TUG 2011 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

Why TEX is more relevant now than ever

TEX is around 30 years old, and was conceived and written before the advent of laser printers, personal computers, PostScript and of course the Internet. At that time the idea of WYSIWYG document editing was just a futuristic idea. When people jumped on the WYSIWYG bandwagon, it was predicted that old technologies such as TEX which used mark-up for text would disappear in time. The advent of the Internet brought mark-up to the attention of the public. Somehow it was acceptable again. The recent move to the semantic web and HTML5 has brought renewed attention to mark-up and the need for clear structure in text. I suggest that we have gone full circle and now realise that mark-up is everything. And TEX, which has the most readable and minimalist mark-up, might just be the best tool today for structured documentation.

Dave Crossland

Freeing fonts for fun and profit

Google Web Fonts (`http://www.google.com/webfonts`) has published hundreds of libre fonts during the past year, at an accelerating pace. Dave Crossland has been driving this through consultancy for Google, and presents his personal opinion about the past, present and future of libre fonts — showcasing the latest designs, designers and tools. (Please note that this talk is entirely the personal opinion of Dave Crossland, and does not represent the views of Google, Inc., in any way.)

CV Radhakrishnan

TEX4ht — A Swiss army knife for TEX

There are several technologies to translate LATEX sources into other markup formats like HTML, XML and MathML. TEX4ht assumes a premier position among them owing to the fact that it makes use of the TEX compiler for translation, which helps to assimilate any complex author macros used in the document. This talk provides an overview of how to configure TEX4ht to output custom markup needed by users. More online at `http://www.cvr.cc/tex/tex4ht`.

Jean-luc Doumont

Integrating TEX and PDF seamlessly in pdfTEX

In its ability to generate graphical elements, TEX is basically limited to horizontal and vertical black rules. Extended versions such as pdfTEX add color options and, especially, the possibility to draw more freely on the page by inserting raw code (PDF code in the case of pdfTEX). Still, these two coding environments — TEX and PDF — are too often regarded as disjoint. It would be nice to integrate them seamlessly, for example, to use in PDF code a color or a dimension assigned or calculated in TEX. This presentation points out the challenges of such a consistent and transparent TEX–PDF integration, proposes a set of solutions, and illustrates how these solutions help create graphs flexibly or design pages consistently on a grid.

Frank Mittelbach

LATEX3 architecture

This talk discusses the architecture of LATEX3, starting with the initial ideas dating back to the early '90s. Using an example covering the whole production cycle it is shown that several different roles with different requirements are needed to turn some draft initial manuscript into a final product. The purpose of the LATEX3 architecture is to provide adequate support for these different needs and to resolve or at least mediate conflicts between them.

While the basic building blocks of this architecture were identified long ago, an initial implementation in 1992 showed that it was impossible to use them in practice due to limitations in the processing power of the underlying engines at the time. Furthermore, several ideas that were toyed with at the time — though not wrong as such — were immature and not fully thought through. As a result the project gave up on the broader redesign and instead focused on producing a consolidated LATEX version largely based on the architecture of LATEX 2.09. This fairly successful endeavor, labeled LATEX 2ε, is still the current standard LATEX.

So why is it still relevant? Basically because the drivers and goals that led to the new architecture are issues that haven't been successfully resolved by other typesetting systems. The difference from the situation from the '90s is that processing power in the underlying engine has increased so much that it has become feasible to implement this architecture in TEX (or rather one of its successors). The other reason is that since then further work has been undertaken, refining many of the initially immature ideas. The result is a coherent vision for a future typesetting system based on the principles of TEX and LATEX but moving them to the next level.

The talk discusses the separation of concerns as propagated by the architecture: between logical structure, design layer and the coding and implementation support. At the same time it is shown that for high-quality results this separation needs to be accompanied by built-in support for formatting adjustments and how this is supported by the architecture.

For design support the architecture provides two major complementary concepts: templates and context management. The use of design templates offers abstractions from which real designs can be derived through customization of parameters. The second approach is a general concept for managing design variations based on actual element relationships within a document. For each concept, the theory is discussed and a short live demonstration is given.

Ross Moore
Further advances toward Tagged PDF for mathematics

This is the 3rd presentation on on-going efforts to develop the ability to generated Tagged PDF output using pdfTEX, in conjunction with other software tools. In this talk I'll show how recent improvements to Adobe Reader and Adobe Acrobat Pro software have increased the usefulness of Tagged PDF documents, containing a MathML description of TEX-typeset mathematical content.

In particular, by careful specification of the words to be "Read Out Loud", mathematical content can be conveyed quite effectively to the visually impaired. Also, using Adobe's Acrobat Pro as the PDF browser, the ability to export to XML means that a fully marked-up, with MathML for the mathematics, version of the PDF document's contents can be obtained from the same file that displays the high-quality typeset visual appearance.

Examples will be shown of diverse mathematical content, generated automatically from standard LATEX, along with suitably generated MathML descriptions.

Rishi
Creating magical PDF documents with pdfTEX

PDF has a rich specification, but Adobe Distiller does not exploit all these specifications. We'll demonstrate how pdfTEX can create useful PDF files that are difficult or impossible to create using other technologies. Examples: PDFs showing differences in two TEX source files; PDFs with useful pop-up tools; a simple but useful composite PDF for comparing two nearly identical PDF files.

Karel Skoupý
Typesetting fancy multilingual phrase books with LuaTEX

We used TEX for typesetting a series of phrase books with a fancy graphical design. Each book contained the same content for a different language pair. There were several dozen of them semi-automatically generated, and thanks to the way that the language data was organized and thanks to TEX as a typesetting engine this process was very time and cost-effective.

We have developed interesting TEX macro modules and used many advanced features of pdfTEX and LuaTEX to meet the challenges raised by the graphical design and by some non-Latin script languages. We will show the general structure and discuss some interesting problems and their pdfTEX/LuaTEX solutions.

Karel Skoupý
Data structures in ε-TEX

To construct macro packages, TEX is used as a programming language. Unlike general programming languages it lacks complex data structures. We present the experience of providing record and array data structures and the supporting operations using ε-TEX features. They were successfully applied in real projects for parametrization and as a base for a special table module involving complex dimension calculations. We will show how the abstraction level provided by more powerful data structures can simplify and unify TEX low-level code.

Petr Sojka
Why TEX math search is more relevant now than ever

TEX is around 30 years old, and was conceived and written before the advent of MathML, not to mention the Internet. At that time the idea of indexing and searching mathematics was just a futuristic idea. When people jumped on the Google bandwagon, it was predicted that old technologies such as TEX mark-up for math would disappear in time (it is not used for tokenization and indexing properly). The advent of the Internet and W3C brought mark-up and global search to the attention of the public. Somehow it was acceptable again. The recent move to the semantic search and MathML has brought renewed attention to the need of unambiguous canonical math representation in texts.

As part of the project of building the European Digital Mathematics Library (`http://www.eudml.eu`) we have designed and implemented a math search engine, MIaS (`http://nlp.fi.muni.cz/projekty/eudml/mias`). It currently indexes and searches more than 160,000,000 formulae originally written by authors in TEX in their scientific papers. We will present the system and will discuss the ways towards a global math search engine based on the TEX math notation.

Dominik Wujastyk
My father's book: Typesetting and publishing a family memoir

In 2010, I typeset a 650-page book of memoirs, political essays, and biographical sketches written by my 97-year-old father. The book is in the Polish language, and was published by the University of Lublin. For the design and typesetting I made choices that stylistically echoed my father's life-long links with Malta and Poland. Due to financial restrictions at the University of Lublin, I worked out a cost-effective pathway for printing and distribution using an American web-based printing and distribution service. The final result is of a high standard, and has been gratifyingly well received by all parties. Some niggles remain, however, regarding publicity and distribution. In this paper, I shall describe my choices and discoveries in producing my father's book.

Dominik Wujastyk
Typesetting Sanskrit in various alphabets: XƎLATEX, TEC files, hyphenation, and even XML

The XƎTEX extended TEX engine provides a wealth of sophisticated features, and meets many of the longstanding needs of people working with multilingual or multi-script texts. I shall describe the use of XƎLATEX for typesetting Sanskrit, with both Roman- and Devānagarī-script inputs, and Roman- and Devānagarī-script outputs. I shall describe the complexities of getting differently hyphenated Sanskrit in different scripts. Finally, I shall offer an example of a free IBM XML tool that uses a XƎLATEX TEC file to auto-convert Sanskrit between Roman and Devānagarī for screen display via HTML. If all this sounds a bit messy, it is. But the results are sometimes quite amazing, and open up exciting possibilities for the beautiful printing of Indian texts.