My DIC Rec

SCIENCE AND ENGINEERING RESEARCH COUNCIL
RUTHERFORD APPLETON LABORATORY

COMPUTING DIVISION

DISTRIBUTED INTERACTIVE COMPUTING NOTE 913

TYPESETTER INDEPENDENT TROFF IMPLEMENTATION
Interfacing T.I.Troff to FR and DISPLAY

Issued by
Alan J Kinroy

24 October 1983

DISTRIBUTION:
E V C Fielding
D R Gibson
R W Witty
D A Duce
L O Ford
P J Smith
S T Frost
H K F Yeung

1. Introduction - T.I.Troff

T.I.Troff is an extension of V7 Troff.The main differences are :-

1) The output is ASCII and not targetted to a specific typesetter.Given a description file for a particular typesetter and font description files output can be generated for that typesetter.

2) There are extra drawing capabilities for drawing lines, circles, ellipses, arcs and splines.Character height and slant may also be adjusted. Fonts can be mounted at any time and the number of fonts mounted concurrently is only limited by the capabilities of the target typesetter.

2. Existing Software

FR which currently runs on the PDP11/70 accepts output from V7 Troff and generates a tape with instructions to drive the FR80. Characters are currently drawn by software using the Hershey fonts.It is also possible to display FR output on a Tektronix.

DISPLAY which runs on PNX on the PERQ performs a similar function to FR on the Tektronix. There is considerably less software involved in the implementation, as the code for accounting, camera selection and tape formatting which is required for the FR80 is not required.The distinction between maintaining the troff state machine and performing graphics operations has been more clearly defined in the DISPLAY implementation.

3. Objective

The objective is to adapt FR and DISPLAY so they will accept output from either V7 Troff or T.I. Troff and perform the required interpretation and drawing actions.Extra graphical support will be needed for the new features of T.I. Troff. For this purpose I propose to adapt some routines which draw ellipses and circles using dots.

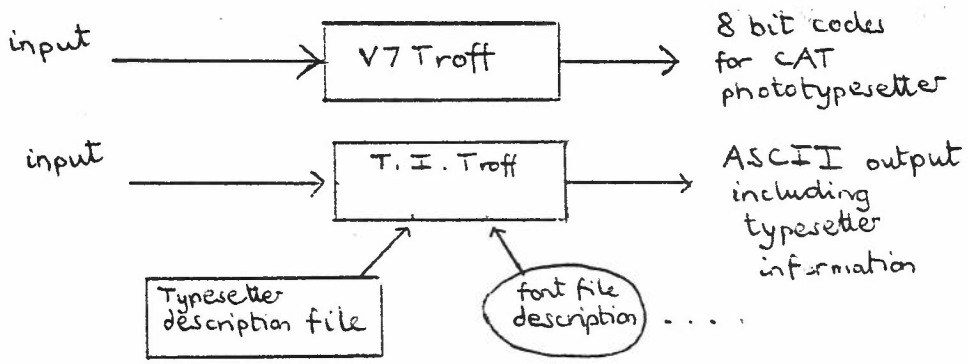
4. Strategy

It would be useful if both FR and DISPLAY used the same interpretation and graphics modules with the only difference being the extra accounting, camera selection and tape formatting performed by FR.

However because of the large software investment in FR and the fact that the extra code is interlaced with the interpretation and graphics code it would be too difficult and time consuming to build the extra FR code on top of DISPLAY.

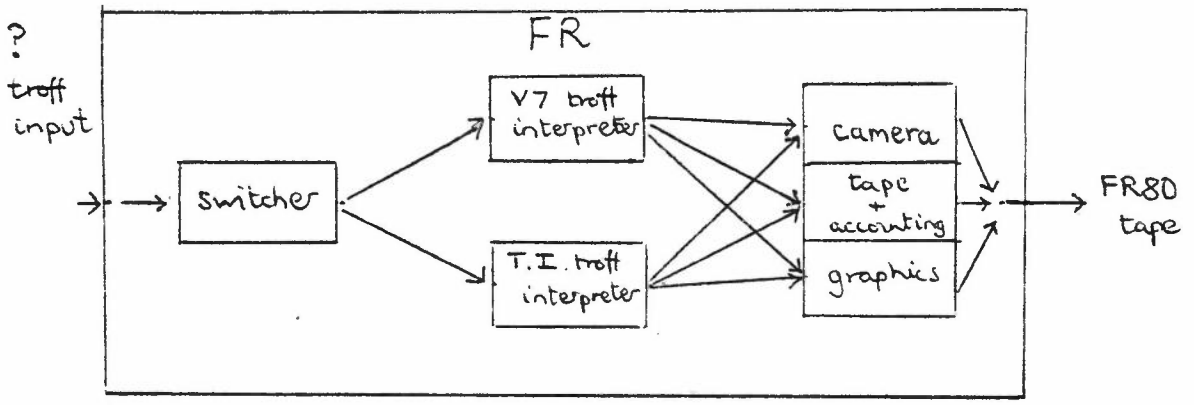
Thus the strategy must be to modify the existing FR by adding a switcher module which will look at the start of its input and then select either the existing V7 Troff interpretation routines or some newly implemented T.I Troff interpretation routines.Both interpretation routines should call the same graphical output routines.

5. Implementation details



The font description files used by T.I. Troff contain the following information for each character:- Character name,width in mc units, ascender/descender information, device code required to print it. These have to be modified to contain the Hershey widths for use with FR and display.

Schematic Illustration of modified FR



5.1 Switcher

Because T.I.Troff always starts with the sequence "x T ..." whereas V7Troff starts with 0100 0111 (octal codes) a switcher module can examine the first few characters of the input stream and decide which troff output we are reading. Note there is no way of determining if input to the switcher is genuine as the first few characters of input can easily be faked.

5.2 Device Resolution

Once the switcher has decided which troff output we have got we can either set the resolution to that of the CAT for V7 Troff or wait for further device information in the case of T.I. Troff.

In the existing FR and DISPLAY implementations the device resolution is wired in as a constant into the code.

5.3 Fonts

To maintain compatability with V7 Troff the fonts would have to be ordered in the manner expected by V7 Troff.If sharing of fonts is not essential T.I. Troff could use separate fonts which could be ordered as convenient.My character printing routine for T.I. Troff would have to map the character name into an offset in the required font file.This might require a table lookup mechanism.

V7 Troff could only have 4 fonts mounted at any one time.Requests to change fonts were only effective at page breaks.These restrictions have been removed in T.I. Troff.Currently FR only holds one font file in memory and reads others as required.DISPLAY holds all 4 font files in memory but this causes an initial delay in start-up time.

On a PDP11 holding more than 2 fonts in core is virtually impossible because there is no virtual memory.On virtual memory machines there is a tradeoff between holding font files in memory (and thus having a large program paging and initial start-up delay) against the delay in reading files as required.

So that FR is the same on the PDP11 and the VAX it is probably best to leave it reading fonts as required.A more general way of knowing which fonts are 'notionally' mounted on each position will be required for both FR and DISPLAY

6. Implementation Strategy

The switcher code to examine the first few characters of the input is trivial.

I propose to adapt display to accept T.I. Troff output first, as I am more familiar with it and it will be easier (and quicker) to implement than working directly on FR.

I think some of the code from dcat.c can be used for interpreting T.I. Troff output and the routines for drawing conic arcs and splines in draw.c can probably be adapted.I will probably have to write a routine for doing character slant.

At the moment I don't intend to get too involved with PIC and IDEAL as they are pre-processors for T.I. Troff and my main objective is to interpret T.I. Troff output and implement the extra graphics functions which T.I. Troff provides.

7. Diagrams of Current & Intended Software

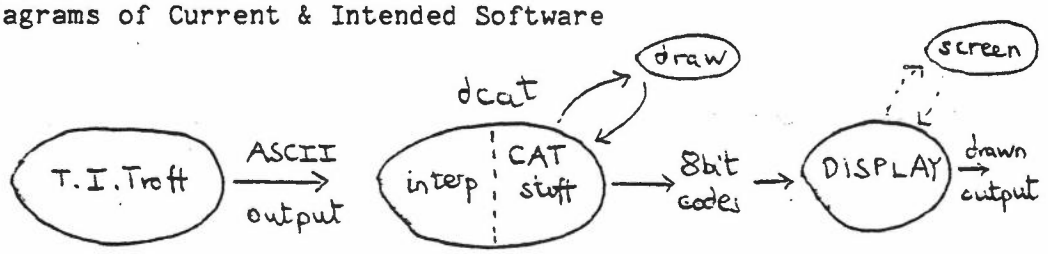


Fig. 1 T.I. Troff , dcat and display

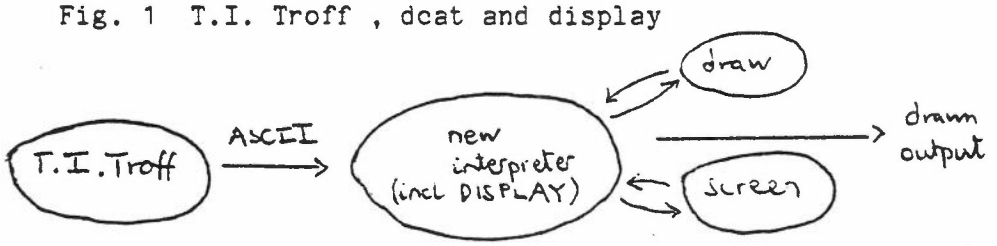


Fig. 2 Aim - T.I. Troff and interpreter calling drawing and character printing routines.

8. Table of T.I.Troff output and actions required by display.

Command	Meaning	Action required for display
x T devname	output typesetter is devname	set typesetter variable
x res n h v	Resolution per inch, min horiz & vert movements.	set res, hres, vres This info. is also available in the typesetter desc. file.
x init	Initialise parameters from device file.	set pointers to widthtab, codetab and fitab for default fonts.Make entry for font 0.Record mounted font names and positions in a table.
x font n fname	Physical position n holds font n. This is output for the default fonts or for .fp commands.	set ptrs to widthtab, codetab and fitab. These are the width and code arrays together with an array which contains values which index the other arrays. Record mounted fonts in a table.
f n	select font as current font for 1 < n < max no. of possible fonts. For n=0 use a hidden font - one that doesn't exist in hardware.	Read the Hershey defns for the font into memory.For the position 0 case mount widthtab, codetab and fitab.
s n	set point size to available size nearest n. Point size set is always rounded up version of n even if size n exists.	Find the nearest size using rtn t_size.Set pt_size (global variable).
cx	ASCII char x	find the character on a font. print the char. (may require a temporary change of font). set lastw (width).
Cxy	special char \(xy	output in same way as ordinary char. but add offset 128 into fitab table.

Hn	go to absolute horiz position.The min horiz dist. that can be moved may need to be taken into account.	set hpos - troffX in display may need to work in multiples of min distance that can be moved.
Vn	go to absolute vertical position.min vert movement may need to be taken into account.	set vpos - troffY in display. round current vpos to multiple of vres and round movement to multiple of vres.
hn vn	relative horiz movement relative vert movement	set troffX set troffY
[0-9][0-9]c	combination of hmot and character print	same actions as hmot and character printing.
nb a pn	newline new page n begins.	set hpos to zero call pagebreak().set vpos to zero
t string	text string.	use t_text to call the character printing routine and hmot.
D ...\n	drawing functions	Interpret cmd letters and call draw fns in draw.c.Need to mimic rtns hmot and vmot.For drawline can use wline system call.
x H n	set character height to n	implies adjusting height but not width. :-a scaling transformation on character height.
x S n	set slant to n degrees from vertical. It is not clear whether this is a rotation or skew.	interpret & perform the required transformation.
x s x p x t	stop pause generate trailer on last page	abort without printing any message ignore ? ignore ?

9. Areas of difficulty

Display which was originally deigned as a simulator for the CAT phototypesetter has some statically defined tables which use the fact that only four fonts are available.There

are also some instances of use of this fact within the code and some #defined constants which are CAT specific.

The occurrences of these CAT dependent parameters need to be identified and parameterised to deal with an arbitrary number of fonts. The statically defined tables need to be dynamically defined, possibly by pointer indirection.

From these observations it is clear that the source code for display will have to be changed.I had not originally anticipated this and hoped it would not be necessary.

There are still some features of the dcat driver which are unexplained:-

1) The requested point size is always rounded up to the next biggest size but the width of the preceding smaller point size is used for recording how wide the last character was.

) There are { } brackets in the T.I. Troff output which signify nesting levels.It is not clear where these are generated - it doesn't appear to be within T.I. Troff itself, so is presumably from one of the preprocessors. The origin of the 't text-string' output is also unknown.

Similar CAT dependent problems are likely to be encountered with FR. They are unlikely to be identical but experience gained with display should enable identification and correction to be carried out more quickly.

10. Time estimates

As always estimating how long the work is going to take is very difficult as it doesn't become totally clear what is involved until you start working on it !

Hopefully I should get the interpretation of basic T.I. roff output (possibly excluding drawing and slant) going on display within the next month.