

ACM

Journal of Computer Documentation

Volume 26

Number 1

February 2002

ISSN 1527-6805

EDITORIAL

- 1 Stan Dicks

NEW ARTICLE

- 3 Egoless Writing: Improving Quality by Replacing Artistic Impulse with Engineering Discipline
Edmond H. Weiss, Ph.D

COMMENTARIES

- 11 Commentary on "Egoless Writing: Improving Quality by Replacing Artistic Impulse with Engineering Discipline"
Kathy Haramundanis
- 17 We Neurotic Amateurs: A Commentary on Edmond H. Weiss's 'Egoless Writing: Improving Quality by Replacing Artistic Impulse with Engineering Discipline'
Geoffrey Sauer

- 22 Consequences of the Engineering Approach to Technical Writing
Bob Waite

REPLY

- 27 Reply to Commentaries on "Egoless Writing: Improving Quality by Replacing Artistic Impulse with Engineering Discipline"
Edmond H. Weiss, Ph.D.

POLICY

- 29 JCD Editorial Policy
Stan Dicks

Editorial Board

Editor -in-Chief

Stan Dicks
Technical Communication Program
North Carolina State University
Raleigh, NC 27695-8105
Phone: (919) 515-4124
sdicks@unity.ncsu.edu

Reprinted Articles

Bob Waite
Dept. 52X
IBM Corporation
Rochester, MN 55901
507-253-7347
rwaite@us.ibm.com

New Articles

Brad Mehlenbacher
Technical Communication Program
North Carolina State University
Raleigh, NC 27697-8105
brad_m@unity.ncsu.edu

Book Commentaries

Robert R. Johnson
Dept. of Humanities
Michigan Tech University
Houghton, MI 49931
906-487-3236

Production Editor

Meredith Blackwelder
Raleigh, NC 27612
mblackw@earthlink.net

At Large

Ed Barrett, Writing and Humanistic Studies, MIT
Rebecca Burnett, Dept. of English, Iowa State U.
David Farkas, Tech. Comm. Dept., U. of Washington
TR Girill, Lawrence Livermore Natl. Lab., Livermore, CA
Laura Gurak, Dept. of Rhetoric, U. of Minnesota
Kathy Haramundanus, HP, Westford, MA
Bob Johnson, Humanities Dept., MTU
Johndan Johnson-Eilola, Tech. Comm., Clarkson U.

Susan Jones, Information Systems, MIT
Robert Krull, Communication, LL&C, RPI
Brad Mehlenbacher, Dept. of English, NCSU
Barbara Mirel, School of Information, U. of Michigan
Christine Neuwirth, Dept. of English, CMU
Janice (Ginny) Redish, Redish & Assoc., Bethesda, MD
Stuart Selber, Dept. of English, PSU
Bob Waite, IBM, Rochester, MN

The contents of *Journal of Computer Documentation* are used in compiling the following bibliographic sources:

- INSPEC (IEE, U.K.)
- ERIC (U.S. Dept. of Education)
- HCI Bibliography (<http://www.hcbib.org>)
- Recent and relevant (Society for Technical Communication)

Headquarters Staff

Mark Mandelbaum Director of Publications
Jono Hardjowirogo Associate Director of Publications
Roma Simon Managing Editor, ACM Journals
Irma Strolia Editorial Assistant

ACM Journal of Computer Documentation (ISSN 1527-6805) is published by ACM, 1515 Broadway, New York, NY 10036. Periodicals postage paid at New York, NY 10001, and at additional mailing offices. Printed in the U.S.A. POSTMASTER: Send address changes to ACM Journal of Computer Documentation, ACM, 1515 Broadway, New York, NY 10036. ACM European Service Center E-mail: ACM_Europe@ACM.ORG

Copyright © 2002 by the Association for Computing Machinery (ACM). Copying without fee is permitted provided that the copies are not made or distributed for direct commercial advantage and credit to the source is given. Abstracting with credit is permitted. For to the recopying of articles, copying is permitted provided that the per-copy fee indicated in the code is paid through the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (508) 750-8400. For permission to republish write to the Director of Publications, ACM. To copy otherwise, or to republish, requires a fee and/or specific permission.

SIGDOC Officers

Chair

Kathy Haramundanis
P.O. Box 1365
Westford, MA 01886-4865
(603)884-0317 (work)
Kathy.Haramundanis@compaq.com

Vice-Chair

Michael Carver
202-483-8024
michael.carver@geis.ge.com

Treasurer

Linda Sue Trapasso
Compaq Computer Corp.
110 Spit Hook Road
ZK02-3R44
Nashua, NH 03062
LindaSue.Trapasso@compaq.com

Secretary

Michael Albers
University of Memphis
8299 Floral Spring Dr.
Cordova, TN 38018
malbers@acm.org

Past Chair

Nina A. Wishbow
Entrust Technologies
750 Heron Road, Suite E080
Ottawa, ON K1V 1A7
613-247-1770
nina.wishbow@entrust.com

Past Vice-Chair

Stephanie Rosenbaum
Tec-Ed
P.O. Box 1905
Ann Arbor, MI 48106
313-995-1010
stephanie@teced.com

Membership Officer (appointed)

Beverly Zimmerman
Brigham Young University
623 E Heather Road
Orem, UT 84097-2349

Information Officer (appointed)

Susan B. Jones
Massachusetts Institute of Technology
MIT N42-240d
77 Massachusetts Avenue
Cambridge, MA 02139
617-253-0877
sbjones@mit.edu

Newsletter Editor

Susan Pillsbury
Kronos Inc.
2 Omni Way
Chelmsford, MA 01824
spillsbury@kronos.com

SIGDOC Mission Statement

We (1) promote the professional development of technical communication practitioners, researchers, and educators; (2) encourage interdisciplinary approaches to solving communication problems related to online and print documentation and to human computer interfaces; (3) provide avenues for publication and for the exchange of professional information; (4) support research that focuses on the needs of humans and their goals and tasks in technological contexts; (5) support the development and improvement of computer-supported technologies.

We Neurotic Amateurs: A Commentary on Edmond H. Weiss's 'Egoless Writing: Improving Quality by Replacing Artistic Impulse with Engineering Discipline'

Geoffrey Sauer

Dept. of Technical Communication, University of Washington

geoffs@u.washington.edu

<http://www.eserver.org/geoff/>

Abstract

The assertion that technical communicators tend to be 'amateurs'—that is, lovers of their work—is a claim with little foundation. Arguments toward regimentation and systematization of documentation writing are not calls to professionalize a currently-immature field, but rather attempts to emulate the hierarchy we have seen implemented in microprocessor engineering in the 1970s, software development in the 1980s, and content management in the 1990s. Such 'egoless' methods may offer advantages to employers, but should not necessarily be considered 'progress.'

I.7.1 Document Preparation—document management

Keywords: *documentation development, project management, collaboration, teamwork*

We Neurotic Amateurs

Edmond H. Weiss asserts in his paper 'Egoless Writing' that some technical communicators tend to lack professional distance from their work, wallowing as writing *amateurs* (in its original sense—'lovers' of our own writing). He suggests we instead follow computer programmers, whom he believes have been transformed by integrated development environments (IDEs) from neurotic individuals into capable team members, collaborating on large group writing projects. In his article, he hints that a coming generation of content management systems (CMS) will offer technical communicators opportunity to follow software developers into professional status as 'engineers,' rather than to remain in a limited, semi-professional status.

This paper will disagree somewhat, arguing that CMS tools cannot and will not by themselves, improve the professional status of writers. Indeed, it will suggest that in some ways CMS development environments may subordinate technical communicators within hierarchical divisions that may be ill-suited to our work. Instead, this paper will suggest that we must develop theories of our work which allow us to recontextualize collaboration in documentation writing into forms that will serve the interests of both professionals and

practitioners in the field.

A nineteenth-century sociologist once argued that some large-scale questions about society cannot even be asked until the conditions for their answer have already come into existence. While this paper will disagree with Weiss about the particular strategy our field should adopt with relationship to CMS tools, we can credit him with having joined other authors (such as Corey Wick and Bill Hart-Davidson) in posing this question early enough to allow us to consider the implications of network-enabled, collaborative writing to the profession.

The Problem Case: Bad Writers Exist

Weiss's argument is structured around a problem case he presents in his first paragraph: intensely motivated writers can lower productivity and compromise the quality of written product (p. 3). He is certainly correct; bad writers do exist, and they can hamper quality technical communication.

But while each of us may remember one or two such writers from our workplace experience, I am by no means persuaded that such a problem is widespread across the field as a whole. When it does occur, it seems to result from individual neuroses or a management environment in which writers can find little with which to identify at work besides their own product. If one cannot identify with one's readers, colleagues, work process, or managers, and cannot distinguish one's personal life from one's work, then anyone might become attached to forms of work at hand. American popular psychology became fascinated with theories about such 'object-choices' in the 1960s and early 1970s, but largely abandoned these after finding them only useful when contrasted against a 'normal' against which to judge 'deviant' object-choices. Such methodology hardly seems conducive to either (a) identify a new problem, or (b) explain how *new technologies* offer serious prospect to resolve such issues.

In this paper I will respond to Weiss's discussion and his suggestions, arguing that the new technologies of CMS and single-source documentation provide an opportunity for us to theorize a new relationship for technical communicators within industry—not merely to follow either business software or other workers' past models, but to expand and develop our existing roles.

Writing and Programming

Weiss opens the 'Writing and Programming' section of his article with the contention that writing and programming are alike. I would agree: the model of professional production that has emerged in software development in the past fifteen years looks very much like collaborative writing, and is an important model for the future of technical communication.

But not, I should say, for the reasons he says. His hermeneutic theory, for example, that readers are 'programmed' by documentation the way that personal computers are by C++ (par. 4), seems exceedingly simplistic. In the mid-1990s several theorists argued that technical writing facilitated instrumental relationships between author and reader (see for example Moore 1996, Charney 1996). Such communication, these theorists argued, transformed readers into performing agents, capable of accomplishing needed tasks. But this instrumentalism has fallen out of favour in recent years to explain communication practice, as documentation has accepted the need to address readers of diverse skills and background, and as it has come to integrate subjunctive and indicative moods into their prose, to complement the traditional imperative of procedural lists.

Weiss goes on to suggest that document modularity promotes communication clarity:

[P]rograms and publications both benefit from modularity: chunking the project into small, manageable components (p. 4).

Document modularity is hardly new. It is the model used in many texts, including the 18th century U.S. Code, which is maintained and updated nearly continuously by a large collection of teams (Congressional committees and subcommittees). Weiss writes that the greatest hazard of solo development, however, was the fact that it made individuals irreplaceable:

This last problem—the opacity of the code and intractability of the logic to anyone but the solo programmer—led to the greatest difficulty of all. It was nearly impossible for other programmers to maintain or modify the work of the original programmer. (par. 8)

This was a problem which occasionally frustrated programmers, but which proved most difficult for employers. Developers in the 1970s and early 1980s

quickly learned to make themselves invaluable, ensuring high salaries and secure long-term employment, and quickly defined themselves as irreplaceable—which enabled some to move into upper management.

The object-oriented programming languages (OOPs) and integrated development environments (IDEs) Weiss subsequently refers to were specifically popularized to counter this new-found majesty for programmers. By developing project teams and narrowly-defined modular tasks that followed well-defined guidelines for application programming interfaces (APIs, the successors to the sub-routines Weiss cites from Yourdon and Constantine), it became in the 1980s once more possible for programmers to be replaced without jeopardizing entire projects. This, in conjunction with a large number of newcomers to the profession, led to a significant “rationalization” of developers’ salaries to their current levels, the elimination of high privilege for individual developers, and the creation of several levels of hierarchy in the software development process—including by the 1990s the introduction of MBAs as project managers for such projects. Weiss quotes Weinberg, who discusses the advantages of OOPs and IDEs for employers firing a developer: “the entire work is less susceptible to being disturbed if one of the involved programmers happens to be...missing.” (Weinberg, p. 59)

This transformation did not occur either smoothly or gracefully. We can find discussions of this change in writings of the 1990s, including Neal Stephenson’s *Snow Crash* (a problematic but bestselling novel about brain-washed, robotic programmers from Asia invading California to replace innovative American programmers). In the 1990s open-source development prospered, self-organized amateur development projects that used the volunteer labor of programmers who have salaried ‘day jobs,’ but who in the evenings work on programming tasks they care about. Indeed, one could credit much of the 1990s Internet revolution to the actions of developers alienated from corporate project management (until the later part of the decade, when corporations finally took them as well, to integrate them into corporate software products).

It would make sense that employers would wish to reproduce the logic of IDEs in technical communication. In the past decade, salaries in our field have risen steadily, and before the dot-com crash of 1999, unemployment in the field was quite low. Though an informal hierarchy has existed for some time between some of our colleagues (between ‘senior’ and ‘junior’ technical communicators in firms, or ‘professionals’ and ‘practitioners’ in other discussions), the integration of IDEs/CMS would enable the

modularization of writing and flexible reorganization of personnel between projects.

In Weiss’s hermeneutic theory, documentation exists to reflect reality. In his theory, then, our aim should be to eliminate errors (to increase the accuracy of the reflection), and to generate documents that permit easy update and repair.

But this theory of meaning, while simple to understand, can’t explain important aspects of documentation which we should plan for in future development of our field, such as adaptive documentation which accommodates the experience and background of the reader, providing information at an appropriate level to readers of diverse competencies. Theories of meaning to accommodate such tasks as these might require a hermeneutic theory to concede that documentation actually *constructs* the meaning of an object for readers. This latter theory could explore, for example, how to localize texts for global reading audiences, how to accommodate readers searching for answers to a particular question as opposed to those seeking to understand a product more generally, and how to integrate Windows Help, HTML Help, XML Help, printed manuals and product websites into a coherent constellation of products.

Document Quality versus the Writer’s Ego

At a few points in his argument Weiss appears to forget himself. Pleased with the implications that engineering discipline must inevitably come to technical writing, he appears to forget his specified task is to *persuade* us this represents ‘progress.’ In his critique of writers, he claims:

It is safe to guess that the least regimented and most egoistical employees in companies are the writers. And even workers who are usually disciplined and team-oriented tend to become artist-like when the assignment is to produce a long publication (p. 6).

Writers such as Walter Benjamin, Roland Barthes, Michel Foucault and others have also discussed the attachments readers and writers have to written works. But their attempts to do so have been less adversarial, and written in an interest to improve the sophistication of both readers’ and writers’ engagement with written works. Weiss appears to favor the opposite. The artist, he suggests, views readers as independent and active (par. 40). Engineered books provide more structure and encourage readers to rely upon the author. Perhaps his argument is to persuade authors to wish to become the disciplined engi-

neers who might serve *in loco parentis* to hapless readers, but the current argument appears merely to patronize both groups. Perhaps future elaboration will permit more sophisticated articulation of the advantages to readers and writers of accepting the roles CMS writing offers, and the roles for creative contribution which must exist, but appear absent from his discussion here of document design:

[T]he new model even militates against letting the technical communicator control the layout and appearance of pages. Such technologies as SGML, HTML, and XML underscore the sudden abrupt change in the power of the technical communicator: from perfect control over every element of the page to provider of "tagged input" (p. 9).

This is, of course, only partly correct. The rendering model found in SGML and its derivatives (such as HTML and XML) distinguish between *form* and *content* of documents. This in no way militates against one person creating both; it merely divides the labor into separate tasks. Some systems may 'lock out' authors from the formatting style sheets, but this is the result of intentional acts by management, rather than the system or model militating against authors.

Professionalism

Weiss discusses leaders' and managers' role in CMS implementation as if it were the obvious implementation of the best result for everyone. He writes:

Wherever egoless methods have been introduced, in whatever discipline, usually they have been supported by leaders and managers—those concerned with overall efficiencies and hard measures of performance—and resisted by those whose egos were judged part of the problem. This should have surprised no one (p. 9)

Certainly not. Managers' interests lie in increasing efficiency by reducing costs. In documentation projects, this happens at the expense of technical communicators. But to theorize that this would be resisted by writers due to *vanity* seems hardly the best use of Occam's Razor.

Weiss goes on to assert that most of the problems of

the documentation have been 'solved'— and therefore in the future, less-talented people will be needed. (par. 48) This seems most unlikely: I would prefer to argue that current expertise in usable documents should be exported into software development, particularly user interface design, web design, interactive multimedia systems, and other genres.

But the appearance of solution may be understandable, given the lack of innovation in recent years. The software industry has seen a fifteen-year period of relative stability. Since the Post-Script desktop publishing revolution of 1985, virtually every genre of software has seen relatively stable competition between two or three major products, and the three major application categories of the 1970s (word processing, spreadsheet, and database) have merely been augmented by a few additional 'killer apps': desktop publishing, e-mail and web browsing.

But Weiss speaks as if this current stability were permanent. He writes:

The current epoch of egoless technical communication could be staffed with people far less talented than the current cohort of North American technical communicators. The intellectual leadership in this field developed their skills and reputations when the manual was still an interesting technical issue.... most of the problems have been solved and the profession has been dramatically altered. (par. 53)

This might be true if the current stability is to be permanent.

If, however, the current stability were unstable—if, for example, a new generation of hardware were to emerge (after personal computer progress which has been relatively constant since 1980). Or if a new form of document were to permit small 'applets,' such as Java and ActiveX promised in the mid-1990s, then the current stability might well become unglued, and our field would have need of innovative and talented technical communicators who might help users to understand the change and each new product.

It remains the readers' choice whether to conclude we have at last achieved an Hegelian 'end of history'—the culmination of innovation in the hardware and software revolutions. Or whether it would be precipitous to restructure technical communication in support of hardware and software as a stable industry.

Conclusion

I write this article not as a reactionary who fears content management systems, but rather as the director of a large one, called the EServer (<http://eserver.org/>)—an online publishing center that organizes the work of 281 writers and editors to publish over 32,000 works (including a large database-driven, annotated bibliography for technical communication). Since 1990 the EServer has worked to develop viable theories to describe the roles which writers should play in the new space being defined by electronic publishing. In some cases, we have been successful; the site has influenced writings about the software and content management industries, and hopes to continue to do so.

But I would suggest that the task is far from complete.

Theory is not written because academics have no 'real work' to do. Theory is the means by which we explain to ourselves (and to others) what we do, how we should do it, and why we do it. Theories may sometimes be seen as rationalizations of current circumstance (indeed, several social theorists have written about this tendency), but the most useful theories are those which identify emergent trends and patterns and produce arguments to mobilize effective responses.

It may be novel for technical communication to theorize about the dangers of its success in the past decade, but I would suggest that—rather than accept willy-nilly CMS technologies as an answer to our questions about future directions for the field—we instead work to generate theories which will lead us into a new relationship with both writing and engineering.

Weiss concludes his article with the contention that technical communicators have a choice: either the mines, the future, or become librarians. Look at his language to see how each options is described:

They can continue to resist egoless methods, swim against the tide in pursuit of personal satisfaction. Or, they can aspire to become the central cadre of inventors and designers, learning to delegate the "writing" to the miners, here or off

shore. Or they can become "knowledge workers," people whose job is less the generation of information than helping people to find it, use it, adapt it, or deal with the frustrations of a workplace rich in data but poor in insight (p. 9).

Toward the end of his influential book, *The Structure of Scientific Revolutions*, Thomas S. Kuhn speculates about the reasons major disciplinary theoretical shifts become accepted, qualifying as what he terms 'paradigm shifts.' Reviewing examples from the history of science, Kuhn concludes that paradigm shifts occur when a field recognizes among the implications of a new paradigm the promise of more and better jobs for people in the field (Kuhn 1970). While it is conceivable that Weiss's model does imply this for us, I cannot myself find that in his argument. I would therefore suggest that rather than simply to conclude CMS to be ineluctable, or to accept fixing errors to be our highest ambition, that we rather begin to devise techniques of writing that will make use of CMS facilities to engage readers more effectively to communicate desired information.

References

- Charney, D. (1996). Empiricism is not a four-letter word. *CCC*, 47, 567-593.
- Flower, L. and Hayes, J. (1981). A cognitive process theory of writing. *CCC*, 31, 365-387.
- Kuhn, T. (1970). *The Structure of Scientific Revolutions*, 2nd ed. Chicago: University of Chicago Press.
- Miller, C. (1979). A humanistic rationale for technical writing. *College English*, 40, 610-617.
- . (1978). Technology as a form of consciousness: A study of contemporary ethos. *Central States Speech Journal*, 29, 228-236.
- Moore, P. (1996). Instrumental discourse is as humanistic as rhetoric. *Journal of Business and Technical Communication*, 10 (1), 100-118.
- Yourdon, E. (1992). *Decline & Fall of the American Programmer*. Englewood Cliffs, NJ: Prentice-Hall.