MIS – 56

Management Information System Task Force Note October 20, 1986

Author: A.Berglund

Subject: **Corrections and Comments on MIS – 12**

Part of MIS – 12 is a memo from J.Jowett/LEP entitled "A Recommendation for Text Processing at CERN". This contains, however, a number of misunderstandings about SGML as well as a few other factual errors. This note aims to correct the major misunderstandings and errors.

CORRECTIONS TO DESCRIPTION OF SGML

Section 8.2 states: "It is claimed that SGML could become a standard text-processing system but it does not begin to offer the extensive mathematical typesetting capabilities of TeX.". This is, however, totally wrong! SGML was especially aimed at being able to handle complex technical text. A mathematics formalism (an SGML application) is being developed in the ISO working group working on Text and Office Systems. Some samples are shown in MIS – 55.

COMMENTS ON THE TEX FEATURES OF CHAPTER 2

- Page 5 point 7 (Automatically generated table of contents) states "The TeX program itself has to run through the input file twice to collect all the information concerning page numbers and chapter headings but the same is true of SCRIPT." With TeX it is **up to the user** to invoke TeX twice, whereas with SCRIPT you specify to the program as a parameter the number of passes it is to make – clearly much easier for the user. The SGML macros generate the table of contents at the front of the document if two pass mode is selected and at the end if single pass mode is used. In the latter case they are numbered as if they were at the front and you can easily move the pages by hand.

- Page 5 point 8 (Index) states "It is easy to make an index for a long document". In SCRIPT (be it using basic SCRIPT, CERNPAPER, or the CERN SGML Application) you only need to insert a command for each index entry and a command to indicate where you want the index to be printed. With TeX the procedure is very much more laborious; you have to write out the entries onto a separate file, then sort the entries and combine the page numbers and finally add the entries as normal text. The "LaTeX User's Guide and Reference Manual" writes:

  > Compiling an index or glossary is not easy, but LaTeX can help by writing the necessary information onto a special file. ... As you write your document, you should type an \index command for every page reference you want in the index. When the document is complete except for the index, add the \makeindex command and run LaTeX on the entire document to produce the idx file. You must then process the information in the idx file yourself to create a theindex environment that will generate the index; the **Local Guide** tells if there are any programs available on your computer to help.

COMMENTS ON THE COMPARISON OF TEX WITH SCRIPT

- On page 10 is stated "I have seen examples of the input to the prototype mathematics processor for SCRIPT and it is clear that it lacks the fundamental advantage of TeX's math mode, namely, that the way you input the formulae reflects their logical structure and the way you think about them."

  In MIS − 55 a comparison is made of various schemes to describe mathematics and it is clear that the "SCRIPT mathematics processor", based on the EQN syntax is even more like the way you think about mathematics than TeX.

- On page 10, a, is stated that SCRIPT takes much more CPU time than TeX. This is contrary to my own experience....

- On page 10, b, is stated "When SCRIPT encounters an error it stops dead and tells you that no browsable file was produced; sometimes you get an inscrutable error code. Sometimes it is extremely difficult to work out what went wrong."

  Part of SGML is a quite rigorous definition of the structure of the document. It is thus possible to give meaningful error messages for tags placed in the wrong place or left out. With CERNPAPER you do get a rather obscure message if you forget the end of a list whereas with the CERN SGML implementation you get a clear message to the effect of not having a closing tag for the open tag at line xx.

- On page 11, e, is stated "TeX will format documents completely independently of the printing device so that page layouts will be exactly the same on all output devices, independently of the computer on which TeX was run. With SCRIPT you have to know what output device is being used...".

  This adaption of the formatting as a function of the capabilities output device is one of the strong advantages of SCRIPT. The TEX output is always destined for a typographic (All Points Addressable) quality output device. This means that many printers can not be used to produce the output. In Europe, until a year or two ago, almost the only output devices that could be used were Versatec plotters (very low quality result), impact dot-matrix printers (very low quality result) and phototypesetters (good quality results, but expensive). More recently the appearance of APA laser and LED printers has changed this situation, but in CERN and collaborating labs these are still in the minority.

  TEX uses its own coding scheme for fonts and a large number of fonts are shipped with TEX. They are, however, not tuned to any particular output device — in contrast to fonts provided by the printer manufacturer — and produce very variable quality results. The TEX fonts made "specially"(!?) for the APA6670 are especially bad.

  It is further stated "as I write this note I find that I cannot follow my normal practice of using the SCRIPT command ".us" to generate italics for emphasis ... only the APA6670 in DD Division (which I normally use) is capable of printing italics. So today I have to change all my ".us" to ".bd" to get bold characters as a compromise solution."

  There are two aspects to this point:

  — The ".us" and ".bd" commands are part of the underlying "programming language" of SCRIPT. As user interface it is recommended to use one of the available macro packages.

In these you do not explicitly state the rendering, but you let the system select the rendering, based on its knowledge of the capabilites of each output device.

– TeX insists on the use of an output device with all "bells and whistles" and you simply can not use certain output devices at all, for example the LEP 6670. I think the SCRIPT way is preferable.

- On page 12, f and g, is stated "... with SCRIPT you never seem to know whether you have the latest version or if what you have applies to the macro package you are using." ... "The PLAIN TeX standard is very clearly defined and will never change again ... SCRIPT seems to change every year."

  It is quite true that Waterloo SCRIPT changes every year – it is a living product and new features and facilities are added every year.

  It seems implied that a TeX input that once works will work at all installations. For LaTeX this is unfortunately not true. We received some time ago a 700 line document from DEC that used LaTeX. It produced 1700 lines or error messages when formatted. In Aleph TeX has been installed and they found an incompatibility in the LaTeX level used in Berlin and in Wisconsin...

- On page 12, i, is stated "for TeX ... there are no inconvenient restrictions like always having to begin SCRIPT commands in column 1." There is no such restriction for SGML.

- On page 16 is stated "you can see a PERPRINT file but this is only an approximation of what will be printed ..." This is quite true, but is a function of the available terminals rather than a fault of SCRIPT.

CONCLUSION

The real requirement expressed in MIS – 12 is for an easy way of entering mathematical formulae. This requirement is very real and important and will be addressed in Waterloo SCRIPT very soon. I have already seen samples output produced by Waterloo SCRIPT on an IBM3812 printer and the results look very good.

This note of comments should be read together with MIS – 55 which gives a more complete picture of what I see the longer term developments in text processing will be. I very strongly agree with J.Jowett's views on stand alone word processors.