

edmond weiss consulting

115 Cooper Rd., Voorhees, NJ 08043
856-753-3440/edweiss@aol.com/www.edmondweiss.com

Bits, Atoms, and the Technical Writer: The Rhetoric of STOP

Edmond H. Weiss, Ph.D.

This article appeared in a similar form in the *Journal of Computer
Documentation*, August 1999

Bits, Atoms, and the Technical Writer: The Rhetoric of STOP*

Edmond H. Weiss, Ph.D.

Preface: Bits and Atoms

In the mid 1960s, Tracey and his associates at Hughes aircraft proposed the radical idea that writers are responsible for the document as *received* by the reader, rather than just for the content alone. Although working in an era of rudimentary word processing and primitive publishing software, they proposed that a well made publication should be “quantized,” chunked into modular units, so that the logical book (bits) and the physical book (atoms) would be isomorphic. To achieve this, despite their primitive tools, they created a top-down, engineered approach to technical communication in which motion-picture-like storyboards enabled large teams of writers to work in parallel—like animators on a cartoon movie—to design and produce a unified, cogent, responsive technical proposal.

For about twenty-five years thereafter, emerging computer technology made the STOP model ever easier to implement. Gradually, word processing and page making software converged and overlapped, so that it made perfect sense for technical writers to create “spreads,” rather than content. In effect, the only practical barrier to STOP—the difficulty of using the single page or the spread as a work unit—was obviated by software.

Starting in the early 90s, however, a countertrend developed, in which authors would be responsible again for content (rather than pages), this time “tagged” with SGML (later HTML) markers of its logical components and hierarchy. Documents, in this view, revert to data, with users and readers empowered to extract countless *virtual* documents from accessible databases. Writers create bits; everyone else turns them into atoms.

In other words, the assertive rhetoric of STOP—in which authors took complete responsibility for the actual physical form of their message—is yielding to a passive or neutral rhetoric in which writers create “resources” and the readers/receivers extract and shape the message to suit their preferences. Writers publish bits; readers turn them into atoms. Is this change for the better?

How STOP Came into my Life

* Tracey, J.R., Rugh, D.E., and Starkey, W.S. *STOP: Sequential Thematic Organization of Publications*. (Hughes Aircraft Corporation: Ground Systems Group, Fullerton, CA, and January 1965; reprinted in *The Journal of Computer Documentation*, August 1999.)

I remember well the day in the late 1970s that the Hughes Aircraft STOP manual came into my possession. I had been conducting a seminar on proposals for an aircraft company in the suburbs of Atlanta when, during the afternoon coffee break of the second day, one of the engineer participants pressed a stapled version of the publication into my hands. “Read this,” he said, with the sort of quiet intensity one hears in the voice of a recent convert distributing tracts.

That night, on my return flight, I scanned the document—a grainy seventh or eighth generation photocopy of the original. And, of course, I was struck at once with the power of the concept. STOP was a writing technology that could enable several authors, of widely diverse writing skill, to collaborate on a persuasive technical proposal. And to produce a document of coherent two-page spreads, “two-page illustrated essays,” far more readable than any proposal or technical report I had ever seen. *For the first time I encountered an approach to writing that made life easier not only for the writers but also the readers.*

For the next several years, I tried unsuccessfully to get my various R&D clients to adopt the STOP technique; I argued that it was the only important innovation in technical communication since Aristotle. Instead, my clients insisted that the method was impractical; one even called it *procrustean*, forcing me to learn the meaning of that useful word. I even tried, again unsuccessfully, to persuade Prentice-Hall to publish my first book, *The Writing System for Engineers and Scientists*, in the modular format. My acquisitions editor refused, saying that the method obviously wasted paper. (Moreover, he could not see the logic of assigning a book designer to a project before the manuscript was submitted.)

Eventually, I might have been forced to give up my crusade for STOP, were it not for an accident of marketing hype. In early 1980, a training company client, sensing that the central problem for technical writers was becoming computer documentation (this was still before the PC, remember), engaged me to develop a course for the people who documented mainframe systems and applications. The course, I was informed, was to be called “Structured Documentation,” because, in that decade, structured methods and technology were “hot” and because IT people (then called EDP people) would, according to my client, sign up for any course with the word “structured” in the title.

I groused a bit about the cynicism inherent in this sales approach, even lobbied for my own course title: *Systematic Documentation*. In the end, though, my client prevailed and I was obliged to research what EDP folks meant by “structured,” so as to incorporate the concept into my seminar.

What I learned in that season of research was that the concepts of structured documentation—the top-down development process, the chunking of programs into small self-contained modules—were astonishingly analogous to the concepts of STOP (which,

incidentally, had been developed earlier). It was out of this research that I formed my “structured documentation hypothesis”:

- Documentation affects readers the way software affects hardware,
- The flaws in software are quite analogous to the flaws in documentation, and
- The methods developed to eliminate those flaws in software can be applied, with slight adaptation, to the management and design of documentation.

It became my destiny, then, to preach the word of STOP to the burgeoning community of software writers of the 1980s. In two books, and hundreds of seminars, I recommended my own adaptation of the STOP method. And, although I always credited Tracey *et al* with the invention of modular documentation, I often received unfair credit for what they had done. (Of course, I usually took the blame when the method was not implemented correctly. [See, for example Hoft, 1995, p.138])

The Central STOP Precepts

When first exposed to STOP, or my own variation (structured documentation), most people react to its striking outward manifestation: the organization of the publication into standard-sized two-page spreads, each supposed to be self-contained and independent of the other parts of the publication. Organizing the universe of discourse into two-page chunks, as procrustean as that may seem, is a startling innovation in itself. (The Toronto chapter of STC once had a banquet at which I was the featured speaker; they called the dinner “The Two-Page Spread,” and published a suitably designed menu.)

It is a mistake, though, to confuse the product of STOP with its process. The STOP method is nothing less than an assault on the traditional notion of how technical publications should be put together. This traditional method—I have called it the “artistic stereotype” elsewhere (Weiss, 1991, p. 40)—entails a solitary writer, working alone, without clear specifications, and with no obligation to produce intermediate work products. Between the initial vague definition of the book needed and the first review draft, no one but the author sees the content; moreover, by the time the draft is ready for review, nearly all the time available for the project has been exhausted, leaving no time for serious testing and revision. (See Weiss, 1988)

STOP, remember, was developed for technical proposals with *fixed due dates*. Unlike most other business and technical communication, where due dates are arbitrary and mutable, proposals *must* be in on time—or else. These immutable deadlines exert a powerful force on writers who, now, must think of themselves as “publication engineers.”

At first, the term *publication engineer* struck me as an affectation. Over time, though, I realized that what replaced the “artistic stereotype” was best defined as an “engineering stereotype,” wherein documents develop through a series of increasingly richer (finer-grained) models, so that, eventually, the final version drops out. In software engineering, this approach is called

top-down implementation, or testing the interfaces first; it is a method based on the simple observation that, in complicated messages, the complications are in the links and connections (interfaces), not in the short, simple processes that are connected. That is, as in documentation, the difficulty is not in writing the short procedures (“how to rename a file”) but in guiding the readers through the maze of procedures to the one they need at any given moment.

In this context, STOP is built on three overarching principles:

First, documents must be **planned, designed and written from the top down**, through an elaborate system of purposeful outlining. Starting with a central aim or goal for the entire project, each constituent must contribute to the larger purpose of which it is part. Moreover, the labeling/titling of each component should be *thematic*, more headline than heading, making clear the contribution and function of each part and permitting not only tests/walkthroughs at the earliest design stages, but also a thematic table of contents will fully “traceable” components. (In effect, usability testing begins with the outline, not the final draft.)

Second, the design **quantizes the publication** (or suite of publications) into a set of small, well-specified, self-contained modules, each fitting into a standard two-page spread, containing a synoptic presentation of all the text and illustrations needed to grasp the “theme” of that module. These modules, like their counterparts in software, can be tested/reviewed independently, reused in other settings, and maintained by replacement. Moreover, these modules have the double effect not only of breaking the writing job into easily managed tasks that can be shifted around a project manager’s schedule when the critical path is in trouble, but also, at the same time, radically reducing the reader’s burden.

Third, the writers are developing a **book—not just the content of book**, in many cases editing and framing the material to satisfy the predefined format of the publication.

The Unique Claim

Again, what is most remarkable about STOP, beyond these innovations of process and appearance, is that it refutes a principle that I once believed to be a Law of Documentation: **That which reduces the burden for writers will increase the burden for the readers** (See Weiss 1981). Historically, technical writing standards and practices have been meant to benefit the publishing organization, not the readers. Non-sequential page numbers, supplemental update sheets, separation of text pages from figure pages, rules against putting “hard carriage returns” (page breaks) into text files... these and other rules have the primary effect of lightening the load of document maintenance.

I have always judged this inverse relationship between the exertions of the reader and writers to be not only one of the technical foundations of the writer's profession, but also its *ethical* foundation. That is, good writing, writing that places the smallest possible burden on the reader, entails *sacrifice* from the writer; the writer works harder, so that the readers will work less. (Generally, I cannot improve the writing of a client who is indifferent to the burden on the readers.)

STOP, however, in nearly every instance, reduces dramatically the stress of designing, drafting, editing, testing, maintaining, and modifying publications—while simultaneously producing the most readable technical documents ever published. It maximizes the talents of the best writers, fully exploits (and disciplines) the work of the worst, allows for the earliest discovery of flaws, supports project management methods and technology, clarifies cost/time/scope tradeoffs, simplifies page and figure numbering—while simultaneously yielding books that can be read with a minimum of searching, branching, looping, and perseverating on what software engineers call “pathological interfaces.”

Moreover, it is a process that can be learned in a week and mastered in about a month. Indeed, with the right templates and software tools, in two weeks.

How did this seemingly unnatural result occur? How could a method that enables fifty engineers to assemble a winning proposal by an inflexible deadline also, at the same time, produce documents that are more readable than nearly anything else in print?

I suspect that the key is the realization by Hughes' publication engineers that the problem was in the *form of books themselves*. That is, what traditionally made the management of technical writing so difficult was the indifference of most writers, professional or amateur, to the ultimate destiny of their texts and pictures; the “content,” which ultimately must be “fit” into books, was typically unsuitable, a “river raft” of uneven, meandering, arbitrarily linked sections, arbitrarily cut and pasted into pages and sections. Similarly, the greatest difficulty for most readers of typical technical documents was the near independence of the structure of books from the natural process of inquiry and study. Much of the energy exerted in reading technical publications was for *finding*, not reading: framing questions, interpreting cryptic chapter and section names, pursuing chains of GOTOs in search of a simple fact or process.

In sum, Tracey *et al* decided that it would no longer be acceptable for technical publications to be a dump of content (bits), arbitrarily packaged and delivered in tangled, GOTO-filled books (atoms), from which readers were obliged to extract coherent messages (virtual publications) for themselves. Their solution was to treat the book as a visual presentation (not a box of content) and to use the natural physical organization of books—a series of two-page spreads, whether the authors intend it or not—as the infrastructure of a new technical rhetoric.

STOP's initial focus, recall, was on proposals. The publication technology they devised was shaped by two powerful constraints:

- It must be accessible to demanding, pressured readers.
- It must be capable of being produced by teams of writers working in parallel.

To satisfy their proposal readers they needed to create a series of “thematic,” engaging, self-contained arguments and demonstrations; in other words, their approach was driven by a desire to produce readable, persuasive messages in which all the text and figures could be apprehended comfortably by even skeptical and overwrought military proposal reviewers. (And in which all the requirements stipulated in the Request For Proposal were easily “traceable.”)

To satisfy their production demands, they needed a new concept of outlining (storyboarding) which would generate a list of specifications for standard-sized modules, each defined clearly enough to permit its author to work independent of all other authors.

The wonderful accident in all this is that the two-page spread, which became the frame of the STOP module (and also of structured documentation), seems to be *more than an accident*. Unlike most standards, which are arbitrary, the two page chunk—500 to 1000 words, a picture or two—is a comfortable, feasible, intellectually satisfying way to parcel writing or reading into manageable units. It is large enough to contain mature concepts and processes, small enough to provide the economies of modularity, but not so small that it creates interface complexity. Its top-down development permits early testing of models/outlines, and the language in those storyboards generates headings and summaries that guide the reader to right sections.

STOP versus Rhetorically Neutral Publishing

To implement STOP, publication engineers/writers must decide exactly what they want the readers to see, in exactly what order. In other language, STOP is the most rhetorically assertive form of technical communication. Deriving its main design tool, storyboarding, from the motion picture industry, STOP generates a series of standard-sized frames of information, passing through the reader's attention like frames of film through the gate of a projector. It tries to control the pattern of the readers' attention, discouraging them from overriding the authors' plans. Ideally, the readers have only slightly more freedom than the movie projector.

Once, when I was teaching the method to a group of technical writers, the head of the group complained that I was teaching *presentation*, not report-writing. And, interestingly, she was right. STOP documents are presentations, meant to be read in a unique sequence, conceived as a series of visual objects containing text. They are the author/rhetor's idea of the most effective way to engage and sustain the attention of the readers. And as such they are grounded on Aristotle's conception of rhetoric, in which effectiveness is the product of the communicator's

resourcefulness in organizing material so that it commands attention and wins assent from the audience.

Indeed, the goal of STOP has been to eliminate one of the main rhetorical differences between speech and writing: the audience's dangerous and self-defeating ability to manipulate the message. Live speech, the foundation for classical theories of rhetoric, is a message entirely structured by the sender; when a live speech adapts to reactions from the audience, it is the *speaker*, not the audience, that changes the message.

Books, though, have traditionally allowed considerable independence in reassembling their parts and creating sequences unimagined by the authors. In the 1980s, this independence of readers, and the likelihood that readers would become lost and confused, was perceived as a technical communication problem to be solved. Just as a blank screen with a blinking cursor gave users too much freedom and needed to be replaced with "usable" graphical and menu-driven screens that circumscribed the users' options and made it harder for them enter invalid data, so did modular and structured documents discourage readers from creating their own books and becoming lost in the process. The reader of a book can never be as restrained as the hearer of a speech, but, at the least, document designers can reduce the amount of skipping and looping and increase the chance that the intended message is processed in the intended order. (See Weiss, 1993, 1995)

In STOP, or structured documentation, writers assume responsibility for the atoms (the physical book) and commit themselves to understanding and supporting the needs of readers, which often entails protecting the readers from themselves.

But this concept, born in the 1960s, nurtured by disciplines as diverse as instructional design and software engineering, supported by increasingly powerful software, is increasingly discounted by a new set of concepts, roughly grouped under the rubric of *information architecture*. Why the change? How has the need for STOP rhetoric been satisfied by alternative technologies?

First, despite remarkable improvements in their design and content, **manuals remain unpopular and largely unread**. Especially in the domain of software, the myth persists that well-designed software needs no manuals, even though software users who hold that view generally limit themselves to a small subset of features. (Today, expensive software from prestigious companies often ships with little or no paper documentation.) For whatever reasons—laziness, poor books, the decline of the public schools—manuals, which used to be a last resort at best, are, for many users, not even that. Moreover, most of what used to be found only in manuals is now imbedded into interfaces and system messages themselves.

Second, **didacticism has been discounted as a teaching method**. The current fashion in all forms of educational communication is to encourage exploration and independence in the readers/students. (People who work with Windows also accept, with patient equanimity, unexplainable results and mysterious events, problems that have no apparent cause, which are solved through no understandable solution strategy.) Astonishingly, the word *lecture* has become a term of opprobrium; instead of referring to the quintessential rhetorical medium through which centuries of knowledge have been communicated from teachers to students, *lecture* now connotes a message in which the audience members are somehow coerced into hearing what the speaker wants them to hear. Similarly, many of today's users/readers do not want to be "helped" by technical communicators; I have heard manuals referred to as "a paper burden" by people who claimed that they would rather have direct access to the developer's raw files than to the documentor's conception of what the users need.

Third, and probably most important, **hypertext has solved the problem of the book**. There is no longer a need to package multidimensional, link-rich materials into an ordinary, linear book; there is no need to reduce the quantity of looping and branching in a hypertext message, when all pages are equally close, all searches are equally stressful, every text has an imbedded index, and when readers can explore without fear of losing the "history" of their moves. Except for narratives (fiction or non-fiction), hypertext/media is an exhilarating alternative to traditional reading. And, consequently, today's writers prefer to place their modules at the nodes of hypermedia networks.

Fourth, the introduction of **universally accessible libraries** into every computer, through the constellation of technologies called the Internet, has ended for most people the dependence on all forms of traditional documentation, especially rhetorically designed manuals. Indeed, today's reader—whether stuck on an impenetrable system message or in need of the answer to a scholarly question—can unleash a set of scholarly and research tools that dwarf the scope and power of the Internet's predecessor: Inter-Library Loan. While the user of the 80s often worked in an information vacuum that needed to be filled by technical writers, the user of the 90s works in an information plenum, where the truth is probably out there somewhere, and where all one needs is navigational aids.

The entire zeitgeist favors the creation of bits over atoms. Usually, working in HTML means relinquishing the author's control over typography, page breaks, column widths, and other basics of document design. Today, even I urge my documentation clients to pursue the goal of "single-source" documentation (one data base entry accessible in a variety of media); this means, to a large extent, putting away PageMaker® and Quark Xpress®. (NOTE: I checked the correct spelling of these two product names by making a quick visit to AltaVista. And a second trip to check on the spelling of AltaVista.)

If structured documentation still has an accepted place, it is in publications for beginners. For example, there is an impressive library of software guides in the *Marangraphics* series (named for the writing Maran family, www.maran.com) which, in addition to its many other virtues, uses two page modules. And there are quite a few publications for novices from CUE press that also use the format to differing degrees. Apparently, only inexperienced readers need a logical, accessible message; everyone else, presumably, prefers the ambiguity of navigating through the information plenum.

Perhaps Sandra Pakin was right in the late 1980s when she told me, during an STC conference, that she feared that my work, and the work of other popular documentation consultants, was overly concerned with the needs of the novice, insecure reader. She observed, wisely, that the design elements used to make documents accessible to neophytes would reduce the acceptability of the documents to more resourceful and intrepid readers, who, she reminded me, were in the majority.

I am growing convinced, but not entirely. This month I was engaged to evaluate and edit about 300 pages of reports and correspondence from a major pharmaceutical company. What I saw was messages with no understandable theme or purpose, documents without understandable titles or summaries, pages without predictable structure, tables several pages removed from the text that analyzed them, prolix detail followed by elliptical arguments, cryptic tables of contents, accidental page breaks... I saw a host of problems that could have been solved by STOP.

References

- Hoft, N. L. (1995) *International Technical Communication*. New York: Wiley.
- Weiss, E.H. (1981). *Five Laws of User Documentation*. Computerworld, December 7.
- Weiss, E.H. (1988). *Usability: Stereotypes & Traps* in E. Barrett (ed.), *Text Context and Hypertext*. Cambridge: MIT Press.
- Weiss, E.H. (1991). *How to Write Usable User Documentation* (2/e). Phoenix: Oryx.
- Weiss, E. H. (1993). *Of Document Databases, SGML, and Rhetorical Neutrality*. *IEEE Transactions on Professional Communication*. 36,2. pp. 58-61
- Weiss, E.H. (1995). *The Retreat from Usability: User Documentation in the Post-Usability Era*. *The Journal of Computer Documentation*. March, pp. 3-18

Seminars, Courses & Speeches from Edmond H. Weiss

(Each course can be tailored to the client's interests and time requirements.)

<i>Business/Professional Communication</i>	How to Sell in Writing (Proposals & Business Cases)
	How to Write <i>Globally</i>
	Final Draft: The <i>Especially</i> Clear Sentence
	The Art of the Pitch
	The Art of the <i>Engineered</i> Memo
	The Art of Effective E-Mail
<i>IT/Technical Communication</i>	A Writing System for IT/Technical Professionals
	Preparing English Tech Documents for International Readers
	Effective Quality Manuals/ Usable Procedure & User's Manuals
<i>Organizational Communication</i>	Turning Words into Money: Business Plans & Cases
	Meetings that Work
	There's Only Now: Managing the Professional's Time
	Raising Culture Consciousness

Books by Edmond H Weiss

These and other works are available at Amazon and other book sources.



The Elements Of International English Style: A Guide To Writing Correspondence, Reports, Technical Documents, And Internet Pages For A Global Audience by Edmond H. Weiss (**Paperback** - Feb 28, 2005)

Buy new: **\$21.95** 21 Used & new from **\$14.27**

★★★★★ (4)

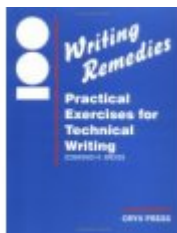


How To Write Usable User Documentation: Second Edition by Edmond H. Weiss (**Paperback** - Jun 26, 1991)

Buy new: ~~\$32.95~~ **\$19.67** 28 Used & new from **\$6.98**

★★★★★ (5)

Books: [See all 45 items](#)



100 Writing Remedies: Practical Exercises for Technical Writing by Edmond H. Weiss (**Paperback** - Sep 4, 1990)

Buy new: **\$25.00** 15 Used & new from **\$2.50**