# A Shakespearean in the Electronic Study

Hardy M. Cook
Bowie State University
Bowie, Maryland

I clearly recall the last paper I wrote on a typewriter: I remember especially the sheer physical effort of typing those thirty-eight pages from my hand-written, cut-and-pasted draft and the Wite-Out all over my fingers and typewriter. I would type for forty-five minutes to an hour at a time until I became exhausted; then I would visit computer stores, muttering under my breath, "Never again." Indeed, my writing methods have changed significantly since 1983 when I purchased my first personal computer. As a result of my interest in computing, I am currently directing a project at my institution to explore ways computers can be integrated into our composition courses to assist instructors with their teaching and record keeping and students with their writing. During the 1989-1990 academic year, we worked on a data base to track a student's progress in basic skills through the composition sequence. For several years, we have included word processing and other computer applications in our writing courses and have ordered authoring software to fill gaps in existing software and to construct exercises specifically for our students. In academic year 1990-1991, we will be setting up a computer-supported writing facility—a network of twenty computers that will include not only word processing but also prewriting and revising software. In this paper, however, I am con-cerned with ways that I have incorporated computer applications into my work as a Shakespearean.

I purchased my first computer primarily for word processing. I relished having a spelling checker and later an electronic thesaurus. I welcomed the ease with which I could revise my drafts. I enjoyed sitting back and letting my printer effortlessly churn out corrected copies without the dread of having to retype a page because I wanted to make major or minor revisions.

This computer came bundled with WordStar, a spreadsheet, and a data base. The first software I bought to supplement these was ThinkTank, an outlining program. I recently upgraded to the more elaborate GrandView and still begin most my writing projects and oral presentations by working with this outliner. In the past, I have used outlining software to keep a record of the complete credits of The BBC TV Shakespeare. In this case, treating the outlining program as a data base enabled me easily to search all the parts an actor played in the series, to navigate through the complete list by collapsing and expanding the credits of individual productions, and to export the credits to an ASCII text file for inclusion in my WordStar-generated documents.

In addition, I find editing and proofreading software helpful. I began with the primitive Punctuation & Style. This pro-gram includes Cleanup, which searches for mechanical problems, and Phrase, which examines diction and usage. I eventually replaced Punctuation & Style with Grammatik because I found it more comprehensive and easier to use. Recently, I have discovered Editor, a program soon to be marketed by the Modern Language Association. The philosophy of Editor differs markedly from that of Grammatik and of its major competitor RightWriter. These two programs are conceived primarily for the business writer and are designed to promote rapid changes and to favor the plain style of writing. Editor provides no quick fixes: it "can find more than fifteen thousand common writing problems in more than forty categories" and is intended to encourage hard copy engagement with one's draft, "offering some of the expert attention that is characteristic of a good teacher or editor marking up a manuscript for improvement or publication."

One begins Editor by running a utility program that numbers the lines in the text, either by creating another file on disk or by sending a copy of the file with line numbers to the printer. One next invokes the Usage program to generate a list of suggestions keyed to the numbered lines, again with the option of creating a file on disk or sending the output to a printer. The Usage program's dictionaries analyze diction, punctuation, mechanics, wordiness, trite expressions, cliches, slang, jargon, colloquialisms, awkward expressions, and commonly misused words. The writer then goes through the list making appropriate changes on the line-numbered draft before transferring those changes to the original draft. The authors of Editor firmly believe editing is done more accurately on paper than on a computer screen, and their program encourages that approach 1 [". . . the Editor programs are not 'interactive' with the user's text–that is, they do not allow immediate, on-screen corrections–because studies indicate that a computer screen is not the best environment for thoughtful revision of one's work. In keeping with recent discoveries about writing and word processing (Collier, 1983; Haas, 1988), we want writers to use printed copies of their texts for markup and revision." (Elaine and John Thiesmeyer, Editor manual, page 4.)], although one may, if one wishes, revise the files on screen with a word processor that supports split-screen editing.

When I want to study the differences between two versions of a paper, I execute CompareRite, which produces a redlined document that shows all variations between the two copies being com-pared. With this program, one controls the appearance of additions and deletions in the redlined text–the printout or screen format–and the comparison style–the way CompareRite organizes those changes.

In my work, I also rely on Bibliography Generator. I have found that this inexpensive bibliographic program adequately meets my present needs. With it I can set up a variety of bibliographies and have them automatically formatted in the MLA style. I can make a selected bibliography from a master bibliography or combine two or more bibliographies into a larger one. I can print bibliographies from the program itself, or I can export a bibliography to an ASCII text file and edit it with WordStar. Bibliography Generator can append annotations to entries in a bibliography, but it does not have the more sophisticated feature of Pro-Cite, which allows multilevel sorting, multiple automatic formats, and complex searching.

When I am writing, I frequently take advantage of WordStar's Windows function. With it, I can view two documents at once, easily switching between them. I also can copy from one file to another or simultaneously edit two parts of the same document. However, when I want to look at a file on another subdirectory or when I know what text I would like to examine but I do not know the file that contains that text or even where that file is lo-cated on my hard disk, I run Gofer, a memory-resident program. Without leaving the document I am working on, I invoke Gofer. At the entry screen, I can type key words or phrases into as many as eight fields, using Boolean Logic (and, or, not, nearby) search criteria. After I have selected the drive, path, and files to search and have indicated whether the results should be viewed on screen, written to a disk file, or sent to a printer, Gofer be-gins searching. I usually view the results on screen; then I can decide whether I want to mark the text to be exported to the document I am preparing or to a file or printer. Besides my papers, I keep many notes I have taken from my reading on my hard disk. Thus, with Gofer, I effortlessly can include in my writings direct quotations or summaries from my sources without having to retype them.

Recently I have discovered an alternative to typing out notes from my reading; now I scan selections from journals and books and convert them to text files, using a scanner and text recognition software. I chose to buy an inexpensive hand-held scanner, ScanMan Plus, even though I have seen advertisements for full-page scanners in the $500 range. I originally thought that I would have more flexibility with a hand-held scanner, believing that I could scan books and journals directly without having to photocopy them first as one must do with a full-page scanner. I must report, however, that I have not yet developed techniques that enable me acceptably to scan directly from books and journals. As my description of the procedure below will indicate, I am also slightly disappointed in the software I selected to convert the scanned images to text files, Read-It O.C.R.

Before using optical character recognition (OCR) software, one has to run the scanner's software, scan the text, and save the text in the TIFF format, one of many digital formats for graphics files. One scanner parameter of importance to the ease and accuracy with which OCR software operates involves resolution. ScanMan Plus scans at 100, 200, 300, and 400 dots per inch (dpi). Generally, one selects the scanner's highest resolution for OCR applications. Unfortunately, the higher the resolution the smaller the total surface area one can scan with a hand-held scanner. I scan at 300 dpi for maximum text input (350 to 400 words per scan) and at 400 dpi for shorter text scans. For OCR software to work most efficiently, lines of text must be scanned evenly: skewed lines cannot be recognized. Before scanning, one must set the width and length of the scan. With texts printed in columns, like PMLA, I set the width to 3.25" and scan vertically (the length will vary depending on how much text one wishes to include). With texts that are in a single column, like Shakespeare Quarterly, I set the length to 5" and scan horizontally (here, the width will vary). Before saving scanned text as a graphics image, one must use the scanner's software to cleanup the image as much as possible. Such cleanup might include erasing unwanted text or rotating horizontally scanned text.

After one has a cleanly scanned image saved to a graphics file, the OCR software can begin to process it. The image produced by the scanner consists of a pattern of pixels (dots). OCR software compares the shapes of these dots to the shapes of the letters of the various typefaces stored in type tables. Because Read-It O.C.R. is "trainable," it can be taught to recognize al-most any typeface. Although the program contains a few type tables of popular fonts, one still must, because of the many variables involved, teach the software to identify each new type-face, that is, build a type table for each typeface. To build a type table, one first chooses an existing type table to adapt or runs one of the learn modes to create the initial table, after which one executes the recognize and learn command. In this mode, as it processes the graphics image into an ASCII text file, Read-It O.C.R. will stop when it cannot find a good match for a character; one then has manually to enter the correct character from the keyboard. This learning procedure is invoked until the type table is so refined that it achieves a nearly 100% recognition accuracy. The process of building and refining type tables is a time consuming one, but once one has a well-refined table, processing to an ASCII text file only takes a few minutes. Since I bought Read-It O.C.R., I have discovered another OCR program, Omni-Page, that does not require building type tables; Omni-Page, however, costs roughly three times as much as Read-It O.C.R.

I also transfer text to my hard disk from my Radio Shack Model 100 laptop computer through a null modem cable. The Model 100 is the original laptop computer2 [The Model 100 was superseded by the Model 102, a slightly smaller but in other ways virtually identical computer. In the past few years, however, the Model 102 has been eclipsed by MD-DOS compatible portable computers.]. It weighs about four pounds, operates for considerably more than its estimated twenty hours on one set of four AA alkaline batteries, and can easily be carried in one's briefcase. The Model 100 does have many limitations: for example, its maximum RAM is 32K, and its LCD screen only displays eight lines of forty characters. To overcome some of the limitations of the built-in word processing program, I added a ROM chip that includes Write ROM, a text processing program. This program gives me greater control over my text output and even has a page preview function.

I have employed the Model 100 in many ways. When I was writing on The BBC TV Shakespeare series, for example, I would take notes on the productions with the Model 100, which creates ASCII text files. Normally, I would sit in my living room or in the media room of the library with the Model 100. As I watched a production, I would pause the video recorder frequently to take notes. Although the 32K RAM capacity restricts the maximum size of a text file, I always had sufficient capacity for my ten to twelve page production notes. When I finished the notes on a production, I would connect the null modem cable from a serial port (communications port) of my desktop computer to the serial port of the Model 100; using the built-in modem in the Model 100 and the Write ROM program, I would send my text to the telecommunication software, Mirror, on my hard disk. Then, I simply edit-ed my notes in WordStar.

With a null modem cable, I did not need a modem with my desktop computer to receive files from my Model 100, but when I did purchase one for my computer, I discovered several uses for it in my work. With a modem and telecommunications software, I can access information services like Compuserve. I can send and receive E-Mail at my university Internet address. I have even joined Humanist, an international electronic discussion group for computing humanists. I have not yet accessed the MLA Bibliography on the DIALOG Information Service, but I regularly connect with academic library catalogues through Internet. If I want to know, for example, the publication information for a book mentioned in my reading, I am less than a minute away from an online catalog and the answer to my query.

Finally, I regularly run WordCruncher with the Riverside Shakespeare. When I am reading, I often check references and frequency distribution claims3 [For example, the word “blood” appears more times in King John than in any other of Shakespeare’s plays.]; I also look up quotations in context and search for echoes of them in other of Shakespeare’s works. In addition, I export quotations to ASCII files to include them in my writing or in handouts or tests for my students. I realize that I now am only taking advantage of a small portion of this program’s capabilities and intend to pursue other applications of it.

I began this paper by claiming that my writing methods have changed significantly since I began computing. At my paper’s end, I realize that I oversimplified the impact the personal computer has had on the way I work in general. Not only do I have writing and editing tools that I could not have imagined a dozen years ago, but I also have far greater and faster access to information than scholars of a generation ago. I eagerly await the new technologies and applications on the horizon. Now, I am trying to figure when I can afford a CD-ROM drive, yet I know that as soon as I have a CD-ROM drive I will begin considering when I will be able to purchase the Oxford English Dictionary on CD; then there is the MLA Bibliography, and so on, and so on.

Hardy M. Cook
Bowie State University
HMCOOK@BOE.TOWSON.EDU