Bibliography Update

Bradford A. Morgan

Adobe Systems. *PostScript Language Tutorial and Cookbook*. Reading, MA: Addison-Wesley, 1986.

Adobe Systems. *PostScript Language Reference Manual*. Reading, MA: Addison-Wesley, 1986.

Alaska Department of Education. *Hand in Hand: The Writing Process and the Microcomputer*. Anchorage, AK: Alaska Department of Education, 1986.

Amsterdam, Jonathan. "Augmented Transition Networks for Natural Language Parsing: These Descendants of RTNs Will Construct Parsers for Front-end Subsets of English and Can Also Be Compiled into LISP." *AI Expert*.1:1(1986), pp. 15-21.

Anson, Chris M. "A Computerized List of Journals Publishing Articles in Composition." *College Composition and Communication*. 37:2 (May 1986), pp. 154-166.

Auten, Anne. "The Viewing-Writing Process." *The Computing Teacher*. 13:9 (June 1986), pp. 14-16. (computers, VCRs, and composition)

Barker, Thomas T. "Guidelines for Opening a Multipurpose Microlab in an English Department." *Collegiate Microcomputer*. 4:2 (Summer 1986), pp. 115-122.

Beinhorn, George R. "WordStar Tips, Tricks, and Traps: These Little-Known Commands Can Save You Time and Trouble." *IBM PC Update*. 3:8 (August 1986), pp. 16-19.

Biancalana, Michael J. "Fancy Font Works: II Computing's Font Editor and Typer." *II Computing*. 1:5 (June/July 1986), pp. 27-29.

Bostwick, Steve. "Digital Scanners Build Font Libraries: Versatile and Inexpensive Image-Capture Peripherals Use Sources As Varied As Novels and Calligraphy To Create Large Font Libraries for Laser Printers." *Mini-MicroSystems*. 19:8 (June 1986), pp. 149-157.

Bove, Tony; Cheryl Rhodes, and Wes Thomas. *The Art of Desktop Publishing*. New York: Bantam Books, 1986.

Bright, David. "Voice-Activated Writer's Block." *Computerworld*. 20:30 (July 28, 1986), pp. 23, 28.

Bright, David. "Nestor Software Translates Handwriting to ASCII Code." *Computerworld*. 20:30 (July 28, 1986), pp. 23, 26.

Cavuoto, James. "Page-Description Language Increases Power, Sophistication of Laser Printers." *PC Week*. 3:23 (June 10, 1986), pp. 101-104.

————. *LaserWrite It!* Reading, MA: Addison-Wesley, 1986

Chauncey, Caroline. "The Art of Typography in the Information Age: Like Everything Else, Type Design Has Gone Digital. Poor Design Makes Many Digital Typefaces Difficult to Read, But the Ancient Art of Typography Is Coming to the Rescue." *Technology Review*. 89 (February-March 1986), pp. 26-31, 79.

Crider, Janet. "Sophisticated Writing with WordPerfect's Thesaurus: If You're Searching for the Perfect Word, This Program Can Help." *IBM PC Update*. 3:6 (June 1986), pp. 9-13.

Davis, Frederic E. et al. *Desktop Publishing*. Homewood, IL: Dow Jones-Irwin, 1986.

"Dedicated Word Processors." (Comparative chart). *Today's Office*. 20:12 (May 1986), pp. 51-52, 56, 59-60.

Doszkocs, Tamas E. "Natural Language Processing in Information Retrieval." *Journal of the American Society for Information Science*. 37:4 (July 1986), pp. 191-196.

Eble, Kenneth E. "Word Processors: Zucchini in the Academic Garden." *Chronicle of Higher Education*. 32:20 (July 16, 1986), p. 41.

Epstein, Jonathan A. "Best Font Forward: A Small, VAX-based Design Studio Produces Typefaces for the Present and the Future." *Digital Review*. 3:10 (July 1986), pp. 82-88.

Etzioni, Oren. "Why Computers Don't Understand English—Yet: But Considerable Progress Is Being Made." *Lotus*. 2:6 (June 1986), pp. 10-11.

Felici, James and Ted Nace. "Typesetting Point by Point: The Typographer's Language Is Peppered with Terms Dating Back to Gutenberg's Time. While Word Processing Has Acquainted You with the Basic Vocabulary, You'll Need To Be Fluent To Make the Jump to Personal Computer Publishing." *PC World*. 4:7 (July 1986),pp. 170-181.

Frederiksen, Mark J. "Programming Rids Word Processors of English Bias." *Government Computer News*. 5 (January 31, 1986), p. 34.

Geist, Ronni and Harry. "Printing with Style: A Look at Three Personal Publishing Systems That Take Plain-Vanilla Text to New Heights with Eye-catching Type Styles and Other Graphics Elements." *PC Products*. 3:7 (July 1986),pp. 25-33.

Georgas, Nora. "'Books in Print' Index Available on CD ROM." *PC Week*. 3:27 (July 8, 1986), p. 38.

Glitman, Russell. "Desktop Publishing Poised for Dramatic Growth." *PC Week*. 3:22 (June 3, 1986), pp. 111-115.

————. "Text-Input Systems Key to Growth of Desktop Publishing." *PC Week*. 3:22 (June 3, 1986), p. 118.

————. "PC Desktop Publishing To Get a Mac-Like Look." *PC Week*. 3:22 (June 3, 1986), p. 119.

————. "Standards Are Needed for Desktop Publishing." *PC Week*. 3:22 (June 3, 1986), pp. 120-122.

Goss, Donald R. "Remote Typesetting: Eliminating Steps in Your Printing Project." *The DEC Professional*. 5:6 (June 1986), pp. 52-54.

Greenberg, Mark. "Whence the Process of Words?" *Collegiate Microcomputer*. 4:2 (Summer 1986), p. 101. (William Blake perspective)

Guthrie, Anne E. "Using Word's Windows and Glossaries: With Two of Microsoft Word's Finest Features, You Can Increase Your Productivity and Avoid Data Redundancy." *IBM PC Update*. 3:8 (August 1986), pp. 9-12.

Herring, Sherry Lee. "Page Description Languages Help Users Merge Text, Graphics." *Computerworld*. 20:16 (April 21, 1986), p. 49.

Hertz, Robert M. *Computers in the Language Classroom*. Menlo Park, CA: Addison-Wesley, 1987.

Hizer, Mark. "Putting WordStar on the Right Path: Learn How to Create WordStar Files in Other Directories through the Clever Use of DOS or Third-Party Utilities." *IBM PC Update*. 3:5 (May 1986), pp. 17-19.

————. "Learn To Be a WordStar Troubleshooter: Don't Get Mad. Get Even—wtih These Solutions To Some Common WordStar Problems." *IBM PC Update*. 3:6 (June 1986), pp. 13-15.

Hockey, Susan. *SNOBOL Programming for the Humanities: An Introduction to Computer Programming for Non-Scientific Applications*. New York: Oxford University Press, 1985.

Holmes, Leigh. "Individualizing Composition Instruction Through CAI Commentary Programs." *Focus*. (Southeastern Ohio Council of Teachers of English) 12:2 (Winter 1986), pp. 39-42.

Holtz, Matthew. *Mastering MS Word*. Berkeley, CA: Sybex, 1986.

Jackson, Denise V. "Computer Assisted Research Writing." *NCTE-ACE Newsletter*. 1:3 (January-March 1986), p. 22.

Janus, Susan. "Spell It Write/Right: The Latest in Orthographic Aids for Apple II Word Processing." *A+: The Independent Guide to Apple Computing*. 4:8 (August 1986), pp. 49-61.

Kalita, Jugal K. et al. "Summarizing Natural Language Database Responses." *Computational Linguistics*. 12:2 (April-June 1986), pp. 107-124.

Kenner, Hugh. "RACTER: Artificial Intelligence or Packaged Lunacy?" *Byte*. 11:5 (May 1986), pp. 289-291.

Knezek, Don. "Periodicals in the Computer Literacy Classroom." *The Computing Teacher*. 13:9 (June 1986), pp.52-54.

Knorr, Eric and Robert Luhn. "Personal Publishing in Black and White: Personal Publishing Offers PC Users Unbridled Opportunities—and More Than a Few Complexities. While the Concepts and the Technologies Are Accessible, Publishing on the PC Requires Dedication and a Lot of Sweat." *PC World*. 4:7 (July 1986), pp.182-188.

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

————. *TEX: The Program*. Reading, MA: Addison-Wesley, 1986.

————. *The METAFONT Book*. Reading, MA: Addison-Wesley, 1986.

————. *METAFONT: The Program*. Reading, MA: Addison-Wesley, 1986.

————. *Computer Modern Typefaces*. Reading, MA: Addison-Wesley, 1986.

Lamport, Leslie. *LATEX: A Document Preparation System*. Reading, MA: Addison-Wesley, 1986.

Mau, Ernest E. "Picking a Spelling Checker That Won't Let You Down: Use This Objective Checklist To Help You in Your Search for the Infallible Proofreader." *IBM PC Update*. 3:7 (July 1986), pp. 7-13.

"Menu-Driven Word-Processing Packages." *PC Week*. 3:22 (June 3, 1986), pp. 76-84. (Comparative review)

Miller, Alan R. "LETTRIX: Almost-Letter-Quality on Your Dot-Matrix Printer." *Byte*. 11:5 (May 1986), pp. 299-304.

Miller, James C. "Giving WordStar Reports the Graphic Edge: Now You Can Use Keyboard Characters To Create Graphics and Jazz Up Your Reports and Presentations." *IBM PC Update*. 3:8 (August 1986), pp.12-16.

Milone, Michael N., Jr. "Desktop Publishing for Schools?" *Classroom Computer Learning*. 6:8 (May 1986), p. 63.

Moses, Albert L. "How Your Computer Can Improve Your Writing, Part Two." *The Lawyer's PC*. 3:18 (May 15, 1986), pp. 1-5.

Mullins, Carolyn J. "Software Evaluation by the Manual: In Which It Is Argued That You Can Judge a Book by Its Cover." *Information Center*. 2:5 (May 1986), pp. 38-39, 42-44.

————. "Add-ons for Word Processing Programs Enhance your Words." *Cause/Effect*. 9 (January 1986), pp. 36-40.

Nace, Ted and Michael Gardner. *LaserJet Unlimited*. Berkeley, CA: Peachpit Press, 1986.

Neil, J. P. "Apple Writer II Word Processing Language: What the Manual Doesn't Tell You." *A+: The Independent Guide to Apple Computing*. 4:8 (August 1986), pp. 45-48.

Northern, Jerry L., ed. *The Personal Computer for Speech, Language, and Hearing Professionals*. Boston: Little, Brown, 1986.

Norton, Peter. "What's Stopping PCs from Catching Macs in Publishing." *PC Week*. 3:22 (June 3, 1986), p. 37.

O'Malley, Christopher. "Write Faster, Speedier Ways To get the Word Out: Exploiting Underutilized Features in a Word Processing Program Cuts Down the Keystrokes and Helps Get Your Job Done More Quickly." *Personal Computing*. 10:7 (July 1986), pp. 71-77.

Pallatto, John. "Word Processing by Menu: Many Users Prefer Easy-to-learn, English-speaking Alternatives to Command-driven Packages." *PC Week*. 3:22 (June 3, 1986), pp. 65-67, 72.

Pepper, Jon. "Choosing a Word-Processing Upgrade: First Ask Whether It's Worth Displacing Your Current Standard." *PC Week*. 3:22 (June 3, 1986), pp. 59-60.

Perryman, Polly. "Standards and Documentation: Writing the First Word Is the Hardest Part of Producing a Manual, But Here Is a Way To Get a Head Start."*Information Center*. 2:8 (August 1986), pp. 31, 34-37.

Porter, Bernajean. "Word Builders: Ten Ways to Use the Word Processor to Exercise Those Vocabulary Muscles." *Teaching and Computers*. 3:6 (March 1986), pp. 21-24.

Price, Anne Jamieson. *A Survey of Computer Use in Projects Sponsored by the National Endowment for the Humanities*.Washington, DC: Office of Scholarly Communication and Technology, 1986, 36 pp.

Rice, William. "The Effects of Database Searches on the Selection of Research Paper Topics." *Collegiate Microcomputer*. 4:2 (Summer 1986), p. 109.

Ritvo, Ken and Greg Kearsley. *Desktop Publishing*. La Jolla, CA: Park Row Press, 1986.

Robertson, Barbara. "Corporate Publishing Takes Off: Pioneering Companies Large and Small Are Reaping Enormous Savings and Dramatically Improving the Quality of Their Public Image by Assigning a New Function to Their PCs—Publishing." *PC World*. 4:7 (July 1986), pp. 190-197.

Rosenthal, Steve. "Apple II Desktop Publishing: You Can Use Your Apple II-Series Computer to Unleash the Powers of the Apple LaserWriter." *A+: The Independent Guide to Apple Computing*. 4:7 (July 1986), pp. 22-29.

Scisco, Peter. "Desktop Publishing Isn't Just for Apple Anymore." *Micro Marketworld*. 9:8 (April 14, 1986), pp. 43-45.

Selfridge, Mallory. "Integrated Processing Produces Robust Understanding." *Computational Linguistics*. 12:2 (April-June 1986), pp. 89-106.

Silverstein, Diane. "Computers and the Language Arts: Creating Software for Classroom-Specific Needs." *The Computing Teacher*. 13:8 (May 1986), pp. 12-14.

Stern, Marc. "From Mind to Manuscript: Putting Words on Papers Is Easy. The Hard Part Is Organizing and Effectively Assembling Text, and This Is Where Outline Processors Make Their Mark." *PC Products*. 3:7 (July 1986), pp. 37-43.

Sullivan, Kristina B. "Network Versions of Word-Processing Software Have Advantages, Drawbacks." *PC Week*. 3:27 (July 8, 1986), pp. 147-149.

Teichman, Milton. "What College Freshmen Say about Word Processing." *Perspectives in Computing*. 5:3/4. (Fall/Winter 1985), pp. 43-48.

Tenopir, Carol. "Student Online Data Base Searching, Part II." *The Computing Teacher*. 13:8 (May 1986), pp. 39-40, 57.

Thomas, Jean. "Creating Fictional Scenes on the Word Processor." *NCTE-ACE Newsletter*. 1:3 (January-March 1986), pp. 18-19.

Thompson, Keith. "Pagemaker Remains Chief Page Composer." *Infoworld*. 8:23 (June 9, 1986), pp. 39-40.

Ulick, Terry. *Personal Publishing with the Macintosh*. Hasbrouch Heights, NJ: Hayden, 1986.

Watt, Peggy. "Pros Wary of 'Desktop Publishing': Initial Acceptance Could Result in Bad Graphics Flow." *Computerworld*. 20:20 (May 19, 1986), pp. 33-43.

————. "Publish or Perish: Start-Up Grabs Early Page Language Lead." *Computerworld*. 20:16 (April 21, 1986), pp. 1, 14.

Weiskel, Timothy. "Bibliographic Research and the Love of Learning." *Perspectives in Computing*. 5:3/4. (Fall/Winter 1985), pp. 13-21.

Whisenand, Susan. "Beyond Merlin and His Magic Staff." (FrEdWriter-oriented excerpts). *The Computing Teacher*. 13:8 (May 1986), pp. 8, 32, 57.

Wilcox, Lance and Craig Hansen. "Microcomputers and the Writing Instructor." *Collegiate Microcomputer*. 4:2 (Summer 1986), pp. 123-130.

Williamson, Mickey. "In-House Publishing: DP Can Prosper by Tapping Print Systems' Potential." *Computerworld*. 20:21 (May 26, 1986), pp. 49-50, 55-60.

Wolf, Dennie and Joseph Walters. "The Writer with a Word Processor: Juggler, Sculptor, Director." *Computers in the Schools*. 3:1 (Spring 1986), pp. 35-46.

THE PROFESSIONAL WRITER'S WORKSTATION

Bryan Pfaffenberger

Electronic Outlining Comes of Age

"Always create an outline before writing," or so goes the admonition heard by generations of writing students. To create an outline is to create what the late E.B. White calls a "suitable design," a structure that serves the writer's aims. A prelude to writing, this indispensable planning step is particularly useful when composing lengthy documents in the computer environment, where the 24-line text display limits a writer's sense of a document's overall structure. A good outline provides what most word-processing programs cannot: a roadmap to the big picture of a document's organization.

Outlines can be created on the backs of old envelopes or restaurant placemats (both of which are, in my experience, particularly fertile environments for planning.) And now, thanks to the efforts of clever programmers, they can be created on computers. Outlining programs such as *ThinkTank* provide professional writers with the tools needed for creating, restructuring, and printing an outline before writing. A writer can "brainstorm," rapidly listing a set of ideas, and then use the outlining program's text-moving commands to organize them. To see the plan's overall structure, subordinate headings can be collapsed or hidden so that only the major headings show on the screen. A "show" or "expand" command quickly reveals the hidden headings so that details can be added.

Outlining programs are fun to use and, doubtless, do for outlining what word processing does for writing: they reduce, quite dramatically, the tedium and paperwork that would be involved in old-fashioned, pencil-and-paper revision. And yet, in one sense, first-generation outlining programs such as *ThinkTank* are as conventional as an outline scribbled on notebook paper. They erect barriers to the revision of the outline as the writing plan changes during the document's composition. And changes in the plan will occur. As research on the composition process has demonstrated, professional writers cycle back and forth among planning, writing, and revision activities as they create a document. Writing, in short, is a discovery process, one in which the writer's sense of the document's overall structure is virtually certain to change. If the outline is to serve as an accurate roadmap of that structure, it must be updated as changes occur. And this is precisely where first-generation outlining programs fall down.

All outlining programs, to be sure, facilitate an outline's revision; most include text insertion, deletion, and moving commands that rival those of the best word-processing software. Yet, because today's personal computers can execute only one program at a time, these tools are available only after going through a process so tedious that few writers will update their computer-generated outlines. Consider: to make an update, you'll have to save your document to disk, exit the word-processing program, load the outlining program, update the outline, save the updated outline, print the updated outline, exit the outlining program, load the word-processing program, scroll to the place you left off, and resume working.

Today's 8088-based PCs can't run two programs at once all that effectively, but some clever programming has made it far easier to switch from one program to another (and back again). Many personal computers have 512K or more of RAM installed, but most popular word-processing programs (such as *WordStar* or *Microsoft Word)* were designed back in the days when few computers were equipped with more than 128K or 256K. Much of a 512K or 640K computer's memory is, therefore, unused. There's room in this underutilized memory for a supplementary program, such as a spelling checker, an appointment calendar, or an outlining program. At a keystroke, the supplementary program—called a "memory-resident" program in computer jargon—can leap onto the screen, suspending a word-processing program's execution and allowing the user to make immediate use of whatever tools the supplementary program provides.

Memory-resident outlining programs, such as Living Videotext's *Ready!* and SoftWorks Development's *PC-OUTLINE*, reduce the tedium of updating outlines considerably. Suppose you're writing a business report and, in the middle of the introduction, you realize that you need a separate chapter to talk about the research methodology. At a keystroke, the outline leaps into view on the screen. You make the changes and save them. Another keystroke returns you to word-processing program, positioning the cursor at precisely the same spot you were at when you left the program to update the outline.

Ready!, the first memory-resident outlining program, comes from the same people who created the first outlining program, *ThinkTank*. Not surprisingly, it resembles *ThinkTank* so closely that *ThinkTank* users will be able to use the program immediately. Because it's a memory-resident program designed to work "on top" of a word-processing program, however, the process of moving back and forth from document to outline is greatly speeded. What is more, *Ready!* includes a command that automatically moves selected text from the outline to the document, so that a detailed outline can become, once transferred, a system of headings and subheadings in a document.

Ready! closely resembles *ThinkTank*, but it lacks many of *ThinkTank*'s beguiling features, such as the ability to store large amounts of text under a heading. *ThinkTank*'s text storage features are useful; indeed, the program can be used as a hierarchically-organized "textbase" management program, storing huge quantities (up to several megabytes) of text under a highly structured set of outline headings (see "A Scholar's Typology of Database Management Programs," *RWPN*, Vol. 3, No. 1 [January, 1985]). *Ready!*-made outlines not only lack *ThinkTank*'s text-storage features, but all *Ready!* outlines are limited to 32K. That's more than enough for creating a "suitable design" as (E.B. White would put it) for an essay or report, but *Ready!* cannot handle some of the more interesting text-management applications of outlining programs (for instance, storing all the lecture notes for a course under an outline of lecture topics).

Ready! has encountered stiff competition from a highly regarded upstart, Brown Bag Software's *PC-OUTLINE*, a well-crafted memory-resident outlining program. (Until recently, *PC-OUTLINE* was available on a shareware basis. Lamentably, it has now "graduated" to the commercial market.) *PC-OUTLINE*, like *Ready!*, limits the size of an outline, but the limitation is far less severe. The default setting is 64 kilobytes' worth of space for outlining, but a command-line option allows the user to specify a maximum size as small as 22K or as large as 576K. Outline files of no more than 22K are just fine for use as outlining tools when writing; they allow plenty of room for a word-processing program and other memory-resident software to coexist in memory. I've successfully loaded *Microsoft Word*, *Turbo Lightning* (a memory-resident spelling checker and thesaurus program), and *PC-OUTLINE* into my PC-compatible's 640K of RAM. [Use caution when loading more than one memory-resident program into your computer's internal memory. Like an uncharted frontier, the little-utilized memory above the 256K mark is a lawless realm; programs devised to inhabit it may not have developed civilized habits of coexistence with others, leading to strange (and often catastrophic) system crashes.] You can also use *PC-OUTLINE* as if it were an ordinary program loaded from DOS. If you have a large amount of free memory, you can create monstrous outlines containing hundreds of kilobytes of text.

PC-OUTLINE handles the blending of outlining and text entry by simply doing away with the clumsy distinction between them that *ThinkTank* makes. Any *PC-OUTLINE* heading can consist of multiple lines, and there is no limit to the size of a multiple-line entry (save those imposed by the maximum file size). Unlike *Ready!*, therefore, *PC-OUTLINE* can be used (as can *ThinkTank*) as a hierarchical database management program for the storage of textual data. *ThinkTank*, to be sure, lets you create much larger data files since their size is limited by disk capacity rather than the size of the internal memory. Yet 576K is more than enough for storing a term's lecture notes. In all likelihood, those who plan to use *PC-OUTLINE* as a database management program will not find the data-file size limit constraining unless they are planning to work with truly massive amounts of stored text. *PC-OUTLINE*, in short, gives you the best of both *Ready!* and *ThinkTank*, and at a bargain price.

PC-OUTLINE is commendable for many other reasons, which have been outlined in detail elsewhere (e.g., *PC: The Independent Guide to IBM-Standard Personal Computing*, March 25, 1986, in which *PC-OUTLINE* was the Editor's Choice out of a field of seven outlining programs, including *Ready!* and *ThinkTank*). The user interface employs pull-down menus, reminding one forcefully of Macintosh software or Ashton-Tate's *Framework*. What is more, *PC-OUTLINE* strikes many users as significantly more approachable and intuitively sensible than *Ready!* and *ThinkTank*. *Ready!* and *ThinkTank*, to be sure, are no slouches when it comes to user-friendliness. Yet both programs present you with a welter of keyboard commands and modes, which send even experienced users to the manual in search of a forgotten command to acomplish some relatively straightforward task. The program seems to have been deliberately intended to improve on *ThinkTank*'s already laudable achievements.

No matter how good a memory-resident outlining program is, however, a writer is still left with the tedious prospect of updating the outline after making changes in a document's structure. That is a task that the computer, in principle, can handle, a point made forcefully by the integration of outlining and word-processing features in *Framework* (*RWPN*, Vol. 3, No. 4 [April, 1984]). *Framework*, one of several programs that stemmed from the ill-fated integrated software fit of 1983-1984, blends four software functions (word processing, electronic spreadsheet, database management, and communications) in a single, massive program, but it is *Framework*'s word-processing function that is of interest here. With *Framework*, one writes in two modes: an outline mode, in which only the headings are visible, and a text mode, in which only the text is visiible. What is more, changes in the outline's structure are immediately reflected in the structure of the document itself. The result is a powerful tool for the revision of large text domains, as well as a useful index, visible at a keystroke, of the document's overall organization.

Although *Framework* pioneered the integration of outlining and word processing, my guess is that few writers would prefer it over conventional word-processing programs such as *Microsoft Word* or *WordPerfect*. In its original version, *Framework* lacked such indispensable amenities as superscripting, footnoting, and spelling checking. You couldn't simply scroll through a multiple-section document, moreover, without going back to the outline mode and selecting a different heading. Some of these deficiencies and clunky features have been remedied in the most recent version, *Framework II*, but for the word-processing market the program has run into stiff competition: a new version of *Microsoft Word* (3.0) that includes an exceptionally well-integrated outlining function.

Word 3.0 will doubtless influence the design of software for years to come. In its text mode, *Word* presents itself as it did in version 2.0: it's a full-featured, "what-you-see-is-what-you-get" word-processing program. (Version 3.0, however, is more nimble in updating the screen than its rather sluggish predecessors, and it includes a bundle of new features designed to make it more competitive with *WordPerfect*, its major rival). Unlike *Framework*, which forces you to organize a document using the outline-processing functions, a *Word* 3.0 document can be created as if the outlining functions did not exist. When you shift to the outlining mode, however, the same document reappears as a *ThinkTank*-style outline, replete with hierarchically-organized headings, facilities for hiding massive amounts of text under the headings, automatic sorting and numbering of headings, and tools for outline restructuring.

Word's outline processing capabilities become particularly powerful when blended with its style-sheet feature (*RWPN*, Vol 3, No. 3 [March, 1985]). In *Word*, a style sheet is a list of user-defined keyboard commands. Version 3.0 includes a new way to define keyboard commands so that they appear as headings in the document mode and as outline entries in the outlining mode. The keyboard command "ALT-H2," for example, can be defined so that it ***simultaneously*** creates what appears to be a second-level head in the document mode (i.e., one that's centered and underlined, with three blank lines above and two below) and a second-level entry in the outline mode (i.e., indented five spaces from the left margin).

Using the style-sheet feature in this way brings about a transparent and intuitively sensible link between the outline and the document. Viewed in the outline mode, a document (with its patterns of headings and subheadings) appears as an outline, in which (so long as the text is collapsed or hidden) the document's overall structure and organization is clearly visible. Viewed in the text mode, the document appears almost exactly as it will when printed. *Word* 3.0, in other words, provides precisely the tools to meet the needs of computer-using professional writers—namely, the need to maintain an accurate view of a document's structure and the need to predict in precise terms how the document will appear when printed.

Word's outlining mode not only provides a superior way to grasp the structure of a document's larger text domains; it also provides powerful tools for altering them. If you restructure the headings in the outline, the document is automatically altered so that it corresponds to the outline—all the text stored under the moved heading is moved with it. This powerful tool for text revisions makes it extremely easy to move large domains of text (say, a ten-page section of a chapter) and at the same time provides an accessible, automatically updated roadmap of the document's structure at any one point in time.

Microsoft Word's successful integration of outlining and word-processing functions points the way to the future evolution of software, but we still have yet to see a program that fully facilitates the writing process (as it has been revealed by research). Writers cycle back and forth not only between planning (e.g., outlining) and writing; they include revision activities in the process as well. Any word-processing program, to be sure, facilitates revision (at least in principle) by making it easier to make textual insertions and deletions, and integrated word processing/outlining programs greatly facilitate the revision of large textual domains. Yet a host of tools for revision, most of which were initially formulated during the creation of Bell Laboratories' *Writer's Workbench* software, have yet to be incorporated in personal computer word-processing packages. Memory-resident program modules could conceivably be developed, for instance, to monitor widely-accepted indices of readability, such as average sentence length, average word length, or the proportion of three-syllable words, and present a "pop-up" screen that displays these indices at the user's request. The future may see such innovations, but only if professional writers make clear to programmers what they would like to see in the next generation of software.

ThinkTank	Living Videotext, Inc., 2432 Charleston Road, Mountain View, CA 94943
Category:	Outlining program
List Price:	$195
Requires:	IBM PC, XT, At, or 100% PC-compatible, 256K RAM, DOS 2.0 or higher, and two disk drives
Summary:	An outlining program that facilitates the creation of huge, hierarchically-structured textual databases as well as the creation of writing guides. The program is not memory-resident; you must exit other programs to use it.
Ready!	Living Videotext, Inc. 2432 Charleston Road, Mountain View, CA 94043
Category:	Memory-resident outlining program
List Price:	$99.95
Requires:	IBM PC, XT, AT, or 100% PC-compatible, 112K of free RAM, DOS 2.0 or higher, and one disk drive
Summary:	A memory-resident outlining program, *Ready!* is so similar to *ThinkTank* that *ThinkTank* users will have little difficulty learning it. Unlike *ThinkTank, Ready!* is available at a keystroke while using other programs. The price paid for this accessibility, however, is a 32K limit in outline size and the sacrifice of *ThinkTank*'s paragraph or text-storage features.
PC-OUTLINE	Brown Bag Software, Inc., 2105 South Bascom Ave., Suite 164, Campbell, CA 95008, (800) 323-5335
Category:	Memory-resident outlining program
List Price:	$89.95
Requires:	IBM PC, XT, AT, or 100% PC-compatible, min. 90K free RAM
Summary:	Excellent memory-resident outlining program that significantly improves on *ThinkTank*'s user interface. Files are limited by the amount of free memory, so outlines of 576K can be created when the program is loaded from DOS. When used as a memory-resident program, *PC-OUTLINE* can be adjusted to use as little as 90K of free RAM. Because the program allows multiple line entries, it arguably combines the best of both *Ready!* and *ThinkTank* with an improved user interface.
Microsoft Word 3.0	Microsoft Corporation, 16011 NE 36th Way, Box 97017, Redmond, WA 98073, (206) 882-8080
Category:	Integrated word processing/outlining program
List Price:	$450
Requires:	256K RAM, two disk drives, DOS 2.0 or higher
Summary:	Full-featured word-processing program that successfully integrates a high-quality outlining function. Changes in the outline's structure are automatically reflected in the document.

Contributing Editor ***Bryan Pfaffenberger***, a writer and anthropologist who lives in Charlottesville, Virginia, is the author of *The Scholar's Personal Computing Handbook: A Practical Guide* (Little, Brown, 1986) as well as several other books on personal computing. His *Personal Computer Applications: A Strategy for the Information Society*, a college-level introductory textbook that focuses on personal computer application software, will by published in 1987 by Little, Brown; Richard D. Irwin, Inc., will publish his *Business Communications in the Personal Computer Age* the same year. Currently, Bryan is developing an anthropological approach to technology and technological innovation.