Abstracts

(See the 'Random comments' column on p. 377 for publication information on items marked with ''. Ed.)*

Three dimensional graphics with Sketch
David M. Allen

Sketch is a 3D scene description translator by Eugene K. Ressler. Its web page is at `http://www.frontiernet.net/~eugene.ressler`.

Sketch is a small, simple system for producing line drawings of two- or three-dimensional solid objects and scenes. Sketch produces finely wrought, mathematically based illustrations with no extraneous detail. It does not do photo-realistic scenes. The input language is reminiscent of PSTricks, so will be easy to learn for current PSTricks users.

Sketch output was PSTricks code until recently. In addition to PSTricks, Sketch now understands Ti*k*Z/PGF options (key/value pairs) and generates Ti*k*Z output. Some advantages are that Ti*k*Z works directly with pdfLaTeX and supports transparency. Happily, the depth sort hidden surface algorithm used by Sketch is perfectly compatible with transparent polygons.

This presentation gives a brief overview of Sketch from the perspective of a beginning user. It gives some detailed examples that introduce a substantial set of commands. There are also examples from the Sketch distribution.

From TeX to XML: The legacy of techexplorer and the future of math on the Web
Don DeLand

In 1997, IBM Research first released techexplorer, a web browser plugin for rendering TeX markup directly within browsers. Since Integre took over techexplorer development in 2003 there have been relatively few advances in browser technology, but tremendous developments in collaboration tools and other web-based applications. This talk gives a brief history of the techexplorer project and explains why its development has shifted away from TeX to its current focus on native XML/MathML authoring. Although delivering math in web browsers continues to be a frustrating process, "Web 2.0" holds substantial promise for a new generation of web-based applications that support mathematics.

Long-time preservation strategies for TeX-sourced content
Paulo Ney de Souza

The amount of published material in the world has grown exponentially since Gutenberg's invention, with a rate of doubling every 7 1/2 years right now. Electronic publishing will only increase this rate, posing new challenges for the long-time preservation of records and usability for the future.

TeX has changed us into our own typists and even graphics designers sometimes, but at the same time has provided the best strategy for preservation of scientific content we have. This talk will examine some of these strategies and how MSP — a non-profit scientific publisher — has used it to improve usability of journals over time.

LuaTeX Attributes: The new kid on the block*
Hans Hagen

When you switch fonts in TeX, normal grouping keeps changes to another font local to the group. But there is more than fonts. Most macro packages provide color support and in ConTeXt we also support some PDF related features like outlines and hidden invisible ink. These features all share a common problem: we need to keep track of their state in the page stream and across pages and splitting content also takes some care. A maybe less obvious example is hyperlinks, which are natively supported by pdfTeX (although ConTeXt does it slightly differently).

In order to make such features easier (and more robust) to implement, LuaTeX provides attributes. These behave like fonts but are by design agnostic as to what they represent. They travel with the nodes (each node can have attributes) and it's up to the macro package to make sure that the intended behaviour takes place. For this, Lua code is used in combination with processing node lists, either by using callbacks or by postprocessing boxes.

In this talk I will explain what attributes are, and how they can be of use to macro writers.

Zapfino: Hermann's torture test for TeX*
Hans Hagen

Over the next couple of years TeXies have to explore the new landscape of OpenType fonts. Most of the implementation details will be hidden beyond user interfaces of macro packages. However this does not hide the potential mess that users can invoke when they start enabling or disabling features related to fonts.

Thanks to the Oriental TeX project Taco can spend substantial time on coding LuaTeX which in turn means that I have lots of testing and protyping on my plate. We also spend much time on discussing the interfaces and extensions to the program and due to this as well as realistic testing LuaTeX develops rapidly. However, in order to fulfill the requirements of the Oriental TeX project we need to be able to typeset high quality Arabic. Since I'm more familiar with Latin and since I had the Zapfino Pro handwriting font waiting for me in OpenType format I decided to use that font as benchmark for advanced node processing in LuaTeX. It proved to be a worthy contender. In the process we were able to optimize node support in LuaTEX and it also triggered reimplementing the OpenType tables (from FontForge format 1 to format 2).

In this presentation I will discuss how we deal with advanced features that are part of OpenType fonts like Zapfino. I will also explain how such features are implemented in Lua and TEX code.

CritTEXt: The critical-edition module for ConTEXt*

Idris Hamid

TEX has become a very important tool in the preparation of critical editions of texts. In particular, `EDMAC` for plain TEX, by Lavagnino and Wujastyk, has come to the rescue of many an editor of texts (myself included). In the last two years the LATEX world has produced at least two general-purpose packages for the preparation of critical editions: L`EDMAC` (a port of `EDMAC` with even additional features) and ednotes. With its extensive and high-level configurability, typographic flexibility, and database capabilities — not to mention its user-friendly and *consistent* interface — ConTEXt is perhaps the most natural typesetting platform for a full-featured and easy-to-use user interface for authors and typesetters who need to produce a sophisticated camera-ready critical edition. With the maturation of projects like LuaTEX and Oriental TEX well underway, the capacity of TEX for even more efficient production and configuration of the highest quality of critical texts in multiple languages is almost at hand.

This document outlines the high-level interface to the critical-text editing module for ConTEXt, called *CritTEXt*. The module is in pre-alpha status and the entire interface can be expected to undergo extensive changes in the coming weeks, especially as LuaTEX matures.

On the technical level, preparation of the critical edition involves three things:

- the main text;
- the apparatus;
- the rest of the book.

The critical edition module we are developing is concerned primarily with preparing and typesetting the apparatus. This module is meant to be consistent with the rest of ConTEXt so that the integration of the apparatus with the main text and the rest of the book is seamless.

In preparation of the apparatus we distinguish two things, *elemental structure* and *typographic structure.*

1. *Elemental Structure*:
   Each building block of the apparatus, up to and including the apparatus itself and relevant parts of the main text, constitutes an *element.* Given an element, it performs a distinct function. For example, a textual *variant* of a passage is one element of the apparatus, the source for that variant is another.
2. *Typographic Structure*:
   Given an element of the apparatus, the editor and/or typesetter must decide where (layout) and how (style) to place that element on the page. Will we use footnotes, endnotes, marginal notes, or a separate volume entirely for the presentation of the apparatus? What symbol will we use to separate the *lemma* of a given *entry class* from its associated *comment*?

`EDMAC` and other critical edition packages for TEX mix elemental structure and typographic structure. In what follows we try to precisely identify the various components of the elemental structure of the critical edition. Then, when we consider typesetting in ConTEXt, one may choose a plethora of options for the typographic structure. This allows for much more flexibility as we will see. Another advantage of this approach is that it gives us a framework for easy translation of apparatus data to and from XML.

The Text Encoding Initiative (TEI) has defined a special XML schema for critical editions (`tei-c.org/release/doc/tei-p5-doc/html/TC.html`). Some parts of our structure roughly correspond to aspects of the TEI schema for critical editions. That schema is more detailed and less precise than we need; for us at this juncture it is more useful to have a clearer and more precisely defined elemental structure. Then, once the typographic structure interface is in place, a user can define those elements of the TEI schema that are needed.

Towards an ontology of Arabic-script typography: An implementation strategy for Oriental TEX*

Idris Hamid

At the core of the Oriental TEX project is the implementation of a scheme for typesetting culturally authentic Arabic-script. Such cultural authenticity has, in general, been on the decline since the onset of digital typography in the Arabic-script world. The coming-of-age of Unicode as well as the new OpenType font standard is playing some part in alleviating this situation at the font level. Yet, although digital typography carries within it by far much the greater potential for high-quality Arabic typesetting, Arabic digital typography has yet to meet the standards set by lead-press typesetting in the Arabic-script world.

In this document we present the outlines of a new approach to the *ontology* or *typology* of Arabic-script typography. Ontology is the philosophical study of the categories of being. In this case, our universe of discourse is the Arabic script, particularly its typeset instantiation. An ontology of Arabic-script typesetting, then, identifies and organizes the categories of the manifestation of Arabic script in the context of typography.

Our analysis is organized along the following lines:

- The three-fold challenge of Arabic-script digital typography: First-, second-, and third-order analysis of Arabic script
- A critical examination of Arabic script and Unicode
- First-order analysis of the Naskh script

- Historical overview of Naskh typography
- Second-order analysis of the Naskh script: Macro-typography
- Third-order analysis of the Naskh script: Micro-typography

An experimental CTAN upload process
Jim Hefferon

Some improvements to the package upload facility at CTAN have the potential to make the process smoother. We'll talk about some of the features being developed and tested, and also have a demo.
(This paper will appear in the EuroBachoTEX 2007 proceedings. Ed.)

The breqn package: revised and revived
Morten Høgholm

The breqn package was originally developed by Michael Downes of the American Mathematical Society to facilitate automatic line-breaking of displayed math expressions. It has recently undergone some restructuring as part of a thesis project and is now actively maintained again. This presentation gives an overview of the current status of breqn, what it can do and what it can't (yet) do, and finally what the immediate, mid-term and long-term goals are.

About Lua
Roberto Ierusalimschy

Lua is an embeddable scripting language that aims for simplicity, small size, portability, and performance. Unlike most other scripting languages, Lua has a strong focus on embeddability, favoring a development style where parts of an application are written in a "hard" language (such as C or C++) and parts are written in Lua. Currently Lua is used in a vast range of applications, being regarded as the leading scripting language in the game industry.

In this talk I will give an overview of the language, covering not only the technical aspects of the language but also its origins back in 1993, its evolution, and its current status.

XƎTEX Live
Jonathan Kew

With the release of TEX Live 2007, the XƎTEX engine has "come of age" and entered the mainstream of the TEX world. XƎTEX, which provides built-in Unicode and OpenType support, is now a standard part of a TEX Live installation, and thus is readily available to any user who installs this distribution.

This presentation will show how users can take advantage of XƎTEX to easily use additional fonts in TEX or LATEX documents, with no complex installation or setup procedures. It will also show how non-Latin scripts such as Chinese, Arabic, Devanagari, and many others can be typeset just as easily as English, thanks to full Unicode support throughout the system.

In addition, some of the newest developments in XƎTEX (beyond the TEX Live 2007 release) will be discussed and demonstrated. These include support for the Graphite rendering technology for complex scripts; extensions that can simplify Chinese/Japanese character spacing and mixed-script typesetting; and more complete Unicode math support.
(This paper will appear in the EuroBachoTEX 2007 proceedings. Ed.)

Everything you wanted to know about PDF but were afraid to ask
Leonard Rosenthol

If you ever wondered just what makes PDF tick, come to this presentation by Adobe's PDF Standards Evangelist. You'll learn about the many features of PDF, from page content to interactive features to 3D, and how it all fits together. PDF is on track to becoming an ISO standard.

Vistas for TEX
Chris Rowley

This is a polemic in favour of liberating the core typesetting structures and algorithms around which TEX is built from the monolithic superstructure of the program called tex and its derivatives such as xetex, luatex etc.

Although the aims of the programme of activity advocated here are have a lot in common with those behind the very exciting and active luaTEX project, the route I support seems to me to be to be very different from embedding the whole of the TEX system within such a vastly more complex monolith (sic), along with its many intrusions (sic) into but a single instance of the ancient base rock of TEX! Of course, this is by no means all that luatex promises to give us, hence the importance and fascination for me of the approach taken by the luaTEX project.

Pursuing the paleontological metaphor well beyond its total collapse, my plan can be thought of as providing many tools for influencing the evolution of automated tpesetting without the need to fossilize the perectly preserved skeleton of the whole ancestral dinosaur.

Incorporating LATEX text into graphics and presentations with LATEXiT
Ari Stern

It is often a challenge to combine text created in LATEX with content from other software, such as graphics and presentation software, while maintaining a consistent typographical style; for instance, matching text labels in mathematical figures to the main text. This demo will show how this can be done easily using LATEXiT, a free utility for Mac OS X included with the MacTEX distribution. Other examples will include 2D and 3D mathematical figures in Adobe Illustrator, as well as presentations using Apple Keynote.