Automatic Conversion from a Scientific Word Processor to TEX

Cay S. Horstmann
Department of Mathematics and Computer Science
San Jose State University
San Jose, CA 95192-0103
408-942-0461
Horstmann Software Design Corporation
Four North Second Street, Suite 500
San Jose, CA 95113
408-298-0828; FAX: 408-298-6157
Internet: horstman@sjsumcs.sjsu.edu

Abstract

In this paper, we report on our experience with a utility which converts files written with the ChiWriter scientific word processor into TEX files. With this converter, it is feasible to write a manuscript in a "what-you-see-is-what-you-get" (WYSIWYG) fashion, with all fonts, special symbols, mathematical formulas, and tables displayed correctly on the screen during editing, and to translate the document into TEX for publication. This method has several advantages over typing straight TEX code. The word processor is easier to learn, and it is easier to revise material that is displayed on the screen without codes. We describe design decisions and limitations of our approach.

Features of Our Word Processor

The CHI2TEX converter described in this article, as well as ChiWriter, its source word processor, are commercial products, available from Horstmann Software and its international distributors. Many of the issues raised here apply to the design of conversion software from another scientific word processor as well, and some observations are valid for general purpose word processors. In the following, we will refer to ChiWriter and CHI2TEX as "our word processor" and "our converter".

Our word processor has the same capabilities as most other word processors: cut and paste, search and replace, spell checking, etc. The program operates in graphics mode. Characters in fonts such as bold, italic, Greek, and math are displayed correctly on the screen. A number of features differentiate it from general purpose word processors. Multiple superscripts and subscripts (e.g., x_1^2, x^{n_k}) are supported and correctly displayed. Mathematical formulas, such as fractions or integrals, can be entered as easily as any other text. There is no separate "equation mode" and no code language for formula entry. No separate preview step is required to view the formulas in the doucment.

An older version of our word processor (ChiWriter version 3) employs a very simple imaging model. It essentially simulates a "golf ball" style typewriter. The cursor can be moved vertically in half-line steps and characters can be placed anywhere on the screen. The user must piece together fractions, roots and integral symbols from building blocks. While this is quite intuitive for the typist and requires essentially no learning curve, it is tedious to revise formulas entered in this way. For a review of this program, see Milne.

It was quite a challenge to write a converter that is able to scan mathematical formulas in this pictorial representation and translate them into the logical structure required by TEX. Our scanning algorithm translates most formulas surprisingly well; and, with a bit of foresight, formulas can be entered to be translated reliably.

The current version of the program (version 4) supports automatic formatting of mathematical structures. For example, when editing a fraction, the numerator and denominator are continuously centered and the fraction bar expands or shrinks to the correct length. Because the word processor is aware of the structures, no guessing is required for conversion of mathematical structures and tables.

User Acceptance

Users who wrote their document using the ChiWriter word processor, then translated it to TEX and shipped a paper copy of the word processor document together with the TEX file, were generally happy. Publishers would have preferred a higher quality TEX file but resigned themselves to a one-time cleanup. The advantage of this approach is clear. The publisher doesn't have to rekey the text or cope with an alien word processor format, and the author doesn't have to spend much time proofreading since the text, mathematical symbols, and special fonts remain untouched by human hands.

We would have preferred it if users could have shipped a disk with their word processor file to the publisher and have had the publisher enter the corrections arising out of copy-editing into the word processor file before conversion to TEX. The word processor file could have been handed back to the author, preserving the changes for future revised editions. Unfortunately, publishers are reluctant to learn yet another word processing system.

Users unfamiliar with TEX expected that the converter and TEX could be used like a printer driver. They were very disappointed because they had hoped they could completely avoid learning TEX. However, some knowledge of TEX is required to produce a professional looking document with our converter. Some users abandoned TEX as a result; most others learned enough Pidgin TEX to succeed.

Other users were reluctant to fix conversion errors in the original word processor file, changing them in the TEX file instead. As a reason, several cited the amount of time required to enter the word processor, making the change there and running the document through the converter before executing the TEX program and the previewer. Some of those users finally abandoned our word processor and became TEX experts.

Most users wrote with the word processor as long as possible. Upon completion of the document, they performed a trial conversion and then corrected converter errors and added tags as required by the submission style of the publisher. These changes were made in the word processor file. Additional markup was performed by the publisher in the TEX file.

Conclusion

Many potential TEX users are justifiably concerned about the drudgery of entering TEX codes in an ASCII file. Our conversion utility, which translates files written in a scientific word processor to TEX, offers a number of advantages. The learning curve for the word processor is not as steep as for raw TEX. Fonts, special symbols, and mathematical structures show up correctly on the screen. This eases editing and revising. Typical TEX keyboarding errors, such as omitted backslashes or mismatched $ signs, are reduced. Documents can be translated into different dialects of TEX. A special font is translated directly to TEX code to access any features not provided by the word processor or converter.

There are several disadvantages. The conversion pass takes time. The user must cope with converter errors and limitations in addition to TEX problems. Sometimes the converter's actions are difficult to predict. The converter cannot detect math mode with perfect accuracy, and the user must occasionally work around the converter's guesses. The code generated by the converter contains a few nonstandard macros which may need to be modified by publishers.

Most users of this system felt that we are on the right track. They need TEX output, either for high quality printing or for submitting documents. They find that the problems of the conversion pass are far outweighed by the convenience of not having to manually enter the TEX codes, and the ease of making revisions in the WYSIWYG screen display.

Bibliography

Adobe Systems Inc., *PostScript Language Reference Manual*. Addison-Wesley, 1985.

Knuth, Donald E. *The TEXbook*. Reading, Mass.: Addison-Wesley, 1986.

Milne, J. S. "Four Word Processors with TEX Capability", *Notices Amer. Math. Soc.* 37, pages 1018–1022, 1990.