

TUG 2014 in Portland

David S. Latchman

This year’s TUG conference took place in the city of Portland, the home of the \TeX Users Group. The city straddles the Willamette River near its confluence with the Columbia River, physically dividing the city into east and west Portland. The downtown area, where the Mark Spencer Hotel is located and where the conference was held, is in the west. Commonly known as the “City of Roses”, Portland features many famous rose gardens, most prominently the International Rose Test Garden. There is certainly a lot to do, from the micro-breweries, restaurants and food trucks found all over the city. If hands-on science demonstrations is something you might be interested in, the Oregon Museum of Science and Industry (OMSI) is worth a visit.¹ You can also visit the USS Blueback, a diesel–electric fast attack submarine moored in the river by the museum.



Figure 1: View from the SW Hawthorne Bridge

1 OMSI

The Oregon Museum of Science and Industry is a science and technology museum that features a variety of hands-on permanent exhibits focused on natural history, sciences, industry and technology. The museum was established in 1944, when it displayed its first collection of natural history objects at the Portland Hotel. In 1949, the museum moved to its first location on 908 NE Hassalo but by 1955, growing annual attendance entailed the building of a new site at Washington Park (this building is now occupied by the Portland Children’s Museum). Museum attendance would continue to rise and by the mid-1990s, continued expansion at Washington Park was deemed infeasible. The museum moved to the Station L power plant, donated by Portland General Electric, where it resides today.

¹ Editor’s note: OMSI has a place in \TeX history. The manual for the OMSI Pascal compiler was produced in \TeX in 1982 by Dave Kellerman and Barry Smith, before they went off to form Kellerman & Smith, which handled the first \TeX distribution for VAX/VMS systems.

David S. Latchman



Figure 2: An animatronic Dilophosaurus



Figure 3: The USS Blueback

1.1 The USS Blueback (SS-581)

In 1994, the museum purchased the USS Blueback. This submarine appeared in the first season of the 1970s TV show, “Hawaii Five-O”. She also appeared in the 1990 film, “The Hunt for Red October”, where her crew was paid \$50US to cut their hair and put on Soviet Navy uniforms.

The USS Blueback is a Barbel-class submarine. This diesel–electric fast-attack submarine was one of the first production warships to utilize a teardrop shaped hull. The Barbel class submarines were designed to dive to a depth of over 700 feet, with a crush depth of 1050 feet. At the time of construction, the Blueback and her two sisters were the most technologically advanced submarines in the world.

The Blueback was the last non-nuclear submarine to join the United States Navy, entering service on 15 October 1959. She was also the last conventionally-powered combat-capable submarine to be decommissioned, leaving the US Navy with a fully nuclear submarine fleet.

2 Monday, July 28

2.1 “Fake spaces” with pdf \TeX — the best of both worlds

The first presentation was by Ross Moore of Macquarie University. Moore began his talk by explaining that we can’t resize text and have it reflow in

PDFs in the same way text can be resized in an Internet browser or word processor. This isn't a problem with the PDF but rather with \TeX itself.

Moore explained that when Donald Knuth created \TeX , he chose to omit space characters from the output and carefully position the start of each word and punctuation character. Though this allowed for better full-justification and typesetting, the 'missing' spaces prevent on-the-fly justification and re-flow.

Moore said that this can be solved by getting the right spaces into the document — a special space character that can be stretched as needed. These interword space characters are available as new primitives in \pdf\TeX as of \TeX Live 2014 and can be controlled when and where they are used, e.g., they're not needed in mathematical content. These new word spaces conform to ISO-19005 standards and meet the PDF/A archival standards.

2.2 A modern regression test suite for \TeX programming

Regression testing is the process of testing changes made to a system to ensure that older code still works and isn't in conflict with new code. One problem discussed in this talk by Frank Mittelbach is the hidden dependencies in \LaTeX ; packages can hook onto various layers or replace macros.

For many years, $\text{\LaTeX} 2_{\epsilon}$ used a custom Perl script to perform regression testing on a large number of files to ensure that any changes do not break anything. This type of test is also useful for highlighting changes to \TeX Live. One of the steps in the development of $\text{\LaTeX} 3$ was to produce a similar system to ensure that the development process was as smooth as possible.

Mittelbach discussed the various ways in which this can be done, e.g. making use of the \TeX log file. One problem is that the log contains much unnecessary information. Today, \TeX ships with \Lua\LaTeX . The included standalone Lua interpreter allows users to write platform-independent scripts without the need to install additional frameworks. Joseph Wright has rewritten the $\text{\LaTeX} 3$ build scripts in Lua to avoid maintaining two separate systems. This should allow for an easy-to-use regression test suite for the \TeX community.

2.3 Liberate \TeX , top part: JSBox

Doug McKenna discussed his monumental work in progress, JSBox, a \TeX -language interpreter written in portable C. McKenna hopes to remove some of the limitations of the \TeX engine by providing integrated support for 21-bit Unicode, namespaces, and OpenType, to name a few. To illustrate what is

possible with this new interpreter, McKenna showed how a fractal image could be placed and dynamically rendered in a \TeX document.

2.4 Liberate \TeX , bottom part: literac

McKenna's second presentation focused on his program `literac`, a command line C program that enables literate commenting by converting source code written in languages that use C-style commenting syntax into a \LaTeX document.

Within the comments there are special commands to support common typesetting tasks. The `literac.c` is itself documented using the literate style and can typeset itself into its own manual. McKenna demonstrated how this can be done.

2.5 A call for standards on the way to internationalization of \TeX

Though \TeX can typeset documents in hundreds of languages, there has been little to no internal support with regard to language. Paulo Ney de Souza explained how the introduction of standards like ISO-639, ISO-15924, ISO-3166-1 and RFC-5646 can help \TeX users, especially those for whom English is not the preferred language.

2.6 MathBook XML

Robert Beezer discussed the need for a source format that can capture the structure of a scholarly document and convert it into a variety of formats. Though XML may have a bad reputation, Beezer contended that it can be compact and a good solution; MathBook XML is both lightweight and practical and simple enough for authors to create mathematical and scientific texts. He discussed a current project to convert XML to \LaTeX and HTML. As the project matures, conversions to other formats such as Sage and SageMathCloud worksheets and iPython notebooks will be developed.

2.7 Converting structured \LaTeX to other formats

David Farmer explained that while PDF is good for printing, reading on screen is less than ideal, especially when a reader clicks on a link to view a reference or equation — the page "jumps around" making it difficult for a reader to visually scan a document.

Farmer said that his project, a collaboration with Beezer, to convert mathematics papers and textbooks from \LaTeX to HTML can solve this problem and provide an alternative to online PDF documents. When a reader clicks on a link, the reference appears in a box below which improves readability.

2.8 Will static CSS someday suffice for online rendering of profiled L^AT_EX

In this talk, William Hammond discussed the use of static CSS and its potential to render L^AT_EX on the web. The use of MathJax has shown that it is possible for CSS, along with JavaScript, to render mathematics online in HTML pages. Hammond explained that the problem with this method isn't quality but speed.

In recent times, CSS has become more powerful and though quality of rendering remains an issue, CSS by itself has become an adequate fallback option for online presentations. Hammond showed what CSS is capable of.

2.9 The easy way to define customized XML-to-L^AT_EX converters

In his talk, Keiichiro Shikano discussed the need for an XML-to-L^AT_EX converter. Shikano explained that though XML syntax may be considered ugly, it can be used to typeset books by applying a preferred L^AT_EX style.

This means that `xm12tex` is not a single converter application but rather a framework that adds a presentation layer in L^AT_EX; rules relating each XML element directly to a L^AT_EX command or environment is needed. Shikano demonstrated a straightforward and intuitive tool written in a dialect of Scheme to do this. The tool has been used to create and publish dozens of books.

2.10 Creating mathematical jigsaw puzzles using T_EX and friends

Julian Gilbey first demonstrated a free Windows-based WYSIWYG software, Formulator Tarsia, that can be used to author mathematical puzzles. These puzzles are made of triangles and squares with questions and answers printed along the edges. Some of these puzzles were distributed to the audience for their solving pleasure.

Gilbey then described a similar T_EX and Python-based system he created which can run on most operating systems.

2.11 SUTRA — A workflow for representing signals

Pavneet Arora talked about a practical workflow for representing signals of an electrical system.

3 Tuesday, July 29

3.1 Building T_EX Live

Karl Berry gave the first presentation of the second day, an overview of T_EX Live's construction. Berry explained that T_EX Live follows the T_EX Directory

Structure (TDS) which segregates files by type rather than placing packages in separate folders of their own. E.g., the `.cls` or `.sty` files are located in one folder while the documentation is located in another. Berry explained what this means for users and package designers. He ended with a discussion of how Knuth's original T_EX is built from source.

3.2 T_EX Live Utility: A slightly-shiny Mac interface for `tlmgr`

In his talk, Adam Maxwell discussed his program T_EX Live Utility, a Mac OS X graphical user interface for the T_EX Live Manager command-line tool. Maxwell explained that many Mac users appreciated a native Mac GUI interface.

Maxwell discussed some of the goals of a GUI interface. `tlmgr` works behind the scenes and does most of the heavy lifting. The program uses a model/view/controller design to give users a consistent Mac-like experience.

3.3 MacT_EX design philosophy vs. TeXShop design philosophy

Richard Koch explained that MacT_EX is an unmodified installation of T_EX Live that, when installed, is fully configured and ready to use. TeXShop, on the other hand, is a Macintosh T_EX front-end, written in Cocoa that incorporates some modern features that Apple recently introduced. This means that TeXShop is not cross-platform, unlike MacT_EX.

Programming the front-end in Cocoa means that it automatically supports and can take advantage of many new Mac features, such as support for automatic saving and reloading of documents and the Retina display.

3.4 Metapolator: Why Metafont is finally catching on

This talk by Dave Crossland was a followup to his 2008 TUG presentation, "Why didn't METAFONT catch on?" Crossland introduced the Metapolator project, a tool allowing typeface designers to harness the power of METAFONT's parametric capabilities without writing any METAFONT code. Crossland gave several demonstrations.

3.5 Recent additions to T_EX's font repertoire

In this talk, Michael Sharpe examined some of the newer additions and changes to the font families in L^AT_EX. Most of the additions have been new Roman text fonts. Sharpe also talked about the issues of matching math fonts.

3.6 Plotly: Collaborative, interactive, and online plotting with \TeX

Etienne Tétreault-Pinard introduced Plotly, an interactive online graphics platform that allows a user to embed and export graphics for publication. In the same way a core goal of \TeX is to allow anyone to produce high-quality documents with minimal effort, Plotly hopes to do the same with graphics.

Users can use data from Python, MATLAB, R, and Excel, among others, to create graphs. Graphs are rendered using JavaScript and annotations are done using \TeX in a MathJax display engine. As all data is stored in the cloud, users can not only get the same results across multiple platforms, they can share graphs online, publicly or privately.

3.7 Texinfo visits a garden

Joseph Hogg discussed using GNU Texinfo to document a collection of about 400 plants in the Herb Garden at The Huntington in San Marino, CA. The Huntington maintains a plant list as a resource for gardeners, docents, and visitors. This list was done in Word before The Huntington moved over to \LaTeX .

Though the Herb Garden is one of the smaller gardens, the plant list is updated every six months. By using Texinfo and other free software, Hogg and other volunteers are able to list and map on a grid the plants in the garden, which can then be downloaded or printed as a PDF. Visitors to the garden can view this PDF as they tour the gardens.

3.8 \TeX and copyediting

SK Venkatesan and CV Rajagopal described how their `copyediting` package can be used to help copyeditors. It provides macros to help editors make corrections and improve consistency.

3.9 An output routine for an illustrated book

In this talk, Boris Veytsman showed how \LaTeX can be used to produce an illustrated book. Veytsman explains that producing an illustrated book in \TeX is difficult because the engine's algorithms assume there is a lot of text occasionally interrupted by an image; \TeX assumes that the text tells the story. In an illustrated book, on the other hand, the illustrations tell the story rather than the text. Veytsman discussed a method for creating an illustrated book using \TeX .

3.10 Creating a \LaTeX class file using a graphical interface

Kaveh Bazargan explained that writing \LaTeX class files is a difficult and specialized job that requires an

intimate knowledge of \TeX . This makes it difficult for users to get into the deeper workings of \LaTeX as, quite often, potential programmers don't know all the options and inner workings of the individual packages that go into a class file. A graphical user interface (GUI) might help address this problem.

In his talk, Bazargan demonstrated such a GUI. A programmer can drag and drop, even order packages in the user interface and see the various options that are available. This system, named Batch Commander and created using LiveCode, dynamically updates the output so the effects of any changes are immediately visible. It is data-driven so that any package or parameter can be supported.

3.11 \LaTeX 3 and `expl3` in 2014: Recent developments

In the last talk of the day, Will Robertson and Frank Mittelbach reported on \LaTeX 3 and `expl3`. Robertson explained that \LaTeX 3 and `expl3` are not the same and neither is \LaTeX 3 the next version of \LaTeX ; rather, \LaTeX 3 is an alternative to \LaTeX where some of its ideas can be layered onto \LaTeX 2 ϵ . `expl3` originally stood for experimental \LaTeX 3. Though this has changed, the name has stuck.

Robertson explained that `expl3`'s goal is to make it easier to write \LaTeX packages and is now being used by many people "in the wild" and for many different package types. Both Robertson and Mittelbach discussed some ideas for rounding out the feature set.

4 Wednesday, July 30

4.1 Dynamic documents

Dynamic documents are documents that can be updated as the data changes. David Allen talked about using R, `tikzdevice`, `knitr` and \LaTeX to accomplish this. As an example, Allen used polling results from Kentucky's current senate election race to demonstrate how the above components can be used to track, graph and generate a report of both candidates' current standing.

4.2 Using animations within \LaTeX documents

Though \TeX exists primarily for typesetting books, the software has evolved to where animations can be placed inside a document. Michael Doob discussed how this can be done, and how it can help students solve mathematical problems.

4.3 \LaTeX in the classroom

Dan Raies discussed his experiences of using \LaTeX to teach mathematics to his undergraduate students.

Raies said that using L^AT_EX is great for students to submit assignments, and learning to use L^AT_EX is an important skill for their academic studies. Raies showed the audience some techniques for helping students get past common problems.

4.4 Typesetting figures for computer science

Colleagues Andrew Mertz, William Slough and Nancy Van Cleave highlighted some L^AT_EX graphics packages to produce informative, high-quality diagrams and figures in computer science.

4.5 Moving an online book to paper

Jim Hefferon talked about his experiences bringing his *Linear Algebra* book into print. The book was initially available only online, as a free download, but Hefferon was eventually convinced to produce a printed book that students could purchase. Hefferon described the self-publishing landscape and many different price considerations.

4.6 PDF files both to print and read

Kaveh Bazargan talked about techniques one can use to make a PDF suitable for both print and online reading. Instead of creating two separate PDFs, Bazargan mentioned various packages that will allow a user to pack the information for both into one file. One example is the use of PDF layers. A reader can select between color and black and white images for screen reading or printing respectively. These layers can be shown or hidden as needed.

4.7 Typography and readability: An experiment with post-stroke patients

Reading is a complex mental process and cognitive impairments may influence how well a patient can read and understand instructions. In this talk, Leyla Akhmadeeva and Boris Veytsman discussed their experiments on whether typeface design has an effect on readability on post-stroke patients.

Tests were done comparing legibility of the Para-type serif and sans serif fonts with post-stroke patients. Reading speed and comprehension were compared with results from healthy volunteers. If typography does indeed influence comprehension, publishers of books for post-stroke patients should take this into account. Though the sample size was small, the study found that differences in reading speeds between both font types were minimal. It is possible that, even after a stroke, the human brain remains remarkably adaptable.

Tracy Kidder

4.8 A quarter century of naïve use and abuse of L^AT_EX

Alan Wetmore recounted his experiences with T_EX and how it has evolved to what it is today. When Wetmore first encountered T_EX, it was much more difficult to install and use but the strong community around T_EX made this easier. Today, installing and maintaining T_EX is much easier and its capabilities have grown.

4.9 Another wiki? OMG why?

Ward Cunningham, the inventor of the wiki, talked about and demonstrated rebuilding the wiki concept to solve more complex sharing situations addressing some of society's toughest problems.

4.10 Trying to write about technical topics

In the final talk, Pulitzer Prize-winning writer Tracy Kidder described some of his experiences with a computer engineering team as they created a next-generation computer, eventually becoming his book *The Soul of a New Machine*. He also mentioned his current work in progress, where he is coming back to the technical world for the first time since *Soul*.

◇ David S. Latchman
Bakersfield, CA 93309
david.latchman (at)
texnical-designs dot com
<http://texnical-designs.com>

Visiting TUG 2014

Tracy Kidder

I came to the conference partly to spend time with my friend Karl, partly to meet others who could help to educate me on the history of computer programming, and partly to find out what TUG is all about. For the past year and a half I have been hanging around in a corner of the world of commercial software development. I thought it would be both interesting and refreshing to be among people who are working on the extension and maintenance of a venerable piece of free software. During many of the formal talks, I found myself quite confused, lost in the technical details, but pleasantly so. One could do much worse than spend time among smart people who are obsessed with fonts. In short, the conference was everything I'd hoped for. And I am grateful for the great kindness and hospitality I was shown. ■