

---

**TUG 2013 abstracts**

Editor's note: The slides and other samples for many of the talks are posted at <http://tug.org/tug2013/program.html>.

— \* —

**Jin-Hwan CHO**

*A case study on T<sub>E</sub>X's superior power: Giving different colors to building blocks of Korean syllables*

In 2007 Dave Walden, the instigator and primary interviewer of TUG's Interview Corner, tossed a tricky question at me: "One of the concerns of many people in the T<sub>E</sub>X world is that T<sub>E</sub>X is relatively unknown in the larger worlds of typesetting and word processing, compared with commercial programs such as Adobe's InDesign and Microsoft Word. How do you see the future of T<sub>E</sub>X when it comes to Asian languages?" Since then, it has been my mission to find a wonderful answer, that is, a T<sub>E</sub>X product which other programs cannot reproduce.

Unicode contains 11,172 modern Korean syllables, all of which are composed by only 24 building blocks. In this talk, I will show an interesting T<sub>E</sub>X example containing a large number of Korean syllables each of which is grouped by building blocks of different colors. Nobody, of course, would try to reproduce this example with other commercial programs.

**Hans HAGEN**

*How we try to make working with T<sub>E</sub>X comfortable*

Just as book and music production is under pressure, so is the way we produce documents. We're accustomed to instant rendering in browsers and even if WYSIWYG is not that important when most of the time is spent on writing instead of messing with the look and feel, there is the comfort factor to keep in mind. The last few years I have spent quite some time on a comfortable edit-proofing cycle: from advanced syntax highlighting to fast rendering. Do such things matter and is it worth the effort?

**Hans HAGEN**

*How we move(d) on with math*

Given the amount of time I spend on LuaT<sub>E</sub>X and ConT<sub>E</sub>Xt I occasionally ask myself if it really makes sense to do this. The answer to that question is determined by several factors. Probably the most important factor is the user base: what are their demands, how do they like to code, what control do they want, and therefore, where can these tools be of help? Another factor is relevance: can this combination do certain things better than other tools? One area that has always drawn users is math typesetting.

So, how up to date is T<sub>E</sub>X in that respect? Can we still claim victory there? Did we evolve well? Can we survive?

**Shizuya HAKUTA**

*LISP on T<sub>E</sub>X: A LISP interpreter written using T<sub>E</sub>X macros*

Although T<sub>E</sub>X macros are useful, writing macros can be difficult for novice users. To make T<sub>E</sub>X easier to use, there is some research combining T<sub>E</sub>X and another programming language. Approaches have included calling an external interpreter and embedding an additional language in a member of the T<sub>E</sub>X engine family. We have taken yet another approach, possible because T<sub>E</sub>X is a Turing machine: implementing a language processor with T<sub>E</sub>X macros. The result, called 'LISP on T<sub>E</sub>X', allows us to embed LISP scripts in a L<sup>A</sup>T<sub>E</sub>X document. The interpreter is written entirely with T<sub>E</sub>X macros and it is available through CTAN (<http://ctan.org/pkg/lisp-on-tex>). In this talk, we would like to illustrate how to use it and contrast it with LuaT<sub>E</sub>X, PerlT<sub>E</sub>X, and related approaches.

**Yoshifumi MAEDA & Masataka KANEKO**

*Making math textbooks and materials with T<sub>E</sub>X+KETpic+hyperlinks*

Because of its precision and simplicity, the graphics capability originally present in T<sub>E</sub>X should have great potential in mathematics education. However, it seems to be burdensome for typical T<sub>E</sub>X users to fully utilize such capability. Although including graphical images generated by using computer algebra systems (CAS) is a typical alternate approach, the resulting documents tend to become inefficient for practical use in a classroom.

The CAS macro package named KETpic is one of the most hopeful candidates for realizing convenient and efficient use of T<sub>E</sub>X graphics. For instance, it enables us to edit high-quality math textbooks and materials containing: 2D-graphics which are precise in shape and length, and 3D-graphics which are readily understandable.

In this talk, we emphasize that the programmability of KETpic (associated with CAS) and T<sub>E</sub>X could make the use of T<sub>E</sub>X more flexible. For example, many documents with graphics can be readily generated by using both for-loop programming and "meta commands" of KETpic, and those documents can be readily linked also by using the `hyperref` package (connected to the "hypertextlink" function).

Such unified use of T<sub>E</sub>X graphics and T<sub>E</sub>X programming through KETpic might be applicable to many other situations in math classrooms, and should enhance the possibility of T<sub>E</sub>X use in education.

**Yasuhide MINODA***TEX in educational institutes*

Tokyo Educational Institute (Tetsuryokukai) is a preparatory school specializing in the entrance exam for Tokyo University. We use  $\text{\TeX}$  for our texts, workbooks, other handouts, and even for internal documents and memorandums.

We used other software in the past, but we switched to  $\text{\TeX}$  and converted our original documents (over 100,000 pages) into  $\text{\TeX}$  files over the last few years.

In Tetsuryokukai, we now have over 200 teachers, with various levels of computer skill, so in order to introduce  $\text{\TeX}$  we:

- developed related software (automatic installer,  $\text{\TeX}2\text{img}$  and so on),
- prepared various style files,
- educate and motivate teachers.

In this presentation, I would like to present what we have been doing in our company, in the hope that it can be an interesting and helpful example of introducing  $\text{\TeX}$  throughout an institution, especially in the field of education.

**Frank MITTELBACH***The stony road to complex page layout*

We discuss the challenges encountered in attempting to automate complex page layout. What are the real life use cases? How can they be approached? What remains unresolved after more than three decades of  $\text{\TeX}$  programming efforts and why? (Slides and video at <http://www.latex-project.org/papers/>.)

**Frank MITTELBACH***L<sup>A</sup>T<sub>E</sub>X<sub>3</sub>: Using the layers*

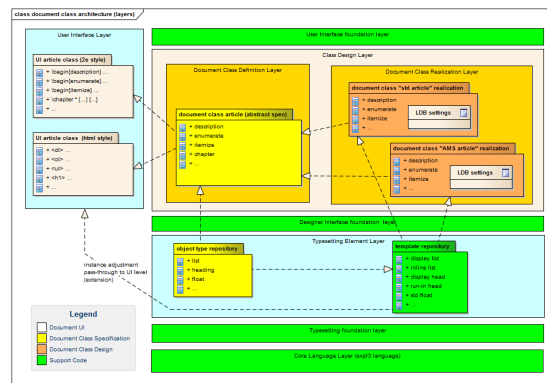
In this talk we will briefly present the architecture of  $\text{\LaTeX}3$  with its four conceptual layers: document interface layer; typesetting element layer; document design layer; programming layer.

We will then look in some detail at `xparse` — a  $\text{\LaTeX}2_{\epsilon}$ -like user interface, as an example of the  $\text{\LaTeX}3$  document interface layer, that can already be used to provide extended functionality for existing  $\text{\LaTeX}2_{\epsilon}$  documents and packages.

We conclude with a brief tour of `exp13`, the foundation layer for  $\text{\LaTeX}3$  that provides the basis for all higher-level modules of  $\text{\LaTeX}3$  but can also be usefully deployed to develop packages for  $\text{\LaTeX}2_{\epsilon}$ .

The `exp13` language is by now in a stable state and gets more and more traction outside the  $\text{\LaTeX}3$  development work, which can be seen, for example, by its use in a growing number of answers on the question and answer portal <http://tex.stackexchange.com> and in the appearance of  $\text{\LaTeX}2_{\epsilon}$  packages that

use it for programming. (Text and diagram based on a previous talk by Joseph WRIGHT.)

**Ross MOORE***Making mathematical content accessible using Tagged PDF and L<sup>A</sup>T<sub>E</sub>X*

‘Tagged PDF’ (more specifically PDF/UA) is the method developed by Adobe to allow the Web Content Accessibility Guidelines (WCAG 1.0, WCAG 2.0) to be satisfied within PDF documents. In this talk I will show the latest developments on using an extended version of pdf $\text{\TeX}$  to allow Tagged PDF documents to be produced, satisfying both PDF/A (Archivability) and PDF/UA (Universal Accessibility).

I’ll show examples which include quite complicated mathematical expressions, fully tagged with MathML, which can be ‘Read Aloud’ in Adobe’s Acrobat and free Reader software. These will include ‘real-world’ documents containing such features as top-matter, nested list environments, logos, watermarks and other pagination artifacts, tabular material within mathematics, and some support of colour and text-styling. A special math-indexing feature has been developed, which allows the result of processing by external programs to be identified and reused in successive  $\text{\LaTeX}$  runs. This indexing feature leads to significant time savings when developing a full document over many processing runs.

The full paper is available at <http://ceur-ws.org/Vol-1010/paper-01.pdf>.

**Keiichiro SHIKANO***Indexing makes your book perfect*

Most of you already know how to make books using  $\text{\LaTeX}$ . And some of you might know how to make back-of-the-book indexes with  $\text{\LaTeX}$ . However, are you ready to worry about how the index of your book should be? Or, if you have already gone through a trouble of writing or editing books, have you actually taken advantage of indexing in your work?

The index, which would be inserted at the back of your book, is not just a reference list of words

appearing in your book. Picking out keywords or chunks of text from your manuscript, then arranging them in another way — usually in alphabetical order, often complements your book. In other words, you can exploit indexing to make your book better!

Through this tutorial, you will find what is required for good indexes, how indexing helps you and your readers, and some techniques for making back-of-the-book indexes with L<sup>A</sup>T<sub>E</sub>X. On top of that, in non-alphabetical languages, you cannot simply use `makeindex` or `xindy`, mainly because these languages don't have any concept of alphabetical order. So, I will also go over practical cases of making back-of-the-book indexes in non-alphabetical languages.

### Yumi TAKATA

*Japanese typeface design — similarities and differences from Western typeface design*

What is Japanese typeface design about? As a Japanese type designer for nearly 30 years, I will explain what it is to design a Japanese typeface, and what it does and does not have in common with designing a Western typeface.

First, we will take a quick look at the history of Japanese characters, in particular how the shapes of the characters have evolved through time.

Then I will illustrate the process of Japanese typeface design in detail. Japanese typeface designers face the challenge of dealing with more than 9,000 characters and multiple constituent scripts. Some examples will be given of the various techniques we use to create readable and visually appealing typefaces, including adjusting for optical illusion.

Another big challenge we face is the vertical and horizontal writing modes. I will show how we fine-tune the glyph design of each character, one by one, for both vertical and horizontal writing modes.

Finally, the complications related to Japanese coded character sets will be briefly explained.

I hope my presentation gives you a grasp of the Japanese typeface design and leads to further discussion.

### Yusuke TERADA

*Development of TeXShop — the past and the future*

TeXShop is a widely-used open source T<sub>E</sub>X editor and previewer for Mac OS X. TeXShop is developed by Richard Koch, emeritus professor of mathematics at the University of Oregon, and many other worldwide contributors, including me. Now it is localized for as many as 10 languages. While it has already sufficient functions for editing T<sub>E</sub>X documents, TeXShop is still being updated. In this presentation, I will give an outline of the design concept of TeXShop and some new features that have been added recently, es-

pecially for editing Japanese documents. In addition, I will show a vision of TeXShop for the future.

### Didier VERNA

*TiCL: The prototype*

Last year, I presented some ideas about using one of the oldest programming language (Lisp), in order to modernize one of the oldest typesetting systems (T<sub>E</sub>X). That talk was mostly focused on justifying the technical fitness of Lisp for this task. This time, I would like to take the opposite view and demonstrate a prototype from the user's perspective. This will involve showing what a TiCL document could look like, the implications in terms of typesetting vs. programmatic features, and also in terms of extensibility (relating this to package authoring).

### Alan WETMORE

*Wind roses for T<sub>E</sub>X documents*

In recent years a great many systems for including plots and graphics in T<sub>E</sub>X documents have been developed. Many varieties of scientific plots are directly supported by these packages. One style of plot which has not been available is a wind rose: describing the probability of wind speed and direction with a stylized polar plot. This report will describe a set of macros for TikZ for preparing wind rose plots.

### Masafumi YABE

*Japanese text layout — basic issues*

This tutorial presents basic issues concerning page formats and typesetting methods applied to the main text of a Japanese book with reference to the typographic characteristics of the Japanese writing system. The issues to be discussed are threefold.

The first section focuses on the text direction, vertical or horizontal writing mode, which depends on an editorial decision and affects, in many ways, the page layout as well as the printed forms of a Japanese text.

The second section concerns typesetting methods applied to basic Japanese text as a sequence of characters without spaces between words, and illustrates relevant typographic building blocks in line with composition rules with an emphasis on the functional importance of punctuation marks and their surrounding spaces for line and paragraph adjustments.

The last section addresses several issues about methods for mixed composition of Japanese and Western texts, presenting major technical problems relating to differentiation and harmonization of typographically heterogeneous elements in sequential texts: Western text in the context of main horizontal or vertical Japanese text as well as Japanese text in the context of main Western text.