Indian TEX Users Group

Review of TUG98

Kaveh Bazargan

Focal Image Ltd., London
kaveh@focal.demon.co.uk

Philip Taylor

Univ. of London, Egham, Surrey
p.taylor@vax.rhbnc.ac.uk

The 19th annual TEX Users Group Meeting took place in Toruń, Poland, in August of this year. The following is the impression of the authors who tried to attend all the talks. Unfortunately a couple of the talks were not attended by either of the present authors. Apologies to those participants.

Daniel Taupin described his LATEX to RTF converter, `ltx2rtf`. His problem is that he wants to transfer LATEX documents to colleagues who may not have LATEX, and not even a PostScript printer. His conclusion is that the great majority have

MicroSoft Word installed, and therefore that should form the basis for the format in which the files are transferred. Since RTF can be read in directly by Word and by other word processors, and the specifications are published, it makes a logical choice for the conversion process. Of course the limitations of Word cannot be overcome. For example equations are saved and placed as bitmaps. Daniel's colourful personality and his vociferous comments at most presentations livened up the proceedings enormously.

RWD Nickalls is a consultant anaesthetist in a hospital. He uses LaTeX in conjunction with GNUPLOT to display and print charts in the operating theatre. The use of ASCII input for both GNUPLOT and LaTeX makes for a reliable system.

Marcin Woliński presented a LaTeX(2e) package for 'pretty-printing' program codes such as Pascal and Prolog directly through TeX. The results are quite amazing, with the code printing with the usual conventions of bold, italic, correct indentation, and cleverly using maths mode to position mathematical constructs. It is a good use of the unique nature of TeX, being a text processor and a programming language in one.

Metapost seems at last to have opened the door to the untapped power of Knuth's METAFONT. Metapost uses only a small subset of PostScript operators. In particular, there is no access to the PostScript pattern operators through Metapost. Piotr Bolek's solution is to allow the user to use routines in MetaPost to define patterns, but then to post-process the file using a PERL script to insert the correct code according to PostScript level 2. His `mpattern` package consists of the metapost code to define the patterns, and the PERL script for post-processing.

Taco Hoekwater, of Kluwer Academic, laid out the procedure to generate PostScript Type 1 fonts from METAFONT sources. The major incentive for this work was to allow the inclusion of these fonts in PDF files generated by PdfTeX (Adobe Acrobat is not good at viewing bitmap fonts). The translation is not straightforward. For one thing, METAFONT has a richer set of primitives for describing font shapes and characteristics than the definitions of Type 1 fonts. Secondly, METAFONT allows the use of the equivalent of PostScript stroked paths, whereas Adobe Type 1 fonts can only be described as outlines. A reliable translator would open the way to more

widespread use of METAFONT, which is undoubtedly the most powerful tool for generating typefaces. At the moment the procedure is quite long, as several manual steps are needed. Taco's paper, incidentally, is an excellent introduction to Metafont, and to Type 1 fonts.

Using his Calculator demo, Hans Hagen posed a philosophical question which emphasized the changing world of document production. '`calculator.pdf`' (available from http://www.ntg.nl/context) is a PDF file, viewable in Adobe Acrobat Reader, which looks and behaves like a scientific calculator. So is it a program or is it a document? Well, it is a kind of intelligent document, which goes one step further than having hypertext links. Its usefulness may be trivial, but the method used to produce it is extremely significant. The 'document' was produced by allowing TEX to embed Metapost and JavaScript code, and produce one composite PDF file. Metapost produces graphics parametrically on the fly, JavaScript does the calculations behind the buttons, and TEX does the overall typesetting and management. The whole process is processed by Hans's powerful `Context` package, which uses PdfTEX to produce the output directly (rather than taking the DVI–PostScript–Distiller route). The Calculator demo is perhaps the single most powerful example of how relevant TEX still is in the world of electronic publishing.

Another little gem from Hans Hagen is the idea of 'visual debugging'. TEX has a subtle, and sometimes unconventional way of using vertical and horizontal space. When writing macros, a programmer often has to get into the mind of TEX to work out why certain spaces have appeared in strange places, or why a paragraph is indented, etc. This can take quite a while to work out, and the answer is usually obvious in hindsight! It turns out that by carefully redefining a lot of TEX's placement commands, such as `\hfil`, `\vfil`, `\kern`, `penalty` etc, TEX can be persuaded to show graphically all spaces, fills, glue, and boxes. Hans has designed different solid and dashed lines to signify each type of box or glue placed by TEX, thus providing a visual picture of what he calls 'an endoscopic view of TEX's stomach'.

Bogusław Jackowski and his colleagues in BOP s.c. are masters of PostScript, especially when applied to TEX. In the first of his most entertaining and grip-

ping deliveries, Bogusław described his CEP and COP packages for compressing PostScript files. Most people who deal with PostScript files are aware of the enormous file sizes. What is not well known is that it is possible to reduce these files substantially by using PostScript's own compression algorithms. Most applications do not bother to use this compression when generating PS files. The most efficient compression is achieved when the type of data in the PS file is known *a priori*. Depending on the type of data, one of the compression programs (CEP and COP) is chosen.

In his second talk, Bogusław described a suite of unusual PostScript tools for use with TEX. `Tiff2ps` is a program that converts Tiff files to PostScript files, the program itself having been written in PostScript. `Pf2afm` tries to extract AFM (Adobe Font Metrics) files from PFM (Printer Font Metrics) files. AFM files are needed for TEX users, normally converted to TFM. PFM files are subsets of AFM files, and are used in Windows. Another utility, `Ttf2pf` converts TrueType fonts to AFM and Type 42 (PostScript's equivalent of TrueType). Finally, `Colormap` is a TEX macro package that makes simple colour and grey scale changes to EPS files that are called by the TEX file, without modifying the EPS files themselves. The two papers have a lot of useful material regarding PostScript and fonts, and are recommended reading. One thing that comes through is the power of the freeware GhostScript program which is used in all this work.

Hàn Thế Thành, author of PdfTEX was not present, but his paper was read by Jiři Zlatuška. His idea is to improve the look of text set on a narrow measure by horizontally scaling the text in Acrobat after TEX has finished its work, thus reducing the large variation in word spacing. This scaling has to be limited to only a small percentage, as otherwise the different shapes of the letters become noticable. Also, the differing stem widths make the letters appear lighter or bolder, a generally undesirable effect.

Miroslava Misáková takes this idea of postprocessing one ingenious step further. (In fact she got the idea by looking at Gutenberg's 42-line Bible!) She solves the problem of the differing stem width by regenerating the font using Metafont, and varying the width of the font but keeping the stem width constant—something that

probably can only be done with Metafont. This makes for a more uniform looking block of text. The procedure is first to set a paragraph by allowing the right margin to vary by some 5%. The right margin is therefore ragged, but the word spacing is more uniform. Using TEX macros and `\specials`, marks are embedded in the DVI file, signifying the positions of the ends of the lines. A Perl script then interrogates the DVI file, and for each line, works out the correct font to use for setting. Usually, it would look at a set of 10 fonts already generated, and choose the closest match. The results presented are most impressive. Even when adjoining lines vary in letter width by some 7%, the text looks uniform, and is comfortable to read. Of course the ideal solution would be to add this capability of variable font width to TEX as one of its parameters, but that becomes a more complicated procedure.

Richard Kinch's contribution was a new math font, Belleek, that he put into the public domain. The font was produced manually, using Fontographer, and designed to work with Times as the body text. His conclusion from the experience of creating this font is that Metafont is the best tool for font creation, but that the main stumbling block for its more popular use is that it creates bitmap output, as opposed to outlines. He discussed some methods for getting outlines from Metafont.

The presentation that created the most heated discussion was kept for the last day. This was the joint presentation by the Dutch contingent (NTG), who had arrived in an orange coloured van at the conference. The Dutch team, led by Hans Hagen, presented some radical ideas. These ideas were a culmination of lively discussions among Dutch TEX users during the months prior to the conference, and led to the 'NTG TEX future working group'. They felt that they had to share these ideas with the TEX community. Here are some of the highlights of their views:

- It is proposed that the three projects currently under way to extend the capabilities of TEX, namely Omega, Pdftex and e-TEX, should merge into one project. NTS (the New Typesetting System) is a project to rewrite TEX in Java, with a guaranteed backward compatibility with TEX82, at least for the first 5 years. The Dutch team feel that this is a mistake, and that backward compatibility should be discarded.

- The group feels strongly that current distribution and implemention methods for TEX are too complicated, incompatible, and involve too many files. A typical complete TEX system can comprise some 30,000 files, which the Dutch group feel should be reduced by a factor of 100. This would be achieved by setting up an intelligent central database which could be queried, and which would install only the necessary files according to a user's specifications. This database could have access to CTAN to obtain any files not available locally. It is also proposed that CTAN be more closely regulated with files and packages being put up with more information accompanying them.
- Another suggestion is to extend the capabilities of TEX to allow the production of electronic documentation. These include more support for colour and hypertext linking within TEX, rather than relying on `\special`s to produce these capabilities at the back end. To allow the direct inclusion of Metapost graphics, a set of `\special`s is proposed which allows the use a larger subset of PostScript commands than is used in Metapost.
- Fonts have always caused problems for all TEX users apart from real experts. The NTG TEX working group recommend an overhaul of the font system, including replacing the way TFM and VF formats are defined, separation of ligatures and kerning information from the character metrics of fonts, and the facility for direct output of outline fonts from Metafont. (The latter echoes the wishes of Richard Kinch.) The group also call for a friendly user interface for mathematics, and a general cleaning up of the TEX syntax.
- The final proposal is a standard for the specification of `\special`s accross all systems. The proposal is based on that of Nelson Beebe, and the aim is to make TEX truly portable.

Laurence Finston presented what should have been an extremely interesting paper on the generation of concordances using a combination of TeX & Lisp. This paper was rather difficult to understand by reading it, so we looked forward to Laurence's oral presentation. Unfortunately, he chose to **read** his paper rather than presenting it *extempore*, as a result of which his audience (including your hum-

ble scribes) started to sneak out after the first five minutes. We learn that M. Taupin asked at the end if Laurence considered TUG'98 merely a practice run for his dissertation. . . !

Janusz Nowacki, speaking through a translator, presented his work on Antykwa Toruńska, a modern digital adapatation of a traditional Polish font. Janusz commenced his talk by recounting the motivation for his work: the simple fact that since the DTP/EP revolution, no Polish text had been produced using a Polish font! Despite the problems of translation, Janusz kept his audience enthralled, and his talk was one of the most interesting and stimulating delivered. Antykwa Toruńska in its current version originates as scanned bitmaps; it is then post-processed in Corel Draw and fine-tuned in Fontographer. It is available in regular, italic and bold versions.

Karel Skoupý presented a report on the current status of NTS ("a New Typesetting System"). This project, which was first mooted in 1992 and which has only recently sprung into life, involves a complete re-implementation of TEX using Sun's JAVA programming language. Rather than simply being a re-implementation, however, the project's real aim is to completely restructure TEX/NTS internally so as to make the inter-module interfaces far clearer and to provide a framework within which current (for example e-TEX, Ω and PdfTEX) and future developments can be easily and straighforwardly embedded. Karel emphasized that the choice of JAVA had many benefits, of which guaranteed portability and network awareness were the greatest, but also had its drawbacks (for example, the lack of rigorous compile-time type-checking for derived types).

Although the majority of papers took place on schedule, the unanticipated absence of John Plaice (originator of the Ω project) caused some gaps in the schedule. These were filled by extempore talks by Kaveh Bazargan ("Can TEX create Farsi scripts?"), Daniel Taupin ("`LaTeX2HTML` for the i'x86 familiy of operating systems"), and Taco Hoekwater ("`DVIview`: a reference previewer").

The four days surrounding the conference proper were occupied by tutorials. Hans Hagen had bravely volunteered to give a split two-day tutorial entitled "Actually making an electronic document"; Philip Taylor & Jiři Zlatuška offered a

preview of a tutorial originally prepared for IFIP'98 ("Document design, document markup, and the converging worlds of computer typesetting and electronic publishing"); and Bogusław Jackowski offered a one-day tutorial on "TEX and PostScript integration). All three were well attended, but the Taylor/Zlatuska tutorial rather fell apart when the participants agreed that it was either too advanced or too basic but certainly not the right level... Hans' and Bogusław's tutorials, on the other hand, both proved very successful.

Of course, a conference is considerably more than a series of tutorials and talks: the social aspects are in many ways as important if not more so. From this perspective TUG'98 was a great success: the organizers had carefully planned the social activities, and the gala concert and banquet at Artus' Hall ("King Arthur's Hall") was one of the high spots. Equally memorable was the outing to the mediæval castle at Golub-Dobrzyn, where the guest were greeted by the sight of an ancient and wizened serf, clad in chain mail, slowly roasting a pig on the spit. There was limitless beer to accompany this feast, and a guided tour of the castle; guests of a weak disposition were probably somewhat discomfitted to learn of the mildly unpleasant ways in which "guests" of an earlier age were accommodated: simple pleasures such as being tied down under a slow but continuous water drip, or having salt rubbed into fresh wounds, were among the many ways in which their hosts passed away the weary hours... Despite (or perhaps because of) this, the sight of the pig being slowly roasted conjured up in several minds an analogous scene in which the central participant was rather more anthropoid than porcine... Simpler pleasures were also to be found every evening, when beer, vodka, guitars, whistle and voice mingled together to create an atmosphere of great conviviality and warmth. All in all we were impressed by the genuine hospitality that we received by from the local organizing party.