

Crouching Error, Hidden Markup

Neville Holmes, University of Tasmania

Microsoft has such a pervasive commercial presence that people working professionally in computing must expect that the company's behavior will affect their own work and reputation in many ways. It's difficult to ignore Microsoft and only too easy to snipe at the company. But a newspaper column published last year sparked my concern that Microsoft's effect on the computing profession, as distinct from its effect on the industry, might be harmful to a degree that would justify severe criticism. In "Pierpont's Long First Night with Microsoft" (*Australian Financial Review*, 26 Oct. 2000, http://www.pierpont.com.au/eco_Archives/default.cfm?behaviour=view_article&cid=3233), the pseudonymous Pierpont describes his first attempt to write a column using Microsoft Word and his fruitless and expensive struggle to get help from Microsoft technical support.

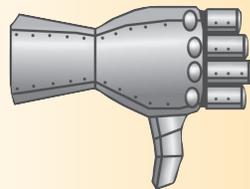
REPUTATION AND SERVICE

Although there is nothing unusual about Microsoft sob stories, that such a story received prominence in the *AFR*—Australia's equivalent to the US's *Wall Street Journal* and the UK's *Financial Times*—is most disturbing. That a reputable and widely read—if sardonic—column should end with the words "There's money to be made for old rope in IT" reflects badly on the computing profession. That Pierpont's primary readership should be precisely the people most likely to be hiring computing professionals, or funding computing innovation, makes the conclusion doubly unfortunate.

By all reports—such as Ted Lewis's "Fast, Expensive, and Horribly Com-

plex" (*Computer*, Sept. 1999, pp. 120, 118-119)—the poor quality of software plays a salient role in the poor quality of service that computer users so frequently complain about. This factor, inimical to the computing profession, has moved us to diligently develop standards intended to foster high quality.

Why have those standards been ineffective in promoting software usability and serviceability? First, it may be that the standards we have are inappropriate. Second, even if appropriate standards do exist, it may be that the computing industry has yet to accept them.



Its lack of a versatile and visible markup language can make using Microsoft Word a nightmare—and reflects poorly on our profession.

If the second answer is even partly the case, the profession has a political problem and perhaps a perceptual problem in distinguishing between architecture and engineering. But the first case—absence of appropriate standards—seems to loom large.

Microsoft Word causes users so many problems that pinning their source to any one factor, or to the absence of any one standard, can be difficult. Word might cause problems simply because it has so many users, far more than any competitive package has. I've had trouble with Word myself the few times I've ventured to use it for tasks other than printing documents e-mailed to me in Word format.

The lessons I've learned from such experiences suggest the need for developing or adopting new standards.

WYSINWYW

My first major experience with Word occurred a decade ago. My task? Compile into one publication various working groups' reports from an international workshop on design. As a long-term user of formatting programs such as Script, Roff, and TeX, I felt apprehensive about using Word's radically different approach, but as every working group had opted to produce its report using Word, I had no choice.

My actual experience proved both stimulating and traumatic. Although accustomed to the markup approach, I found Word's mouse-and-menu approach effective in setting up my skeleton document with fonts, headings and footings, contents, and indexes. I remember being delighted with Word's WYSIWYG—What You See Is What You Get—capability, which let me work with my skeleton document displayed just as I would see it in the printed version—a sig-

nificant improvement over my customary style of working with plain text.

Then the group reports began arriving. I had no problem bringing a report into my overall document, but strange things would happen. I would load a new report into Word, bring it in line with the overall design, then notice changes in other parts of the document that I had not looked at for some time. In many cases, although these changes merely altered fonts, I found a kind of font contagion at work: I would change an errant font in one place only to discover that the revision seemed to cause a font change somewhere else.

Continued on page 126

My fancy page headings and footings caused the biggest problem, costing me more time by far than any other aspect of my work. Although the final version of my document had them just right, the people at the sponsor's office failed to notice that when they added a preface to my version, it completely disrupted the headings and footings again. The published work disappointed me bitterly.

Although I like the WYSIWYG approach, I feel that its implementation in Word can better be described as WYSINWYW: What You See Is *Not* What You Want. Some of my difficulties certainly came from using Word for the first time, and on a major project. But I felt that other difficulties arose from a much more basic cause.

EPIPHANY AND ANTICLIMAX

The problem's true nature became clear when I chanced upon "Lilac: A Two-View Document Editor" by Kenneth Brooks (*Computer*, June 1991, pp. 7-19). Brooks described a system that combines overt markup with WYSIWYG. For me, Word's uncontrollability resulted from being unable to see what caused the unwanted effects. A system like Lilac would solve this problem by letting a user look at a formatted document, at the markup specifying the format, or both. The Lilac system should have been called Paradise.

Feeling sure that the spread and dominance of such dual-view packages must be imminent, I eagerly awaited developments. Eventually I discovered that, although such systems had become available, they were expensive and far from dominant.

I feel that Word's implementation of WYSIWYG can better be described as WYSINWYW: What You See Is Not What You Want.

Such a lapse could not be caused by the markup approach's inherent difficulty. After all, Word lets a user load and save documents with markup codes for formats such as Hypertext Markup Language (HTML) and Rich Text Format (RTF)—but must hide some kind of markup language beneath its own fancy façade. What Word lacks, however, is an *overt* means for formally marking up plain text while developing the document. I get the impression that Word's developers add formatting features impulsively, without the unifying philosophy or moderating principles that an underlying plain-text markup scheme would foster.

Markup

Markup conventions have a rich history. If you take a long-term view, markup conventions have been used in the data processing industry for thousands of years. Markup is conventional annotation designed to convey guidance to the user of plain text about the text's intended treatment: This guidance originally applied to how the text should be read aloud and is otherwise known as punctuation.

In *Pause and Effect* (University of California Press, 1993), Malcolm Parkes describes the remarkable history of the punctuation marks embedded in our Latin writing system. The earliest punctuation, used when reading aloud was the norm, indicated where the reader should insert pauses between rhetorical units. Writers rarely inserted spaces between words until Irish scribes in the late seventh century found it convenient to abandon the traditional *scriptio continua*. A large measure of standardization in punctuation occurred while printing with movable type was being developed, because the typesetting process made it practical to use only a limited number of conventional marks.

Recent technology

Recent technological developments have changed that situation dramatically—and for the worse. To start with, the earliest character sets the computing industry adopted contained almost no punctuation symbols. Subsequently, the main 7-bit and 8-bit character sets, ASCII and EBCDIC, respectively, included only a few more. These new standards largely ignored the printing industry's mature standards and practices, providing only some of the previously standard punctuation and other common symbols while indiscriminately adding new ones.

These poor character sets greatly hampered early developers of computer support for document formatting, who were forced to develop various markup conventions and programs based on the flawed sets. Programs such as Script and Roff proved quite useful on the relatively crude printers available at the time.

When more versatile printers became available, developers modified these pro-

COMPUTER

Innovative Technology for computer professionals

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society. **IEEE Headquarters**, Three Park Avenue, 17th Floor, New York, NY 10016-5997; **IEEE Computer Society Publications Office**, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1314; voice +1 714 821 8380; fax +1 714 821 4010; **IEEE Computer Society Headquarters**, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903. IEEE Computer Society membership includes \$14 for subscription of *Computer* magazine (\$14 for students). Nonmember subscription rate available upon request. Single-copy prices: members \$10.00; nonmembers \$20.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855. Periodicals Postage Paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail (Canadian Distribution) Agreement Number 0487910. Printed in USA.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

grams accordingly, but their lack of typographical background seemed to hobble such efforts. They put the focus on content, which led to Generalized Markup Language (GML), then Standard Generalized Markup Language (SGML), and now HTML.

Fortunately, the markup problem caught the attention of Donald Knuth. He studied the typographical tradition and brilliantly adapted the computer industry's pathetic 7-bit character set to handle text markup capable of producing documents that would make a professional compositor proud.

Knuth's markup system, called TeX, provides the kind of typesetting system that would make a superb basis for a controllable WYSIWYG document editor.

Knuth's markup system, called TeX (<http://www.tug.org>), provides the kind of typesetting system that would make a superb basis for a controllable WYSIWYG document editor. It's versatile, extensible, thorough, and traditional. It adopts rather than ignores the printing industry's traditions. It's also quite widely used by professionals such as mathematicians, who have their own particular problems with typography.

Online documents

Printed documents play a much smaller part in the computing industry than they did a mere decade ago, thanks to the many documents now available only on the World Wide Web.

The old Web is based on the more user-friendly than useful, content-oriented HTML. Curiously, at the heart of HTML's markup style is a punctuation symbol called the *diple* (pronounced to rhyme with Ripley)—a quotation mark of scribes that the printing industry developed into several, all absent from both ASCII and EBCDIC.

The new Web is moving toward the Extensible Markup Language, a more

heavily content-oriented markup language. XML is to HTML what Unicode is to the ASCII character set. Just as Unicode ignored the writing system that ASCII so poorly served, and went on to curdle the world's languages, so XML has ignored the document and gone on to curdle its bibliography. And now a widely touted extension of XML, Electronic Business XML (ebXML), presages a solely commercial future for the Web or its successor.

We need a standard markup language with the depth and versatility of TeX that would let us use a single marked-up, plain-text source file to specify a printed document and screen layout, allowing user interaction with the content and layout. Adopting Knuth's approach, or even TeX itself, would allow

- a standard for handling text in any language or language mixture that uses the Latin alphabet,
- the creative symbolism within the plain 7-bit ASCII character set necessary for trademarks and currency symbols, and
- a basis for similar standards for other writing systems.

With such support, document formatting would provide a sound basis for content-oriented standards.

By developing and adopting such a standard, the computing industry would move toward the maturity the printing industry attained during its movable type era, and its professionals might enjoy some of the respect accorded typographers and compositors in their time. Microsoft might even be able to give Word software the look and feel of new rope. *

Neville Holmes is an honorary research associate and a lecturer under contract at the University of Tasmania's School of Computing. Contact him at neville.holmes@utas.edu.au.

How to Reach Computer

Writers

We welcome submissions. For detailed information, write for a Contributors' Guide (computer@computer.org) or visit our Web site: <http://computer.org/computer/>.

News Ideas

Contact Lee Garber at lgarber@computer.org with ideas for news features or news briefs.

Products and Books

Send product announcements to products@computer.org. Contact Stephanie Kawada at skawada@computer.org with book announcements.

Letters to the Editor

Please provide an e-mail address or daytime phone number with your letter. Send letters to Letters, *Computer*, 10662 Los Vaqueros Cir., PO Box 3014, Los Alamitos, CA 90720-1314; fax +1 714 821 4010; computer@computer.org.

On the Web

Visit <http://computer.org> for information about joining and getting involved with the Society and *Computer*.

Magazine Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieec.org. Make sure to specify *Computer*.

Missing or Damaged Copies

If you are missing an issue or received a damaged copy, contact membership@computer.org.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieec.org. To buy a reprint, send a query to computer@computer.org or a fax to +1 714 821 4010.

COMPUTER
Innovative Technology for computer professionals