TEX: Rejoining the mainstream

Jonathan Fine

UK Unix Users Group

Birmingham

8 August 2009

Slides at mathtran docs on SVN on Sourceforge

Jon Udell's insight on TEX, LATEX and the web

In 2000 Jon Udell, in a report on *Internet Groupware for Scientific Collaboration* wrote:

> *TEX and LATEX define scientific publishing for a generation of scientists. But these formats don't integrate directly into the shared spaces of the Web.*

In July 2009, after encountering the Polymath project, he wrote:

> *Why didn't I see, then, that the crux of the issue wasn't XML and MathML and SVG, but rather the ability to* **"integrate directly into the shared spaces of the Web"** *? And that what ought to be integrated directly was the typesetting language already familiar to mathematicians, namely LATEX?*

http://blog.jonudell.net/2009/07/31/polymath-equals-user-innovatio/

Knuth's vision

[T]he TEX research project . . . was driven by two major goals . . . **quality** *. . . the best . . .* **archival** *. . . 100 years*

I'm not going to design a programming language; I want to have just a typesetting language.

In some sense I put in many of TEX's programming features only after kicking and screaming [from users].

Now, if there were a universal simple interpretive language that was common to other systems, naturally I would have latched onto that right away.

Let us regard these systems [TEX and Metafont] as fixed points, which should give the same results 100 years from now that they produce today.

Quotes from **Digital Typography**, pages 559, 648, 648, 649, 571.

Knuth's request: use another name

When he announced, in 1990, that his work on developing TEX, Metafont and Computer Modern had come to an end, (except for extremely serious bugfixes), Don Knuth wrote (DT, p571):

> *I welcome continued research that will lead to alternative systems that can typeset documents better than TEX is able to do. But the authors of such systems must think of another name.*
>
> *That is all I ask, after devoting a substantial portion of my life to the creation of these systems. I sincerely hope that the members of TUG will help me to enforce these wishes, by putting severe pressure on any person or group who produces an incompatible system and calls it TEX or METAFONT or Computer Modern — no matter how slight the incompatibility might seem.*

Introduction and overview

This talk is a simplified history of TEX: past, present and future.

Mainstream was print, now includes Web. The basic ideas are:

- TEX was mainstream end-user software.
- TEX is no longer in mainstream of open source.
- TEX can rejoin the mainstream.
- Helping TEX rejoin the mainstream.

By the word 'TEX' we mean one of:

- The program `tex` written by Don Knuth.
- A backslash and braces markup language.
- Software and other resources associated with TEX.
- A community of users and developers.

But not (as requested by DEK) some other typesetting program.

LaTeX and Python documentation compared

Please visit http://www.latex-project.org/guides/

- First link on page is to books (mostly by team members).
- Second link is to 33 page PDF [2001, stale addresses].
- Most of the other links are to 3rd party PDF files.
- HTML links to wikibooks and Andy Roberts sites.
- Doesn't link to Indian TUG Tutorial, Nicolas Talbot, ...

Please visit http://docs.python.org/

- Docs available in HTML and PDF
- Search page for docs
- Online Global Module Index and General Index
- Permalinks to page fragments
- Syntax highlighted code
- Copyright, the Python Software Foundation

LaTeX3 project — digging deeper

- Started in 1993 or so (predates XML, Google, . . .)
- No-one is using LaTeX3 for typesetting
- Last year, LaTeX3 source placed on SVN server, but . . .
- They say it's **explicitly forbidden** to publish LaTeX3 code
- Uses proprietary license (Debian accepted, not OSI-approved)
- Current activity focused new macro programming interface

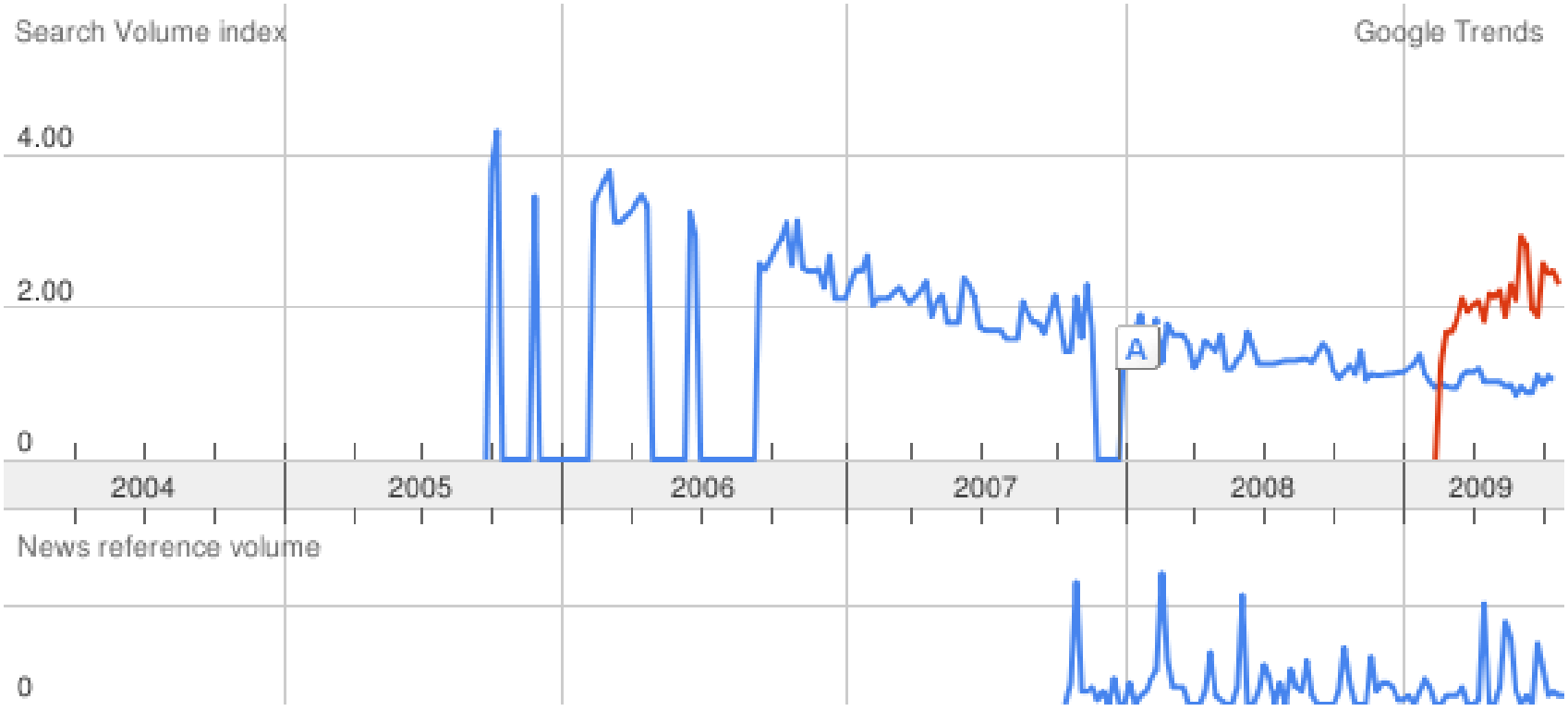
Here's an example of the old and new interface:

```
\def\mymacro #1{\setbox #1\hbox\bgroup}  % Old
\cs_new_nopar:Npn \hbox_set_inline_begin:N  #1 { % New
  \tex_setbox:D #1 \tex_hbox:D \c_group_begin_token }
```

It doesn't even get named parameters (instead of #1).

Google trends for MathML

http://www.google.com/trends?q=MathML,+lmgtfy



MathML in blue (from 2005), lmgtfy in red (from 2009).

Terry Tao's blog and the Polymath project

Terry Tao is a young and extremely eminent mathematician. He's an early adopter. His blog http://terrytao.wordpress.com is published in book form. Typeset in LaTeX, of course.

Along with Tim Gowers, Gil Kalai and Michael Nielsen, he's set up http://polymathprojects.org. (And see also http://www.tricki.org)

Polymath is all interested people across the world working together on a problem of common interest. It's using a WordPress blog (with LaTeX support).

Computer industry commentator Jon Udell wrote of Polymath:

> *Why didn't I see, then [2000], that the crux of the issue wasn't XML and MathML and SVG, but rather the ability to "integrate directly into the shared spaces of the Web"? And that what ought to be integrated directly was the typesetting language already familiar to mathematicians, namely LaTeX?*

XeTeX — scripts.sil.org/XeTeX

Used in production by the SIL linguistics institute.

An open source typesetting program based on a merger of Donald Knuth's TEX system with Unicode and modern (OpenType and AAT) font technologies.

The XeTeX typesetting system is cross-platform. It provides the same Unicode and OpenType font support on GNU/Linux, Mac OS/X and Windows (but no AAT font support outside Mac OS/X).

Developer Jonathan Kew now works for Mozilla, on internationalisation.

LuaTEX: www.luatex.org

Recall Don Knuth's goal was to create *just a typesetting language*.

LuaTEX is going in the opposite direction.

- It *embeds* Lua into an extension of TEX.
- Describes itself as a 'variant of TEX' (loud tutting).
- For many (most) users, not better than XeTeX.
- Embed rather than extend is a *big mistake*.
- Gives Lua access to typesetting internals.
- Big performance and memory hit.
- Requires extensive rewrite of Don Knuth's code for TEX.

In my view, XeTeX provides now all the benefits LuaTEX will provide, except: Lua is better than the TEX macro language.

However, there are other ways to obtain this benefit.

Embed and extend compared: client and server

We *extend* a client to connect to a server. For example, in Python to connect to an SQL database we import an extension module.

We can also *embed* Python in an SQL database server. This allows SQL queries to use Python, for example to filter a result set.

With databases, almost always we extend (with an extension module). It's rare to use Python embedded in a database server. Apache is the only common use I know of Python embedded (via `mod_python`).

It's common to import into Python many extension modules. But there's no easy way to combine two embeddings.

LuaTEX chooses embed because TEX has an embedded macro language (which was probably right then). The key thing now is to provide a service, not a better embedded language.

MathTran: Flickr for formulas

Wikipedia uses LaTeX for complex math formulas, and provides a nice page of input/output examples. But we can do better ...

http://en.wikipedia.org/wiki/Help:Displaying_a_formula

Service is good. MathTran provides TeX typesetting (and production of images) as a public web service.

http://www.mathtran.org/

- Funded by JISC and the Open University.
- Developed mostly by speaker.
- Provide instant preview editor for formulas.
- Runs TeX as a daemon.
- Create images that are too cheap to cache.
- Now uses Pinax/Django to provide social website.
- Requires interactive online documentation.
- Hope soon to allow authoring of such documentation.

Python documents: latex2html and Sphinx

Python documentation was produced using customized latex2html (a Perl script). Provides both HTML and PDF output. LaTeX source documents.

Since Python 2.6, documentation authored in *Restructured Text* and translated by Sphinx into HTML, Windows HTML Help, and LaTeX for PDF. See http://sphinx.pocoo.org/

Sphinx says *many of its strengths come from the power and straightforwardness of reStructuredText and its parsing and translating suite, the Docutils*.

Python did not adopt plasTeX, a successor to latex2html which implements TeX's markup language, in Python.

Sphinx and plasTeX are pointers to LaTeX succession.

Conclusions

The underlying theme is to more easily *integrate directly into the shared spaces of the Web* (Jon Udell, 2009).

- Interactive online LaTeX documentation.
- Tools for authoring and displaying mathematics on the web.
- Copy-and-paste (standards for TeX-encoded math).
- Evolve LaTeX into standard wiki-like markup language.
- Provide web services.
- Print backend for web pages (a *big win*).

A big problem in achieving this is legacy: documents, macros and ways of working. So we'll also need

- Regression tests for documents.
- Enthusiastic and committed developers.
- Engagement with young people.

What next: Online LaTeX documentation

A Google search for latex+quote produces Sheldon Green's hypertext help from 1995 as top result.

ConTeXt has http://wiki.contextgarden.net/ — much better than the LaTeX offering. It also allow try out ConTeXt without installing it. This is a step in the right direction.

We need interactive and social help for TeX and LaTeX.

What next: Firefox Mathematics plugin

The idea here is to provide a toolbar button, or some other interface, that allows you to add, view and edit mathematics on a web page.

Mozilla's Ubiquity is designed to make it easy to do such things.

http://labs.mozilla.com/projects/ubiquity/

Key developer for Ubiquity is Aza Raskin, head of user experience at Mozilla Labs.

Aza Raskin was previously with Humanized, who created a Windows 'desktop assistant' Enso (similar, for the aged, to Borland Sidekick).

Enso provided a *TEX anywhere* component that uses MathTran's web service for typesetting (and has a neat *unrender* function).

UK Math Content Workshop - Wed 9th Sept 2009

One-day workshop on technical issues related to mathematical content in electronic media. Three main themes

1. Content related technical problems in supporting eLearning in mathematics
2. Standards related to digitisation of mathematics research literature
3. Formulas and equations in otherwise non-mathematical content

The aims of the workshop are improved practice, more collaboration and reuse of software, and a published roadmap to inform and guide future work in this area.

To take place at The Open University, Milton Keynes, UK. Possible due to funding from JISC and the OU. Organised by Jonathan Fine (OU), David McKain (Edinburgh) and Petr Sojka (Masayrk).