Typesetting at the CWI – Part 1

by Jaap Akkerhuis

Introduction

In this article I will discuss how the typesetting of various publications, like the Newsletter you are reading, is done at the CWI.

Since the founding of the Institute in 1946, the Centre has published reports. The typing and printing of its publications has always been done "in house". This made it possible to get a quick turnaround time between the writing and printing of a report. In this way it is also possible to have a close interaction between the author and the typist, which allows the author to have a good control not only over the contents but also over the appearance of his publication.

It was recognised at an early stage that a computer could provide some help with the composition of a publication. This lead to a program, called TEKSTSCHAAF [1], on the Electrologica X8 computer written by D. Grune. This system allowed you to format a text and it would automatically provide page headings, page numbering, *justification**, etc. There was a nice mechanism to prevent *widows*†, but the most interesting aspect of the program was the way the justification was done. Most programs do justification on a line by line basis, but in this system a couple of lines (normally 3) were considered before the final adjustment took place. Where the extra spaces were added in the line was dependent on an "elastic band" between the words. The stretchability of this band depended on certain conditions that could be changed by the user, and some analysis of the (Dutch) text. A similar approach, using stretchable glue and taking into consideration a total paragraph before justification, is used in the TEX system [2] designed by Knuth.

A successful system that has been used for a long time was a formatting program made by H. Noot [3]. It had good error reporting and a mechanism for avoiding widows. All these above-mentioned programs did not give support for mathematics, and could only handle non-proportionally spaced characters (i.e. typewriter fonts).

The current system in use consists of the UNIX‡ [4] formatting tools. These were first used by the department of computer science. When the CWI obtained a phototypesetter, the rest of the institute slowly started to use these

* Justification is the the process of adding extra space between words to improve the appearance of a text, for instance to get right adjusted margins.

† A widow is a single line of text or just one word on the top of a page before a new paragraph starts. It actually belongs to the paragraph before.

‡ UNIX is a trademark of Bell Laboratories.

tools too. Although they are simple and straightforward, they turn out to be very flexible and capable of producing a lot of different things, from simple text up to line drawings, all of high quality.

Computers and typesetting

In the last decade a revolution has taken place in the typographical world. Nearly everywhere the hot-metal typesetters have been taken over by modern equipment like laser- and ink-jet printers, phototypesetters, daisy wheel and matrix printers. This revolution did not only take place at the output side; the process of keying in text has also been strongly influenced by computer technology.

This has lead to two different approaches: interactive typesetting and batch typesetting.

Interactive typesetting starts at the low end with a typewriter with some editing facilities and leads via *word processors* and their derivates to systems which allow the mark-up of complete and complicated material, including digitised pictures. These can be characterised as simple to very sophisticated picture editors, where the picture is built from simple characters to very complex entities. Some of these systems are even able to automatically make the necessary corrections for reproducing colour pictures on a printing press [5]. Although these systems are very flexible and can produce high quality material, they are very expensive and cost a lot in computer power. A cheaper approach is batch oriented typesetting, also known as formatting. This is much more popular in the scientific world, because there is no need for huge productivity, and it is yet another application which will run on the probably already available research computer. The quality of the output can nevertheless be high.

At this moment there are three different popular and generally available formatting programs. A well-known program is the SCRIBE formatter [6]. One of the interesting aspects of SCRIBE is that the input does not specify the form of the output. Instead it specifies the different segments of the text, like titles, headers, paragraphs with or without *hanging tags**, etc. The only way the output is controlled is by specifying the style of the document. The actual output produced is dependent on the output device and on the way the different styles are implemented. Support for the typesetting of mathematics is quite poor.

The TEX formatter, designed by D. Knuth, gives the user close control over the final appearance of the output. It is known for its good output quality, due to the fact that it formats a complete paragraph at once. The mathematics support is good, since the program has a lot of built in information about the

* A hanging tag is the part of the text hanging like a tag on the left side of a somewhat indented paragraph. This line of text in this footnote is a bit superfluous, but it will make the footnote big enough to demonstrate the use of the hanging tag.

shape of the character. A disadvantage is its enormous size and the crude input syntax. The system is especially written for raster printer devices, which makes interfacing to real typesetters a problem [7].

The standard UNIX formatter, TROFF [7], is specially designed for a typesetter and has extensive macro capabilities. In accordance with the general UNIX-philosophy, it is a central formatting tool. Special tasks like mathematics typesetting are performed by *preprocessors*. This system is the one used at the CWI.

The three mentioned formatters have had a great impact on "in house" typesetting in the (scientific) world and their influence is noticeable everywhere, e.g. in the proposed international standard for text processing [9].

History of TROFF.

Around 1964 a formatting program called RUNOFF made its appearance at MIT on the Compatible Time Sharing System. This program had a lot of influence and implementations were made for many different systems. One of them was the program ROFF at Bell Labs. Around 1973 ROFF was completely revised to NROFF (new roff). The output device for NROFF was basically the Teletype 37. It also got a cousin called TROFF (typesetter ROFF) which used a Graphics System C/A/T-4 typesetter as its output device. These three programs were written in assembly language, but in 1975 TROFF and NROFF were recoded in the higher level language C, and at the same time its capabilities were expanded. In 1979 Bell Labs acquired a new phototypesetter. This would have been a good opportunity to replace TROFF by something better, but no one could come up with a better design. Brian Kernighan started to modify TROFF, so it would run "henceforth" without any change on a variety of typesetters. This renewed version of TROFF is known as typesetter independent TROFF, and is usually called DITROFF [10] .

In 1981 the CWI obtained its own typesetter, a Harris 7450. This machine is connected by a serial line to a port selector so it can be shared between various computers. In order to get the machine running as soon as possible, I developed a filter which took the TROFF binary output code specific to the C/A/T and translated it into yet another binary code for the Harris. It actually consists of two filters, one for the interpretation of the C/A/T code and another one which maintains the protocol with the Harris, which is rather too arcane to describe here. Having two filters makes it possible to have other programs like TEX and a plotting package produce Harris code without their having to know about the ghastly protocol. When I was investigating the changes that had to be made to TROFF, I learned about Kernighan's work and decided to wait until it was available, not wanting to reinvent the wheel. When DITROFF finally arrived, I made the necessary changes to let it drive the Harris typesetter [11]. Also, Dutch hyphenation rules were added, which actually amounted to incorporating the BESTESPLITS [12] program inside TROFF. This addition makes it possible to switch from the American to the Dutch hyphenation algorithm in a single text. German hyphenation rules will be

added soon*.

Using the system

As explained before, TROFF is actually the central program among the typesetting tools. It is hardly used on its own but most of the time it is used in combination with a macro package and/or with one or more preprocessors. Note, that the preprocessors do not know anything about the output device, but they merely generate TROFF *requests*† and macro/string definitions, which are processed by TROFF, which does all the necessary calculations. By means of examples I will show you the capabilities of a macro package and some of the preprocessors. The input of the examples will be set in a line printer style font, OCR-B, and the corresponding output will be generated in the laurel font, which differs from the Times font used for the running text. The macro package explained is a standard one and is known as the *—ms* macro package [13].

The —ms macros

These macros give TROFF SCRIBE style capabilities. By classifying parts of the text, the output is produced according to a certain standard. It is not uncommon to do this: the LaTeX system [14] is especially designed to give TeX these capabilities as well. In this system some of the features of the TBL TROFF preprocessor are also implemented.

So one could start a paper with:

```
.TL
The Title
.AU
The Author
The Ghostwriter
.AI
The Author's institution
The Ghostwriter's address
.AB
The abstract of the paper begins, it will be printed as a
centered
block using 5/6 of the current line length.
Note also the rearrangement and justification of
these
input lines in the output.
.AE (abstract ends)
```

and the output will be:

---

* This is a widow

† A request is a line of text, in the text to be typeset, requesting TROFF to perform a certain function, for instance to generate a new page or to print the current page number.

The Title

The Author
The Ghostwriter

The Author's institution
The Ghostwriter's address

ABSTRACT

The abstract of the paper begins, it will be printed as a centered block using 5/6 of the current line length. Note also the rearrangement and justification of these input lines in the output.

As you can see, a line starting with a period means something special to TROFF.

Normally text is divided into paragraphs. To start an indented paragraph, a `.PP` command is used. For a left aligned paragraph, the `.LP` command is used.

Paragraph headings

Headings like the previous one are generated with

```
.SH
Paragraph headings
.LP
Headings like the ...
```

It is also possible to have the headers numbered automatically. The following example shows this.

```
.NH
Basic CPUs, Processor options, Memories
.NH 2
Central Processors
.NH 3
PDP-8 Kits
.NH 3
PDP-11 Kits
.NH 2
Memories
.NH
Mass Storage
.NH 2
Cartridge Disk Kits
.NH 0
Appendix
```

1. Basic CPUs, Processor options, Memories

1.1. Central Processors

1.1.1. PDP-8 Kits

1.1.2. PDP-11 Kits

1.2. Memories

2. Mass Storage

2.1. Cartridge Disk Kits

1. Appendix

The numeric argument to the `.NH` macro call specifies which part of the generated number needs to be incremented. The `.NH 0` command will reset the top level to one.

Indented paragraphs
It is possible to have hanging paragraphs with or without hanging tags. The following example illustrates this.

```
.IP [1]
Text for first paragraph, typed normally as long as you
wish. The first argument is the so called hanging tag.
You may omit it as in:
.IP
and the paragraph will be just indented. A second
argument changes the amount of indentation like:
.IP first 12
and the indentation will be 12 en's.
This value will be the new default value so the next .IP
.IP next
will give you this.
If you don't want a tag but do want to change the
indentation use
.IP "" 5
As done here, the "" denotes an empty argument.
```

[1]Text for first paragraph, typed normally as long as you wish. The first argument is the so called hanging tag. You may omit it as in:
and the paragraph will be just indented. A second argument changes the amount of indentation like:

first and the indentation will be 12 en's. This value will be the new default value so the next .IP

next will give you this. If you don't want a tag but do want to change the indentation use

As done here, the "" denotes an empty argument.

Footnote generation

Footnotes* are placed at the bottom of the page. They are generated with

```
Footnotes*
.FS
* Like this one
.FE
are placed ...
```

The character size† will be automatically reduced by two points.

* Like this one

† In the American typographic world the size of a character is usually given in Pica points, while in Europe Didot points are used. There are 12 points in a Pica and 6 Picas in an inch. A Pica point breaks down to 0.35146mm and a Didot point to 0.376065mm, so a Pica point is 0.934572 Didot point. Of course, TROFF uses Pica points.

Keeping blocks together
Sometimes the output should be kept together. This is achieved with the keep commands.

```
.KS
Text inserted here will be kept together until the
.KE
```

If the output of the material placed between the `.KS` and the `.KE` won't fit on the current page, a new page is generated before the actual output takes place. Using a `.KF` instead of a `.KS`, the new page will not be issued, but the output of the material placed between the `.KF` and the `.KE` will be delayed until there is enough space. This way material will *float* trough the actual output.

Displays
To get an exact replica of the input, so TROFF will not rearrange (apart from the typeface) the output with respect to the (current output) line length, the material must be surrounded by the *display* macros.

```
.DS
This is how the input to the examples is made
.DE
```

To get this display as a block of text, the keep macros are used, to prevent a split of the material over a page.

Note the caveat in displays. If the output device has proportionally spaced characters the appearance of the output will be different from the input. Consider the next input:

```
.DS
We want this X to line
up with this Y.
.DE
```

And this is how it looks:

We want this X to line
up with this Y.

The above example shows a basic problem for the ignorant user. There is a difference between what has been written down and the way it looks —how it is read is yet another problem.

Controlling the typeface
As you have seen the typeface changes when there is a different function of the text. Titles will be set in the **bold**-typeface somewhat bigger, while headers will just become bold and footnotes just smaller.

Apart from these automatic changes of the typeface, it can of course be specified by the user.

```
.I
After this command everything will be italic until the
.R
```

After this command everything will be italic ...

If the `.I` has an argument, only that argument will be in italic, and an optional second argument will be put right after it. So this is the way

```
.B bold -item.
```

to produce **bold**-item.

To get (two points) smaller output:

```
.SM
This will be smaller,
.NL
and back to normal.
```

This will be smaller, and back to normal.

The `.LG` command will make the size Larger.

There are of course a lot more things you want, and can do with the *-ms* macros, for instance creating running page headers, etc. However I will not treat this here, nor will I give any details of the bare TROFF commands since I merely want to give an overview of the system and do not want to write a user manual.

In the next issue of the CWI-Newsletter I will discuss the typesetting of tables, mathematical formulas and graphics.

Finally, I am grateful to Sape Mullender for allowing me to use his programs which made it possible to create the "artwork" on the cover.

References

[1] D. Grune, *Handleiding bij de TEKSTSCHAAF*, LR 2.5, Mathematisch Centrum, Amsterdam, 1972.

[2] Donald E. Knuth, *TEX and METAFONT,* American Mathematical Society and Digital Press, Bedford, MA, 1979.

[3] Han Noot, "Structured Text Formatting", *Software Practice and Experience*, **13**(1983), 79-94.

[4] K. Thompson and D. M. Ritchie, *UNIX Programmer's Manual*, Sixth Edition, Bell Laboratories, Murray Hill, NJ, 1975.

[5] A. F. de Winter, "De grafische industrie — Elektronica tussen pers en post", *Natuur en Techniek*, **51**(1983), 432-447.

[6] Brian K. Reid, *Scribe: A document Specification Language and its Compiler*, Carnegie Mellon University, Pittsburgh, PA, 1980.

[7] A. H. Noot, "DVI-Code to the Harris 7500", *TUG-Boat*, March 1984.

[8] J. F. Ossanna, *NROFF/TROFF User's Manual*, Computing Science Technical Report 54, Bell Laboratories, Murray Hill, NJ, 1976.

[9] ISO standard, *Information Processing Systems - Programming Languages - Text Interchange and Processing* (Sixth Working Draft), ANSI, New York, 1 June 1983.

[10] Brian W. Kernighan, *A Typesetter-independent TROFF*, Computing Science Technical Report 97, Murray Hill, NJ, revised, March 1982.

[11] J. N. Akkerhuis, "Typesetting and Troff", *Proceedings EUUG conference 7-9th September*, 29-41, Dublin, Trinity College, 1983, Also available as IW 247/83, CWI, Amsterdam 1983.

[12] J. C. van Vliet, *Bestesplits 1*, NR 28/72, Mathematisch Centrum, Amsterdam, 1972.

[13] M. E. Lesk, *Typing Documents on UNIX and GCOS: The −ms Macros for troff*, Bell Laboratories, Murray Hill, NJ, 1977.

[14] Leslie Lamport, *The LaTeX Document Preparation System*, Second Preliminary Edition, SRI international, Menlo Park, CA, December 13, 1983.