

Reviewed by Ronald A. Sudol, Oakland University

The Modern Language Association, concerned that indifference might prevent its members from effectively using the new information technologies, has initiated a series of publications entitled Technology and the Humanities under the general editorship of Hans Ruttimann. The first number in the series is A Grin on the Interface: Word Processing for the Academic Humanist, a collection of essays edited by Alan T. McKenzie about both the technology and experience of word processing.

In "Some Words for the Trepid," Carolyn Heilbrun offers a narrative with commentary about her own initiation into word processing. The plot includes such familiar themes as the computer industry's ability to intimidate, the need to depend on supportive colleagues, the confrontation with the computer's stolid logic, the fretting about technological advances rendering even the newest hardware obsolete, and the ultimate triumph over all obstacles. Mindful of her audience, Professor Heilbrun embeds literary allusions throughout these rites of passage—Macbeth, Glendower, Hotspur, Derrida, Antigone, Austin, Shakespeare, Jonson, even Kilmer, in just the first couple of pages.

Like the rest of us, she started word processing because she was impressed by its speed and efficiency but then discovered some higher value in it, something mysterious, difficult to articulate, but very nearly a universal experience among thoughtful users of word processing. "Oddly enough," she writes, "the fears expressed by those who infer the impersonality of the product from the impersonality of the machine that produces it have obscured the fact that the person composing on a computer is more in touch with the work than is the person who uses a secretary or typist." More in touch with the work? Just so—an exciting and as yet little explored benefit of composing with a word processor.

In "The Word Processor and the Work Habits of the Humanist," McKenzie reviews the general features of a typical word-processing program, including explanations of concepts that underlie terms like *cursor*, *word wrap*, *scrolling*, *boilerplate*, *menu*, and *buffer*. Again, the experience of word processing follows a
predictable plot: "Within a month exhilaration will set in, followed shortly by absolute dependence. Many users grow positively uneasy after three consecutive days away from the terminal." And when obstacles arise, they have a literary point of reference: Professor McKenzie's recalcitrant printer is known to its frustrated users as "Bartleby," a reference that may even illuminate Melville's text. I wonder, though, about McKenzie's overall assessment of word processing: "It is the ease with which a word processor accumulates accuracy that makes it most valuable to the humanist." Surely, humanists value more interesting things than ease and accuracy. To my mind, Heilbrun's "more in touch with the work" comes closer to identifying the humanist's potential interest in word processing.

Of course, ease and accuracy are quite properly the chief concerns of the next two chapters, also by McKenzie: "The Word Processor and the Typing Chores of the Academic Department" and "The Various Systems—Some Considerations and Costs." The cyclical repetitiveness of documents generated by academic departments certainly calls for computerized applications. For most of us, administrative uses of word processing have preceded our own, since the benefits in the former are more easily quantified. Anyone who does not already have a word-processed department office will certainly benefit from the guidance and examples offered here. Many administrative routines, moreover, do not require any specialized software other than a good word-processing program. McKenzie provides a most useful guide with checklists to assist departments and individuals in selecting equipment and software. Inevitably, however, in the rapidly evolving world of personal computing, such discussions become dated even before they reach print. For example: "Whatever microcomputer you buy, you will need more help setting it up than the manuals supplied by the manufacturer will offer." Maybe so, but such an admonition, probably written in 1982 or 1983, sounds quaint in 1987. Today, a Macintosh computer can be removed from its carton and put into operation by a chimpanzee in 45 seconds.

The authors of the two final chapters review some scholarly uses of word processing. In "The Word Processor and Scholarly Editions," Charles T. Cullen demonstrates the advantages in speed, efficiency, and accuracy that were achieved by Princeton University Press in its Writings of Thoreau series. Whereas an earlier volume had taken a year to go from copy to galleys, a more recent word-processed volume took only a week—a remarkable achievement, to be sure. Cullen points out that word processing brings the roles of editor and publisher closer together, a most interesting observation, and one that reminds us that the methods of publication we are accustomed to grew out of the nature of print technologies. In contrast, word processing requires only one version of a text—not a text laboriously recopied at each stage of production, with the chance of fresh errors being introduced at each stage. In "The Word Processor and the Scholarly Journal," Mark P. Haselkorn and Jim S. Borck discuss the use of word processing as a matter of economical survival. Again, the need for just one version of a text, held in electronic storage, makes possible impressive improvements in cost-effectiveness, speed, and accuracy.

This slim volume provides a useful and well-written introduction to academic word processing. The contributions of Professor McKenzie in particular are written in a sprightly style that delights as well as it informs. My one reservation is with the vaguely precious way in which the word "humanist" is used in both title and text. I am not sure I see how the needs of "the academic humanist" for word processing are any different from those of other academics. According to one of the essays, the ease and accuracy of word processing makes it "most valuable to the humanist." Why "humanist"? I also feel uneasy about the image of the "humanist" that emerges from this volume, for the humanist here is a person who writes scholarly articles, does scholarly editions, edits scholarly journals, and administers academic departments. While this emphasis is not unexpected in a publication from the Modern Language Association, I am disappointed to see nothing here about the humanist's role as teacher, writer, communicator, and thinker. A better subtitle would have been "Word Processing for the Academic Scholar."

Another collection of essays emphasizes pedagogical rather than scholarly classroom applications. The Computer in Composition Instruction: A Writer's Tool,
edited by William Wresch for the National Council of Teachers of English, offers a summary of the early gropings into this new and promising frontier. The authors of the essays are both teachers and software developers who describe the origins and aims of their programs as well as the process of creating, testing, and integrating them into writing curricula. The essays are divided into four groups: prewriting approaches, editing and grammar programs, word-processing research and applications, and programs for the writing process. In addition, Bruce C. Appleby has provided a helpful 57-item annotated bibliography.

In "Recollections of First-Generation Computer-Assisted Prewriting," Hugh Burns, a genuine pioneer in computer applications to writing, provides examples from his INVENT program that demonstrate how interactive computing can open new territories of student writers' minds and imaginations. Since the "processes of rhetorical invention are basically actions and reactions," he writes, "the computer acts to create the tensions, and the writer reacts to create the intentions." He offers some illustrations of ways to overcome the problems of writing, such as "open program"—a program, that is, in which there are no right or wrong responses. His five recommendations for the second generation of invention programs are authoritative enough to warrant summarizing: (1) account for a writer's "natural recursive habits," (2) allow teachers and users to modify and shape programs to their individual needs, (3) make the program respond more specifically to student inputs, (4) increase the use of color, graphics, and sound to create "media-enriched Socratic dialogues," and (5) match the program to the cognitive style of the user.

Dawn Rodrigues describes a more specific kind of programmed invention in "Computer-Based Creative Problem Solving." The aim of this program is to encourage and guide creative problem-solving as an alternative, or perhaps as a supplement, to the systematic inquiry of formal heuristics. One method uses "visual synetics"—a computer-guided set of comparisons between a topic and some set of objects. By using the power of analogy and metaphor to stretch thought, the program extracts analogies and then encourages the user to explain and elaborate. Another approach uses "subtopic matrixes," in which the program prompts students to elaborate on the connections between items on a list generated through free-writing. The program appears to make good use of the computer's capacity to manipulate information and to provide alternative visual displays.

Another prewriting program aims to help students generate content for a particular kind of academic paper—the type that states and supports a thesis. In "SEEN: A Tutorial and User Network for Hypothesis Testing," Helen J. Schwartz presents examples from an interesting case study of three students who use SEEN in order to analyze the character, Don Quixote. The tutorial part of the program interactively guides the formation of thesis statements and support. The network part of the program allows the students to respond to each other's ideas and to test the validity of their support on an on-line audience. Finally, in the textfile part of the program, the students develop a draft from the language generated in the tutorial and on the network. Although the printed record of these interactions looks like good class discussion, Schwartz is certainly right when she says that the computer interactions are better than discussion because every participant is forced to produce written language. At the same time, an open program such as this one is "a vehicle for exploration, discovery, and intellectually serious playfulness."

Departing from such intellectually serious playfulness are the three editing and grammar programs described in this collection. The creators of open programs have a much easier time than their colleagues who attempt to write more complicated programs for checking texts against specified standards. Perhaps the most venerable of these text-checking programs is described by Kathleen Kiefer and Charles R. Smith in "Improving Students' Revising and Editing: The Writer's Workbench System." Anyone who has used a spelling-check program will appreciate a program that instantly discovers usage and mechanical errors. But programs like WRITER'S WORKBENCH do not easily distinguish between editing and revising. Consider, for example, this advice: "Since your introduction is considerably shorter than average, please check to be sure it introduces the essay gracefully." Does the student understand what a graceful introduction is? How does...
WRITER'S WORKBENCH know that the introduction's relative brevity means that it lacks grace? Or this: "In this text, 2.8 percent of the words are abstract. Psychological research shows that concrete texts are easier to read, easier to use, and easier to remember. Generally, the lower the abstract index, the better. Your percentage of abstract words given above, however, is higher than the usual limit, 2%. Have you illustrated each of your points with fully developed examples and specific details? If not, do so before handing your paper in." One wonders if it is as easy to get a normal mix of abstract and concrete words as it should be to revise this last sentence so that it does not end with a preposition. Why is it "percent" in one place and "%" in another? How does WORKBENCH infer faulty illustration from a slight excess of abstraction? But even setting aside these quibbles, what kind of philosophy of composition underlies such directives? WORKBENCH can help writers who make a lot of small surface errors, and it can help advanced writers, and even professional writers, to be more thoughtful about style and grace, but in its present stage of development I wonder how helpful it is for the great mass of average student writers. Of course, Kiefer and Smith are experienced teachers who do have a philosophy of composition. I point out what I think are defects in WRITER'S WORKBENCH only to illustrate the enormous task facing programmers who must treat the rhetorical features of a text in the same way that they treat its mechanical features.

In "HOMER: Teaching Style with a Microcomputer," Michael E. Cohen and Richard A. Lanham describe a program that approaches style more narrowly, and thus more successfully. Based on Lanham's text, Revising Prose, this program assumes that student writing is characterized by the same stylistic faults as bureaucratic writing. Accordingly, it checks a text for overuse of prepositions and forms of "to be," for "shun" words (those ending in -sion or -tion), and for "woolly" words. A line-by-line map provides the writer an opportunity to locate and revise these particular kinds of lapses. Another helpful feature is a visual display of the relative length of sentences. These displays, together with statistical data, help a thoughtful writer understand how to polish sentences for an audience. Unfortunately, the audience in the machine has a rather smarmy voice: "GOOD LUCK, Anonymous!!"; "OH! MY! All these "TO BE" VERBS distress me!!"; "You keep your PREPOSITIONS under control—how admirable!" One wonders what prompts the machine to respond with one, two, or three exclamation points. The creators of the pioneering programs were intensely concerned about personalizing these electronic procedures, but as we become more accustomed to using computers for a variety of purposes, I look forward to these bogus familiarities disappearing. Like the other authors in this collection, Cohen and Lanham are careful to remind us that these programs help some of the students some of the time—an observation wise in its modesty.

In "The COMP-LAB Writing Modules: Computer-Assisted Grammar Instruction," Michael G. Southwell describes a program that attempts to teach basic elements of grammar through direct interactive instruction and "autotutorials" rather than by pointing out errors in a student-generated text. This program's more positive approach makes it more appropriate for basic writers, whose learning of standard written English, the assumption is, resembles learning a second language. The intuitive response of most teachers reading through the examples of programmed instruction and interactive exercises here will be that this is little more than the drill-and-practice pedagogy to which basic writing students have been so steadfastly resistant. Still, Southwell has accumulated as yet unpublished evidence that students achieve better results performing these exercises in the lab than they do in the classroom. If so, this program is a step in the right direction.

The three essays in Part III deal with word processing research and applications. In "Integrating Computers into a Writing Curriculum; or, Buying, Begging, and Building," Lillian S. Bridwell and Donald Ross offer a narrative about the tribulations and triumphs of setting up an upper-division writing program taught by graduate students and employing word processing. One of their useful discoveries is that the instructions in word-processing manuals should be rewritten for the diverse needs of student writers. In addition, the authors have experimented with data-
based adjuncts to word processing such as text analysis and generic formats, as well as interventions in the writing process through guided invention, conferencing, and reader responses.

Stephen Marcus's McLuhanesque probes into the nature of the new technology provide some of the best reading in this anthology. In "Reading-Time Gadgets with Feedback: Special Effects in Computer-Assisted Writing," he explores the special features of "videotext"—writing on a screen that has the "quality of light." By toning down the light of the video display, students can detach themselves from the outpouring of their texts, a method of "invisible writing" that has already been widely adopted. The fluidity of videotext enables a writer to move around inside a text that expands and contracts to accommodate additions and deletions. In a program called SCREEN SCENES, the writer fills a gap between seemingly unrelated sentences, an exercise in coherence made possible by the fluid character of videotext. The wires connecting computers with video displays can be rearranged to enable a student to have a video dialogue with a partner, enhancing the collegial and collaborative nature of writing and relieving the sense of isolation that some people feel when they work with computers. By using a program like COMPUTPOEM, students can explore the nature of authorship and how poems work syntactically. Although it was once thought that television had undermined literacy, Marcus quite correctly observes that the need to interpret visual codes on a computer display may demand greater literacy. And, in any case, working with videotext is "intrinsically motivating," which may explain why even ordinary drill-and-practice programs that fail in classrooms will succeed in computer labs.

Where Stephen Marcus probes the nature of writing in video through close observation of human behavior, Colette Dalute probes by means of controlled study of children's use of word processing. In her essay "Can the Computer Stimulate Writers' Inner Dialogues?" she reviews her research study, including details about developing and testing the word-processing program CATCH and evaluating the results through statistics and case studies. In general she found that the program succeeded in its main goal—to increase students' ability for self-monitoring and decision-making in their writing, leading to more self-generated correction and revision. She provides more details about this research in her recent book Writing and Computers (Reading, MA: Addison-Wesley, 1985).

The final essays in this collection focus on programs designed to assist the entire writing process by integrating several smaller programs. William Wresch, for example, has brought prewriting options, a word processor, and a test analyzer together in WRITER'S HELPER. Since these programs are integrated, a student is able to move easily from one type of activity to another. As a skillful programmer, Wresch has been able to localize his program so that it suits his students' needs and extends his own teaching style. Although such individualized programs are necessarily eclectic and eccentric, the opportunity to create them is an exciting prospect for teachers. Of course, its homegrown character also makes WRITER'S HELPER of limited value outside of its home. For example, teachers who prefer that writing assignments arise from situations would object to a prewriting program built around the rhetorical idea of "find a subject."

In contrast to the individualized, intuitive, and local features of WRITER'S HELPER are the research-based, grant-supported, collaborative features of "WANDAH: Writing-Aid AND Author's Helper," described by Ruth Von Blum and Michael E. Cohen. (The wretched name and acronym of this program have mercifully disappeared in its commercial version—HBJ Writer.) Perhaps the most distinctive features of the development of WANDAH are its genesis in basic research in the cognitive processes of writing, applied research on computer intervention, and the collaborative efforts between experienced writing teachers and computer programmers. In order to remove some of the overload that often hinders students, the collaborators have divided simultaneous writing tasks into discrete segments of the program. Thus, students can choose from among many options as they move through prewriting, word processing, and reviewing. Whether or not one likes WANDAH, the record of its creation should stand as an example of research-based problem-solving applied effectively to software development.
In contrast to WANDAH's whole process approach, Cynthia L. Selfe's WORDSWORTH II is a supplementary integrated program whose eight modules focus on eight kinds of writing assignments based on such current-traditional categories as narration, description, persuasion, and so on. The program offers the student assistance in a module by providing examples, reviews of classroom material, prewriting strategies appropriate for the assignment, and various guides to revising and editing. It also provides both the student and teacher with a written record of all transactions. Its rather rigid distinctions between "planning" and "polishing" and between early, middle, and final drafts are modified somewhat by the supplementary nature of the program. Like WANDAH's development, that of WORDSWORTH II reveals exemplary collaboration between teachers and software designers.

Christine Neuwirth's program, DRAFT, is aptly called a "computer-based writing environment," for it includes options that enable the student to do almost anything while writing or revising. Heuristic guidelines, organizational patterns, illustrations, analyses of drafts, and all sorts of tutorials are on-line at the fingertips of any student who feels comfortable with the relatively complex computing required to run the program. No doubt the students at Carnegie-Mellon, where DRAFT is under development, are able to manage the computing well enough to shape the program to their individual needs. In exchange for this complexity, the student gets on-line access to a great deal of information and guidance as well as immediate feedback on the emerging text. The instructor gets an opportunity to modify the program in order to accommodate special needs and is able to use a printed record of interactions for diagnostic and research purposes.

All of these programs reveal the efforts of thoughtful and talented teachers of writing to discover how technology can help them do what they know needs to be done. More experience and further development will probably help us figure out where the computer is truly helpful, where its apparent helpfulness depends on novelty, and where it is a waste of time and energy. If I had to make a prediction, I would say that the programs for guiding invention or particular types of writing assignments, or those that involve massive, networked data bases or drill-and-practice exercises have less probability of enduring than uncomplicated and flexible word-processing programs capable of easy interaction with reliable text analyzers and networked responses.


Reviewed by Richard B. Larsen, Francis Marion College

Helen Schwartz's *Interactive Writing: Composing with a Word Processor* is fully as much as one could hope for in the first course-usable textbook of its type from a practicing teacher/researcher (who has published on CAI and designed her own courseware). When I say course-usable I have in mind options that enable the student to do almost anything while writing or revising. Heuristic guidelines, organizational patterns, illustrations, analyses of drafts, and all sorts of tutorials are on-line at the fingertips of any student who feels comfortable with the relatively complex computing required to run the program. No doubt the students at Carnegie-Mellon, where DRAFT is under development, are able to manage the computing well enough to shape the program to their individual needs. In exchange for this complexity, the student gets on-line access to a great deal of information and guidance as well as immediate feedback on the emerging text. The instructor gets an opportunity to modify the program in order to accommodate special needs and is able to use a printed record of interactions for diagnostic and research purposes.

All of these programs reveal the efforts of thoughtful and talented teachers of writing to discover how technology can help them do what they know needs to be done. More experience and further development will probably help us figure out where the computer is truly helpful, where its apparent helpfulness depends on novelty, and where it is a waste of time and energy. If I had to make a prediction, I would say that the programs for guiding invention or particular types of writing assignments, or those that involve massive, networked data bases or drill-and-practice exercises have less probability of enduring than uncomplicated and flexible word-processing programs capable of easy interaction with reliable text analyzers and networked responses.


Reviewed by Richard B. Larsen, Francis Marion College

Helen Schwartz's *Interactive Writing: Composing with a Word Processor* is fully as much as one could hope for in the first course-usable textbook of its type from a practicing teacher/researcher (who has published on CAI and designed her own courseware). When I say course-usable I have in mind options that enable the student to do almost anything while writing or revising. Heuristic guidelines, organizational patterns, illustrations, analyses of drafts, and all sorts of tutorials are on-line at the fingertips of any student who feels comfortable with the relatively complex computing required to run the program. No doubt the students at Carnegie-Mellon, where DRAFT is under development, are able to manage the computing well enough to shape the program to their individual needs. In exchange for this complexity, the student gets on-line access to a great deal of information and guidance as well as immediate feedback on the emerging text. The instructor gets an opportunity to modify the program in order to accommodate special needs and is able to use a printed record of interactions for diagnostic and research purposes.

All of these programs reveal the efforts of thoughtful and talented teachers of writing to discover how technology can help them do what they know needs to be done. More experience and further development will probably help us figure out where the computer is truly helpful, where its apparent helpfulness depends on novelty, and where it is a waste of time and energy. If I had to make a prediction, I would say that the programs for guiding invention or particular types of writing assignments, or those that involve massive, networked data bases or drill-and-practice exercises have less probability of enduring than uncomplicated and flexible word-processing programs capable of easy interaction with reliable text analyzers and networked responses.


Reviewed by Richard B. Larsen, Francis Marion College

Helen Schwartz's *Interactive Writing: Composing with a Word Processor* is fully as much as one could hope for in the first course-usable textbook of its type from a practicing teacher/researcher (who has published on CAI and designed her own courseware). When I say course-usable I have in mind options that enable the student to do almost anything while writing or revising. Heuristic guidelines, organizational patterns, illustrations, analyses of drafts, and all sorts of tutorials are on-line at the fingertips of any student who feels comfortable with the relatively complex computing required to run the program. No doubt the students at Carnegie-Mellon, where DRAFT is under development, are able to manage the computing well enough to shape the program to their individual needs. In exchange for this complexity, the student gets on-line access to a great deal of information and guidance as well as immediate feedback on the emerging text. The instructor gets an opportunity to modify the program in order to accommodate special needs and is able to use a printed record of interactions for diagnostic and research purposes.

All of these programs reveal the efforts of thoughtful and talented teachers of writing to discover how technology can help them do what they know needs to be done. More experience and further development will probably help us figure out where the computer is truly helpful, where its apparent helpfulness depends on novelty, and where it is a waste of time and energy. If I had to make a prediction, I would say that the programs for guiding invention or particular types of writing assignments, or those that involve massive, networked data bases or drill-and-practice exercises have less probability of enduring than uncomplicated and flexible word-processing programs capable of easy interaction with reliable text analyzers and networked responses.