Abstracts

Editor's note: Many of the conference presentations are available at `http://www.river-valley.tv/conferences/tug2008` in video form, thanks to Kaveh Bazargan and River Valley Technologies.

— — ∗ — —

MathTran and TEX as a web service
Jonathan Fine

In 2006/7 I developed and set up the public MathTran web service `www.mathtran.org`. This was done with funding from JISC and the Open University. It provides translation of TEX-notation formulas into high-quality bitmaps. In April 2008 it served 48,000 images a day for a growing range of sites. After tuning, the server could provide about 2 million images a day. It takes about 10 milliseconds to process a small formula.

MathTran images contain rich metadata, including the TEX source for the image and the `dvi` and `log` outputs due to that source. This makes it straightforward to edit and resize such images, or convert them to another format, such as SVG or PostScript.

MathTran, used with JavaScript, makes it considerably easier to put mathematics on a web page. In particular, the author of the page does not need to install any special software, and does not have to store thousands of image files.

The MathTran project is now focussed on the authoring of mathematical content. It has produced a prototype instant preview document editor. Funded by the 2008 Google Summer of Code, Christoph Hafemeister is developing JavaScript to provide autocompletion for commands and analysis of TEX errors, all integrated with an online help system embedded in the web page. Separate work is focussed on developing MathTran plugins for WYSIWYG editor web-page components.

This talk will present progress and prospects. It will also discuss some of the broader implications for the TEX community and software, such as

- People using TEX without installing TEX on their machine.
- Help components for web pages.
- Integration with third-party products.
- Standards for TEX-notation mathematics.
- Learning and teaching TEX.

Why we need LATEX3
Jonathan Fine

The LATEX3 project started in 1992. Since then, much has changed. XML has replaced SGML and along with X/HTML has become the dominant markup language. CSS has replaced explicit style attributes in HTML pages, and is now a widely understood and used language for specifying design. Internet access is considerably more widespread, the web has gone from 1.0 to 2.0, Microsoft has replaced IBM, Linux went from nothing in 1991 to an open-source standard, and Google is on track to replace Microsoft.

In 1997 the LATEX3 project said that LATEX3 would provide:

- A new *input document syntax*, that aligns with SGML/XML.
- A new *class file interface*, that aligns with SGML/XML.
- A new *style-designer interface* that can work with a *visually-oriented, menu-driven specification system.*
- An *effective interactive help system* for document authors.
- *Thoroughly documented* and *modular* source code.

These goals are still worth achieving. This talk will focus on some recent progress, and in particular:

- Use of key-value syntax within tags.
- Separation of parsing from processing.
- An improved development environment.
- On-line interactive help systems.

LuaTEX, MPlib, and random punk
Hans Hagen

We use new LuaTEX and MPlib features to generate random characters from Donald Knuth's punk font. This was the 'surprise' talk on the TUG'08 program. The full paper will appear in a future issue of MAPS.

Image handling in LuaTEX
Hartmut Henkel

The Lua language allows for defining new variable types, and LuaTEX uses this concept for types like 'node' and 'font'. In this talk an image library as part of the LuaTEX engine is presented, built around a new 'image' type, giving extended image handling and embedding capabilities. The image primitives inherited from pdfTEX are still fully functional for compatibility.

First the process of image embedding and its limitations using the pdfTEX primitives is described. Then, after a short introduction about Lua libraries, the 'image' type of LuaTEX is presented together with the set of new Lua functions for image handling, and their use is illustrated by examples. As work is still ongoing, possible future extensions are discussed as well.

LuaTEX: what has been done, and what will be done
Taco Hoekwater

At TUG 2007 in San Diego, the first beta version of LuaTEX was presented. This year the team presents a version where significant parts of the TEX-Lua API are stable. This talk will give an overview of the components that make up LuaTEX: what libraries do we have and what callbacks are available. The team has some ideas about the next stages of development and these will be presented as well.

The galley Module or: How I Learned to Stop Worrying and Love the Whatsit

Morten Høgholm

TEX has a well-deserved good reputation for its line breaking algorithm, which has found its way into other software over the years. When it comes to inter-paragraph material such as penalties, skips and whatsits, things start getting murky as TEX provides little help in this area, especially on the main vertical list where most of the action is.

This article describes the galley module which seeks to control line breaking as well as taking care of inter-paragraph material being added at the right time. In other words, galley can assist packages such as breqn, whose goal is to construct paragraph shapes on the fly while taking current ones into account as well as ensuring the output routine doesn't get tricked by penalties, skips and whatsits appearing in places where they could allow breakpoints where none are intended.

Minion Math: The design of a new math font family

Johannes Küster

"Minion Math" is a set of mathematical fonts I have developed over the past 6 years. Designed as an add-on package to Adobe's Minion Pro font family, it consists of 20 OpenType fonts (4 weights, times 5 optical sizes). In future releases it will cover the complete Unicode math symbols, and more.

In the design I tried to remove constraints and to avoid flaws and shortcomings of other math fonts, with the aim of creating the most comprehensive and versatile set of math fonts to date.

Here I present the main design principles of Minion Math, and the most important design decisions I took. I will show samples of the fonts and will compare the fonts to other math fonts as well.

Trademark attribution: Minion is either a trademark or registered trademark of Adobe Systems Incorporated in the United States and/or other countries and used under license.

Multiple simultaneous galleys: A simpler model for electronic documents

Blanca Mancilla, John Plaice, Toby Rahilly

We present a general model for electronic documents supporting parallel containers of content, tied together through link components. This model is usable for a wide range of documents, including simple textual documents with footnotes and floats, complex critical editions with multiple levels of footnotes and critical apparatus, maps with multiple layers of visual presentation, and music scores.

This model is inspired from the C++ Standard Template Library, whose basis is that Containers + Iterators + Algorithms = Programs. In our approach, the 'iterators' are pointers into the parallel containers, keeping track of callouts for notes, floats, and parallel links.

The data structures required for this model are remarkably simple, and will allow the rapid development of many different kinds of algorithms.

Windows of opportunity: A (biased) personal history of two decades of LATEX development — Are there lessons to be learned?

Frank Mittelbach

Looking back at twenty-odd years involvement in LATEX development and maintenance the author highlights the (in his opinion) most important milestones and pitfalls.

- What are significant events that came at the right moment?
- Which important events came at the wrong moment?
- What were the biggest failures and why?

From this data the article attempts to draw conclusions as to how the future of LATEX could be shaped in a way beneficial to everybody involved and what needs to happen to make this possible.

A pragmatic toolchain: TEX and friends and friends of friends

Steve Peter

In this talk, we present the toolchain used to produce the award-winning Pragmatic Bookshelf titles (`http://www.pragprog.com`) and examine some of the pleasures and pitfalls encountered using TEX, XML, XSLT, Ruby and other open technologies.

Parallel typesetting

Toby Rahilly, John Plaice, Blanca Mancilla

We present the general mechanism by which logical content, arranged in multiple interacting containers, can be typeset into a set of visual substrates. The overall algorithm is iterative, with the successive iterations refining a multidimensional context that parameterises the behavior of the algorithm.

Each iteration consists of three parts. First, each visual substrate is informed which parts of which logical containers are to be placed thereon. Second, in parallel, the content placed in the substrates is typeset. Third, the resulting layout in each substrate is assessed for goodness, thereby resulting in the refinement to the overall context.

In the talk, we will present the theory and the practice behind this algorithm.

Three typefaces for mathematics

Daniel Rhatigan

This paper examines the issues involved in the design of typefaces for mathematics. After a brief discussion of some of the typographic and technical requirements of maths composition, three case studies in the development of maths types are presented: Times 4-line Mathematics Series 569, a complement to the Times New Roman text types as set with Monotype equipment; American Mathematical Society Euler, an experimental design intended to contrast against non-mathematical typefaces set with TEX; and Cambria Math, designed in concert with a new text face to take advantage of new Microsoft solutions for screen display and maths composition.

In all three cases, the typefaces were created to show the capabilities of new technological solutions for setting maths. The technical advances inherent in each font are shown to be as central to its function as its visual characteristics.

By looking at each typeface and technology in turn, and then comparing and contrasting the issues that are addressed in each case, it becomes apparent that even though certain challenges are overcome with technical advances, the need to consider the specific behaviours of type in a maths setting remains constant.

See `http://www.ultrasparky.org/school` for the complete paper and other typographical items.

Medical pedigrees with TEX and PSTricks: New advances and challenges

Boris Veytsman, Leila Akhmadeeva

A medical pedigree is an important tool for researchers, clinicians, students and patients. It helps to diagnose many hereditary diseases, estimate risks for family members, etc. Recently we reported a comprehensive package for automatic pedigree drawing. Since then we have extended the algorithm for a number of complex cases, including correct drawing of consanguinic relationships, twins and many others.

In this talk we review the facilities of the current version of the program and the new challenges in computer-aided drawing of medical pedigrees.

We try to make the talk interesting to TEXnicians by discussing the experience of design a TEX-based application working in a “real world”.

Observations of a TEXnician for hire

Boris Veytsman

Several years ago the author was tempted by extremely cheap rates for *TUGboat* advertisements, and declared *urbi et orbi* he was a TEX consultant. This audacious step led to many interesting experiences. Some results of this work were published on CTAN and listed at `http://borisv.lk.net/latex.html` (the list includes both commissioned packages and the ones I wrote for my own purposes).

In this talk I report on my past projects, big and small, and discuss the lessons learned from my journeys in the fascinating world of publishers, editors and authors. I describe writing book and journal styles, communication with customers and other issues relevant for TEX consulting.

TUG Institutional Members

Aalborg University, Department of Mathematical Sciences, *Aalborg*, *Denmark*

American Mathematical Society, *Providence*, *Rhode Island*

Aware Software, Inc., *Midland Park*, *New Jersey*

Banca d’Italia, *Roma*, *Italy*

Center for Computing Sciences, *Bowie*, *Maryland*

Certicom Corp., *Mississauga*, *Ontario*, *Canada*

CSTUG, *Praha*, *Czech Republic*

Florida State University, School of Computational Science and Information Technology, *Tallahassee*, *Florida*

IBM Corporation, T J Watson Research Center, *Yorktown*, *New York*

Institute for Defense Analyses, Center for Communications Research, *Princeton*, *New Jersey*

MacKichan Software, *Washington/New Mexico*, *USA*

Marquette University, Department of Mathematics, Statistics and Computer Science, *Milwaukee*, *Wisconsin*

Masaryk University, Faculty of Informatics, *Brno*, *Czech Republic*

Moravian College, Department of Mathematics and Computer Science, *Bethlehem*, *Pennsylvania*

MOSEK ApS, *Copenhagen*, *Denmark*

New York University, Academic Computing Facility, *New York*, *New York*

Princeton University, Department of Mathematics, *Princeton*, *New Jersey*

Springer-Verlag Heidelberg, *Heidelberg*, *Germany*

Stanford Linear Accelerator Center (SLAC), *Stanford*, *California*

Stanford University, Computer Science Department, *Stanford*, *California*

Stockholm University, Department of Mathematics, *Stockholm*, *Sweden*

Université Laval, *Ste-Foy*, *Québec*, *Canada*

Universiti Tun Hussein Onn Malaysia, Pusat Teknologi Maklumat, *Batu Pahat*, *Johor*, *Malaysia*

University College, Cork, Computer Centre, *Cork*, *Ireland*

University of Delaware, Computing and Network Services, *Newark*, *Delaware*

University of Oslo, Institute of Informatics, *Blindern*, *Oslo*, *Norway*

Vanderbilt University, *Nashville*, *Tennessee*