

TUG 2010 abstracts

Editor’s note: Many of the conference presentations are available at <http://www.river-valley.tv> in video form, thanks to Kaveh Bazargan and River Valley Technologies.

Kaveh Bazargan

Batch Commander: An interactive style writer for T_EX

Batch Commander is a general graphic user interface for any batch system that runs a text file as a batch job and creates an output. It allows quick manipulation of parameters which it writes to an external config file and which it then uses to show the output. The latest incarnation of the system will be shown, with a live demo.

William Cheswick

Ebooks: New challenges for beautiful typesetting

T_EX and other traditional text layout markup languages are predicated on the assumption that the final output format would be known to the nanometer. Extensive computation and clever algorithms let us optimize the presentation for a high standard of quality. But ebooks are here. The iPad has sold more than two million units in under three months, and, combined with other book readers, offers a new way to store and read documents. While these readers offer hope to newspapers (and perhaps doom to many physical bookstores), they are an increasing challenge to high quality text layout. Ebook users are accustomed to selecting text size (for aged eyes and varied reading conditions) and reader orientation. We can’t run T_EX over a document every time a reader shifts position. Do we precompute and download layouts for various devices, orientations, and text sizes? Do we compromise our standards of quality to use HTML- and XML-based solutions? These are new challenges to the T_EX community.

Jean-luc Doumont

Quantum space: Designing pages on grids

Most (L^A)T_EX documents are vertical scrolls: essentially, they place content elements under each other, possibly running the scroll in two columns, but hardly more. With the exception of floats, they basically place items on the page in the order in which these are encountered in the source file: that is, they construct pages by piling up boxes horizontally and vertically, gluing them carefully together to achieve the desired (elastic) spacing.

Effective page design, in contrast, often benefits from a more global approach to the page or spread, one that replaces the scroll paradigm by a true two-dimensional layout. Pages are then usually constructed on an underlying grid, in reference to which the items can be positioned flexibly yet harmoniously. To produce all the documents created by our company (Principiae), I have developed such an approach in T_EX. The session will present the ideas behind both grid designs in general and

the corresponding T_EX macros, and illustrate these ideas with a variety of examples (flyers, brochures, slides, etc.).

Our grid approach works in two steps: first create all the items that will appear on a page or spread (text blocks, illustrations, etc.), then place them in the desired locations on the grid, in any order. In a sense, the macro allow the user to specify, “this block of text goes there, that figure goes here, this title goes there, etc.” — not unlike what page layout software allows, but with the infinitely superior accuracy that T_EX allows. The macros I created to this end are simple, they have worked well for me for many years now, and the resulting documents very often surprise people (“This was done with T_EX?!?”). The grid approach in T_EX is best exemplified with my recent book (sample pages available at <http://www.treesmapsandtheorems.com>), in which grid alignments are pushed to an extreme, but it is behind all our documents, notably slides.

Steve Grathwohl and David Ruddy

Math on the Web: Implementing MathJax in Project Euclid

Project Euclid, a collaboration between Cornell University Library and Duke University Press, provides an online repository and publishing environment for independent mathematics and statistics journals. We discuss the issues surrounding the online display of mathematics at Project Euclid and, more specifically, the implementation of MathJax, an open-source, Ajax-based math display solution supporting both T_EX and MathML notation.

Hans Hagen and Taco Hoekwater

How T_EX and Meta finally got married

A story of T_EX and METAFONT, 1996–2010. At first, information was passed between them via `\write18` and specials, e.g., for backgrounds and layers. Then LuaT_EX started, and it became clear METAPOST should be a library, to avoid overhead. The MPlib project was funded by T_EX user groups and others, with much code cleanup and backends for SVG and self-contained EPS. Ultimately, MP(lib) was married to LuaT_EX. In the happy couple’s future is arbitrary precision arithmetic, complete dynamic memory management, and more. As an ultimate test, Khaled Hosny created PunkNova, an OpenType font using the RAND feature. The text here uses that font.



DON KNUTH 46 DON KNUTH 47 DON KNUTH 48 DON KNUTH 49 DON KNUTH 50 DON KNUTH
51 DON KNUTH 52 DON KNUTH 53 DON KNUTH 54 DON KNUTH 55 DON KNUTH 56 DON
KNUTH 57 DON KNUTH 58 DON KNUTH 59 DON KNUTH 60 DON KNUTH 61 DON KNUTH 62
DON KNUTH 63 ... 1996 ...

John Hobby

Is boxes.mp the right way to draw diagrams?

This talk explains the motivation behind `boxes.mp` and discusses some alternatives. Automatic graph layout can be combined with MetaPost in various ways, but this technology is somewhat hard to control.

Jonathan Kew*TEXworks for newcomers — and what’s new for old hands*

This presentation introduces $\text{T}_{\text{E}}\text{X}$ works, a simple $\text{T}_{\text{E}}\text{X}$ environment based on modern standards — including Unicode text encoding, and PDF output by default — with an uncluttered interface that does not overwhelm the newcomer. It is built using cross-platform, open-source tools and libraries, so as to be available on all today’s major operating systems, with a native “look and feel” for each.

First conceived during discussions at the time of TUG’07 in San Diego, $\text{T}_{\text{E}}\text{X}$ works is now widely available, being included in both $\text{T}_{\text{E}}\text{X}$ Live and $\text{M}\text{i}\text{K}\text{T}_{\text{E}}\text{X}$ for Windows, $\text{M}\text{a}\text{c}\text{T}_{\text{E}}\text{X}$ for Mac OS X, and in packages for various GNU/Linux, *BSD, and similar systems.

Following the first “stable” release (v0.2) in September 2009, the most significant new feature added to the application is a scripting interface. This allows users to extend and enhance the basic program in several ways, both by adding custom menu commands and by providing “hook” scripts that are automatically run at specific times, such as when a file is opened or after a typesetting run finishes. We will look at examples of how $\text{T}_{\text{E}}\text{X}$ works can thus be extended using any of several available scripting languages.

The $\text{T}_{\text{E}}\text{X}$ community is invited to participate in the ongoing development of this environment, either at the level of actual code or in any supporting area, such as document templates or interface localization.

Frank Mittelbach*Exhuming coffins from the last century*

In *The $\text{T}_{\text{E}}\text{X}$ book* Don Knuth poses the following exercise: “Why do you think the author of $\text{T}_{\text{E}}\text{X}$ didn’t make boxes more symmetrical between horizontal and vertical, by allowing reference points to be inside the boundary instead of insisting that the reference point must appear at the left edge of each box?” and gives the following answer: “No applications of such symmetrical boxes to English-language printing were apparent; it seemed pointless to carry extra generality as useless baggage that would rarely if ever be used, merely for the sake of symmetry. In other words, the author wore a computer science cap instead of a mathematician’s mantle on the day that $\text{T}_{\text{E}}\text{X}$ ’s boxes were born. Time will tell whether or not this was a fundamental error!”

In this talk we will show how multiple reference points on boxes allow for a completely different approach to design specification and what can be done to successfully overcome the limitations resulting from Don’s cap worn that day.

Ross Moore*TEX+MathML for Tagged PDF, the next frontier in mathematical typesetting*

This talk will be a follow-on to the introduction to “Tagged PDF” given at last year’s $\text{T}_{\text{E}}\text{X}$ Users Group meeting. Here I’ll present several examples of tagged PDF documents containing real-world mathematical layouts,

which demonstrate the advantages that tagging provides, in terms of long-term Archivability (PDF/A) and Accessibility (PDF/UA) and sharing of content and markup via export to XML.

A script, written in Perl, is under continuing development. This script combines the MathML presentational description of a piece of mathematics with corresponding $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ source for its visual appearance, creating a detailed $\text{T}_{\text{E}}\text{X}$ coding using new primitives that are processed by an enhanced version of $\text{p}\text{d}\text{f}\text{T}_{\text{E}}\text{X}$ to produce fully tagged PDF documents. If time permits we can discuss some of the complications that arise due to differences in the way mathematical structures are handled by $\text{T}_{\text{E}}\text{X}$ and for MathML.

This is joint work with Hàn Thê Thành (River Valley Technologies), author of $\text{p}\text{d}\text{f}\text{T}_{\text{E}}\text{X}$.

Chris Rowley*A brief history of $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ — with a prediction*

Not only brief, but *very* brief and with a lot of personal bias! History with attitude!! Left as unpredictable until the last minute will be both of these: What I mean by $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$; and of course the prediction!

Robert Rundell*Using the Knuth-Plass algorithm to help control widow and orphan lines*

The Knuth-Plass line-breaking algorithm is one of the many exceptional features of $\text{T}_{\text{E}}\text{X}$, taking a paragraph of text and converting it to a vertical list of well-proportioned lines. Through glue and penalty markers $\text{T}_{\text{E}}\text{X}$ gives the user almost complete control over the spacing and look of the paragraph.

However, in some instances $\text{T}_{\text{E}}\text{X}$ does not provide the user quite as rich a set of options to control the vertical list as in other areas. In particular, eliminating widow and orphan lines can require inserting forced break points into the text, break points that can only be found from previous passes of $\text{T}_{\text{E}}\text{X}$. Subsequent changes to the document can require changes to some or all of these manually inserted line or page breaks.

In AML, an experimental typesetting program under development, the Knuth-Plass algorithm is enhanced to find not only the optimal line-break points for a paragraph, but also to give alternate mappings of the paragraph into different numbers of lines (where possible). AML stores these different sets of break points and uses this information, along with natural page break points, to automatically eliminate widow and orphan lines in many cases. Once a bad page break point is detected, AML will backtrack and adjust previous paragraphs to create better page breaks. With far greater memory and processing capabilities than were available at $\text{T}_{\text{E}}\text{X}$ ’s creation, multiple pages can be examined and processed before a final page break needs to be finalized, allowing the overall document layout to be improved. The combination of keeping multiple pages and also keeping alternative paragraph line-breaking sets in memory allows AML to automate and improve this aspect of document typesetting.