ǝ</sub>L<sup>A</sup>T<sub>E</sub>X and Lua<sup>A</sup>T<sub>E</sub>X have become more popular. In `unicode-math`, there is currently a mismatch between the interface provided for OpenType/Unicode maths fonts and the original interface of L<sup>A</sup>T<sub>E</sub>X itself, since they provide different alphabets for different purposes. Specifically, in L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub> `\mathbf` selects a text-based font and uses it in a maths context. In `unicode-math`, `\mathbf` selects bold math glyphs from the Plane 1 Unicode mathematics range. This has some subtle implications depending on usage. More problematically, `\mathit` produces incorrect spacing with `unicode-math` since Unicode mathematics doesn’t provide glyphs for multi-letter italics, and `unicode-math` overlooked this essential requirement.

Replacing the maths font commands in this way was a poor design decision when considering backwards compatibility, and in 2015 this is causing problems with authors wishing to switch to OpenType Unicode fonts. In this talk, I’ll present what has been provided historically for alphabetic maths fonts in L<sup>A</sup>T<sub>E</sub>X<sub>2<sub>ε</sub></sub>, and what Unicode mathematics provides instead. A brief retrospective will be given on `unicode-math`, emphasising the design decisions that lead to the current incompatibilities, with possible solutions.

**Joachim Schrod**

*DocCenter: T<sub>E</sub>Xing 11 million documents a year*

The talk is an experience report about DocCenter, a product of our company Net & Publication Consultance GmbH (NPC). The report is about one of our reference customers, the online bank 1822direkt, where we handle all typeset written communication with T<sub>E</sub>X. It differs from other T<sub>E</sub>X use cases in that we don’t have to create long complex documents, but small simple standardized ones, lots of them, and very quickly: 11 million documents a year, and the amount is steadily increasing.

This experience report describes the unique challenges that we faced when we introduced the

system in 2008, how we approached them from both a technical perspective and the business processes needed to put such a system into place. In particular, in the technical section it details why DVI output of  $\text{T}_{\text{E}}\text{X}$  processors should not be abandoned — it is not a thing of the past if you go beyond book publishing.

On a side track, technical provisions to support audit trails demanded by bank regulations are outlined. This might be of interest for other industry sections where audit demands about document creation are stringent.

Overall, the talk shows the incredible flexibility of  $\text{T}_{\text{E}}\text{X}$ , and the peculiarities of using the  $\text{T}_{\text{E}}\text{X}$  ecosystem in a commercial environment that doesn't belong to the publishing industry or to academia.

### **Herbert Schulz**

*Workshop: TeXShop tips & tricks*

An interactive workshop for users of TeXShop who want to get to know about some of the lesser-known but useful features of that front end to a  $\text{T}_{\text{E}}\text{X}$  distribution on the Mac.

### **S.K. Venkatesan**

*A proposal to construct pagination as a three-step cookie-cutter process*

Standard typesetting engines do H & J and pagination together as a single step; we suggest that it be split into a three-step process. In the first step, hyphenation and justification is done on each paragraph and in the second step vertical glue is applied to create a scroll with as much height as the content it contains. We show how using cookie-cutter algorithms it is possible to break the content into multiple pages with required results. The flexibility that this three-step procedure offers has many interesting possibilities and layouts. We also show that this produces pages much faster.

### **Boris Veytsman**

*$\text{T}_{\text{E}}\text{X}$  and controlled access to information*

While we in the  $\text{T}_{\text{E}}\text{X}$  community usually strive to make information open, there are cases when controlling access to information is legitimate. We do not want to publish our passwords, medical histories and other sensitive details. Sometimes the information is not confidential, but different audiences require different levels of detail: consider students' and teachers' versions of a textbook.

There are two approaches to this problem in  $\text{T}_{\text{E}}\text{X}$ . Output-level access control means that we have a single  $\text{T}_{\text{E}}\text{X}$  source which can produce different PDF files depending on the compilation options. In some cases we need source-level access control, when we distribute different versions of  $\text{T}_{\text{E}}\text{X}$  file obtained from the same master source.

In this talk we discuss the tools for both these approaches and their implementation in a  $\text{T}_{\text{E}}\text{X}$  system.

### **Boris Veytsman and Michael Cohen**

*A new multibibliography package: nmbib*

Two years ago we presented the multibibliography package, providing multiple lists of citations with

alternate orderings. The nmbib package is a complete refactoring of that program. It offers a broader variety of citation commands, streamlines the creation of bibliographies, ensures compatibility with the natbib package, and offers other improvements.

### **Boris Veytsman and Leila Akhmadeeva**

*Trilingual templates for an educational institute in Bashkortostan, Russia*

Creating document styles for an organization that uses a non-Western script is always a challenge: the organization is supposed to have both Western and non-Western elements in their documents, for instance, putting addresses in both English and its own language. The new Institute of Continuous Professional Medical Development in Ufa poses a special challenge, because we want its templates to be trilingual: with English, Russian and Bashkir elements. The Bashkir language uses a Cyrillic script which is close to but different from Russian Cyrillic. In this talk we discuss the challenges and solutions in the design of document styles for this institute.

### **Joseph Wright**

*Through the  $\backslash\text{parshape}$ , and what Joseph found there*

Paragraphs come in a variety of shapes. While the standard rectangular text block dominates most documents, a variety of document elements need additional treatment. This can range from simple adjustment of margins, for example for quotations through hanging indents for lists to uniquely shaped blocks for one-off applications. This variety may then be coupled to the need to take 'bites' out of the galley, most commonly for 'wrap around' figures.

All of this complexity can be handled by the  $\backslash\text{parshape}$  primitive, but that requires potentially complex manual set up. As part of efforts to develop a new galley model for  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ , we have developed a new interface for setting  $\backslash\text{parshape}$  which separates out the shape into three distinct concepts: margins, measure and cutouts. These different views of the  $\backslash\text{parshape}$  allow complex outcomes to be constructed whilst retaining a readable interface.

### **Joseph Wright**

*$\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$ : Getting Unicode data into the right places*

The 'modern'  $\text{T}_{\text{E}}\text{X}$  engines  $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$  use Unicode as their native input encoding. Handling varied input is much easier using Unicode rather than 8-bit engines, but it does require that the basics are done correctly. Setting up  $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$  for real work relies on data from the Unicode Consortium which defines how different code points (characters) should be handled: that information needs to be made accessible to  $(\text{L}^{\text{A}})\text{T}_{\text{E}}\text{X}$  for a variety of tasks.

In this talk I focus on two separate areas: setting up the various codes  $\text{T}_{\text{E}}\text{X}$  needs when using  $\text{X}_{\text{E}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  and  $\text{LuaT}_{\text{E}}\text{X}$ , and carrying out expandable case changing with these engines. The two areas have their own challenges, but both mean learning about the terrain of Unicode-land.