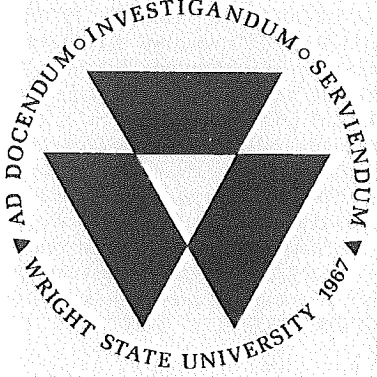
TEXTECHNOLOGY



Volume 1 Number 4
July 1991

In this issue . . .

EDITOR'S CHOICE

Jim Schwartz

How Are We Doing?

Newsletters, unlike books, journals, and many magazines, prove valuable to readers by providing timely and substantive information in concise formats—not in extended essays. While it is true that some topics require more elaboration and, subsequently, more column inches than others, a newsletter editor should be ever vigilant against her or his desire to simply "fill space."

As charter *TEXT Technology* readers, you all have had exposure to many fine articles keyed by a number of expert columnists. Some of these pieces have been short, while others have spanned as many as four pages—one-fourth of the entire 16-page publication.

Now the Editors ask that you tell us how we're doing. Which columns do you like? Which do you think could be improved? Would you like to see more or fewer entries in TEXTechography? What about topics you would like to see us cover. In-depth reviews of software (specify types or products)? Hardware? What about the guy who keys Editor's Choice—keep him, or fire him?

In any event, let's hear from you. Truly, we want this to be your publication.

TEXT TECHNOLOGY

Editors
Jim Schwartz
Arthur A. Molitierno

Contributing Editors
Randal Baier
Cornell University
Paul Fritz
University of Toledo
Eric Johnson
Dakota State University
Stephen Miller
Oxford University
Brad Morgan
South Dakota School of Mines & Technology
Guy Pace
Washington State University
Ian Richmond
University of Western Ontario

This publication is funded in part by a research grant from the *Wright State University President's Club.*

TEXT Technology. Volume 1, Number 4. ISSN: 1053–900X. Also indexed by ERIC, COMPUTER AND CONTROL ABSTRACTS (INSPEC), COMPUTER LITERATURE INDEX, LITERARY CRITICISM REGISTER, and SOFTWARE REVIEWS ON FILE. *TEXT Technology* is published bi-monthly by Wright State University—Lake Campus, 7600 State Route 703, Celina, Ohio, USA 45822-2921; (419) 586-2365. Postage paid at Celina, Ohio. SUBSCRIPTION RATES US funds: $20 (US); $27 (Canada); $35 (Foreign). Address all subscription inquiries and manuscripts to the Editors, *TEXT Technology*, Wright State University—Lake Campus, 7600 State Route 703, Celina, Ohio, USA 45822-2921. Note that the Editors are not responsible for any unsolicited manuscripts. Allow 4 to 6 weeks for subscription processing. Selected back issues are also available; contact the Editors for more details. POSTMASTER: Send address changes to *TEXT Technology*, Wright State University—Lake Campus, 7600 State Route 703, Celina, Ohio, USA 45822-2921.

WRIGHT STATE LAKE CAMPUS

TEXT Technology is composed on a Northgate Elegance 486-25 computer and printed using an IBM Personal Page Printer II Postscript printer. Images and text are scanned with a Hewlett-Packard ScanJet Plus scanner. Software used includes Microsoft *Windows*, Aldus *PageMaker*, *Caere Omnipage 386*, *WordPerfect*, *HP Scan Gallery* and *Paintbrush*, Micrografx *Designer*, *CorelDRAW!*, Lotus *Ami Professional*, Microsoft *Windows Write*, and Symantec's *Q&A*.

COLUMN ONE

Eric Johnson

Fast Processing for Text

For those who employ microcomputers for text processing, there can be substantial advantages in using faster, more powerful hardware. This discussion will focus on the comparative computing speed of MS-DOS computers containing Intel's 80386 and 80486 processors, but similar increases in speed will be gained by using Apple Macintosh computers with advanced Motorola chips.

Table 1 shows a comparison of the timings of six programs executed on (1) a clone of an original IBM PC containing an Intel 8088 processor running at 4.77 Megahertz (MHz) and on (2) an ALR PowerCache 4 containing an Intel 80486 microprocessor running at 25 MHz. The programs ran from forty to one-hundred times faster on the 80486.

PROGRAMS	80486@ 25MHz	8088@ 4.77MHz	Factor
Program 1 (math)	1.0	41.4	41.4
Program 2 (math)	4.8	209.9	43.7
Program 3 (text)	0.8	33.6	42.0
Program 4 (text)	0.5	34.2	68.4
Program 5 (text)	2.9	209.4	72.2
Program 6 (logic)	1.4	147.0	105.0

Table 1. Comparison of times (in seconds) of Intel 80486 and 8088 microprocessors.

Programs one and two contain only statements that perform very simple integer arithmetic and numeric comparisons; they are not the kinds of programs that are written for text processing, but they are included for comparison. It is interesting that while the faster processors are sometimes thought of as number crunchers, and while they are, of course, much faster at numeric calculations (by a factor of about forty), the factor of advantage for non-numeric computing is at least as great, and it can be far greater. Literary and linguistic computing often requires more processing power than number crunching.

Programs three, four, and five are versions of a grammar and style checker I wrote. Each contains about one thousand lines of SPITBOL code. Each performs hundreds of pattern matches, comparisons of lexical data, and searches using tables and arrays. These are exactly the kinds of programs that are used for text processing.

Programs three and four are similar, except that different algorithms are used to accomplish the same results; program three runs 42 times faster on the 80486, and, program four is 68 times faster (obviously, the algorithms in program four allow faster execution on the 80486). Programs four and five are virtually identical, but the data set analyzed by program five is larger; notice that the 80486 becomes still more efficient as the amount of data increases: 72 times faster, rather than 68.

The relative amount of time each program uses to execute a program on each computer is significantly better for the 80486; however, the absolute timings are perhaps even more important. The output of programs three and four will be produced in less than a second on an 80486, and it will seem almost instantaneous. The output of these programs on an 8088 will appear only after about half a minute.

Thirty seconds is not a long time, but it will seem much longer to a user sitting idly in front of a

computer screen. The nearly three seconds required for execution of program five on an 80486 will be noticed by the user, but it should not seem significant. The three and one-half minutes needed for execution of program five on an 8088 will seem exceedingly long.

There are a number of differences between microcomputers that contain 8088 processors and those that contain 80486 processors; the differences in disk access speed and data transfer rates (and the performance of related hardware used for input and output) can significantly affect timings. Because of these differences, program six was included. It is an Icon program that performs a complex series of logic tests; it is extremely processor intensive, and input and output operations are minimal. As Table 1 demonstrates, the difference in speed between processors is even greater for program six; the 80486 computer is over one hundred times faster.

PROGRAMS	80486@ 25MHz	80386@ 20MHz	Factor
Program 7 (text)	0.82	3.73	4.55
Program 8 (text)	2.93	16.58	5.66
Program 9 (logic)	2.47	10.18	4.12
Program 10 (logic)	10.00	47.00	4.70
Program 11 (text)	926.00	3559.00	3.84

Table 2. Comparison of times (in seconds) of Intel 80486 and 80386 microprocessors.

Table 1 shows that an 80486 can be forty to one-hundred times faster than an 8088, particularly for non-numeric processing. If a researcher's time is important, the 80486 is greatly to be preferred to the 8088. But a researcher may wonder if it is necessary to jump to the 80486. Perhaps the far more common (and lower priced) 80386 will suffice.

PROGRAMS	486-25	386-33	386-25	386-20
Icon 1	2.52	4.19	7.78	10.16
Icon 2	10.99	20.07	35.85	46.89
Icon 3	1.37	2.33	4.27	5.61
Icon 4	0.55	1.05	1.85	2.40
SPITBOL 1	0.74	1.35	2.31	3.11
SPITBOL 2	0.77	1.32	2.25	2.97
SPITBOL 3	1.26	2.25	3.79	5.33
SPITBOL 4	0.66	1.05	1.98	2.53
SPITBOL 5	0.88	1.59	2.86	3.68

Table 3. Timings (in seconds) of non-numeric programs on 32-bit Intel microprocessors 486-25, 386-33, 386-25, 386-20.

Table 2 illustrates a comparison of five programs run on a CSS microcomputer (containing an 80386 DX processor running at 20 MHz) and the ALR PowerCache 4. Programs seven and eight are grammar and style checkers similar to programs three and four. Programs nine and ten are Icon programs similar to program six. Program eleven is a SPITBOL program that calculates the interval among all occurrences of all words in a text.

Each of these five programs is written for the 32-bit processing capacity of the 80386 and 80486. The timings of the programs are all about four times faster on the 80486 (from a factor of 3.84 to 5.66). The comparative advantage is significant, but, again, the absolute time required for program execution may be most important. On the 80486, program eleven will tell the user the interval among words in a long novel in about fifteen minutes, but when executed on a 80386, program eleven will require about an hour to give the same information.

A user might want to run program eleven several times to obtain comparative information for several novels; if an 80486 is available, the information for four novels can be compiled in an hour; probably a researcher would do that. If only an 80386 were available, the same four novels would require nearly four hours to test; half a day might seem too long to bother with the information.

Program eleven will not run on an 8088 because the program is written for 32-bit processing found only in the 80386 and 80486, but if it could be executed on an 8088, four novels would require about forty hours of processing time: a normal work week!

An 80386 running at 20 MHz has only one quarter of the speed of the 80486, but perhaps an 80386 running faster than 20 MHz will do much better. Table 3 shows the timings of nine programs executed on an 80486 running at 25 MHz, and on 80386 processors running at 20, 25, and 33 MHz.

PROGRAMS	486-25	386-33	386-25	386-20
Icon 1	2.52	4.19	7.78	10.16
Icon 2	10.99	20.07	35.85	46.89
Icon 3	1.37	2.33	4.27	5.61
Icon 4	0.55	1.05	1.85	2.40
SPITBOL 1	0.74	1.35	2.31	3.11
SPITBOL 2	0.77	1.32	2.25	2.97
SPITBOL 3	1.26	2.25	3.79	5.33
SPITBOL 4	0.66	1.05	1.98	2.53
SPITBOL 5	0.88	1.59	2.86	3.68

Table 3. Timings (in seconds) of non-numeric programs on 32-bit Intel microprocessors 486-25, 386-33, 386-25, 386-20.

All of the nine programs are written for and use 32-bit compilers. Icon programs 1 to 4 perform a series of logic tests; they are processor intensive with minimal input and output operations. SPITBOL programs 1 to 5 are similar to program 4; they make pattern matches, use table lookup, and reference arrays.

The relative speeds of the 32-bit processors are quite consistent. Running at 25 MHz, the 80486 is not quite twice as fast as the 80386 at 33 MHz; it is about three times faster than the 80386 at 25 MHz; it is about four times faster than the 80386—20. In the near future, there will be still faster speeds. Manufacturers are starting to ship computers containing the 80486 running at 33 MHz, and microprocessor speeds of 50 to 100 MHz are being tested.

Probably, these will be the fast research machines, and the 80386 running at 20 to 33 MHz will be the day-to-day workhorse. With this is mind, I did not test microcomputers using 80286 or 80386 SX processors because their price is not enormously lower than genuine 80386 machines, and their speed is significantly slower.

* * * * * * * * * * *

Eric Johnson is Professor of English and Dean of the College of Liberal Arts at Dakota State University, Madison, SD 57042. He is the Director of the International Conference on Symbolic and Logical Computing, and he has published articles and reviews about computer programming, writing, and literary study. His BITNET address is ERIC@SDNET.

THE RESEARCH ASSISTANT

Paul A. Fritz and Jeffrey D. Peters

The Senator's Office: An Information Decay Simulation Using Computer Networks

Advanced students can program, manipulate databases, manage spreadsheets, and process words, but can they do anything with that information? Do they design projects as per instruction, deal with information decay, cope with overload, write abstracts, store information, assess source credibility, store and retrieve messages, or design logical "search" procedures?

Learning what to do with information is already a vital professional skill students will need on the job. The literature on employment calls college graduates to demonstrate explanatory skills, staff coordination/organization skills, decision-making skills, and imaginative ways to use computer networks (Patton, 1988; Bump, 1985; 1987; Newman, 1989; Batson, 1989; Bridwell-Bowles, 1989; McLean, 1989).

Wheeler (1991), surveying patterns of message transmission by mail services, found that a handful of cities had become informationally dominant in the U. S., with the other 40+ major metropolitan areas becoming informationally dependent on them. Certain cities (New York, Atlanta, Chicago, Dallas-Ft-Worth, and Los Angeles) have learned how to control both the quantity of information and, more importantly, the quality of information received by "less-important" metropolitan centers.

"Clearly the vitality of businesses in the main power centers depends on their ability to produce and receive the latest, most specialized information. And the health of American businesses in other cities depends on their relationship to these information sources. . . . the vitality of America's cities will depend on how well they can communicate with power centers and whether they can assemble a critical mass of specialized information sources" (Wheeler 42).

Recognizing students' needs for managing and manipulating information, we designed a classroom simulation for advanced students. Here students: handle network messages on a staff, organize a work task, meet deadlines, and learn how message decay and message credibility affect the value of information.

SIMULATION OVERVIEW

The instructor acts as Senator; students are divided into teams and act as either Senator's Staff Members or as Research Interns for private "think tanks." Each cycle of the simulation runs for a five day week. At the start of the week, typically Monday, the Senator makes a request of the staff to write a "white paper" on a current issue.

The staff members, sub-contract the research for the white paper to interns. The interns prepare abstracts on current issues and send the abstracts to the staff members. The staff members use the abstracts to write formal reports and then send them to the Senator.

Neither staff members nor interns are paid a traditional salary for their labors. Their wages are computed on the basis of the quality of information they present. Quality is determined by a numerical "decay" factor built into each White Paper. Both staff members and interns also have overhead costs.

A payment system rewards members for their work. From earnings, all members must pay fees for use of the computer bulletin board used in the simulation. Final grades are computed on the

normal curve based on the total sum of each team's earnings.

EQUIPMENT NEEDED

At least two computer clusters and communication hardware are needed. The ideal is to have one terminal per team and have each team located in different parts of the campus. It is best if the Research Intern teams could be located in or near the library.

Students are organized in teams of three to five members each. A class of eighteen students would have three teams of "Staff Members" and three teams "Research Interns." Each team elects a "Manager."

An instructor acts as "Senator" in the simulation and makes a White Paper assignment at the beginning of each week.

A panel of three faculty members judges the entries and selects the winning White Paper.

PARTICIPANTS' JOB DESCRIPTIONS

Senator:

1. The Senator posts a "Call for a White Paper" on Monday mornings. She or he needs the paper to summarize her or his position on a current issue in a speech the Senator plans to deliver to constituents on Friday evening. This call is printed in a confidential file. This file can be accessed only by the managers of each Staff Member team.

2. The confidential file indicates the topic of the white paper, the audience, the desired use of the information, and the deadline.

3. The confidential file may not be duplicated. It can be read only once by the Staff Member managers, and then it is purged from the screen. This process tests "listening" skills for their adaptation to the electronic age. The Managers' jobs are to interpret the original "call" for the members of their staffs.

4. The prize for the winning white paper is $500,000 if the entry is submitted by the Friday deadline. An additional bonus of $100,000 is added for each day to the winning submission if the entry is turned in earlier than the deadline and is selected as the winner. If the manuscript is submitted past the deadline, the decay factor is multiplied times "10" and subtracted from the team's total fund.

5. The best White Paper must also have the lowest "message decay factor" among the entered papers.

6. The instructor receives one White Paper entry per "staff" team, copies the files, and mails the submissions to the judging panel. The entries are evaluated and the results posted by bulletin board in thirty minutes. The winning essay is also displayed as a model for future rounds of competition.

7. The Senator also receives a confidential file from each "staff" and "intern" team disclosing team finances.

Staff Member Teams:

1. Staff members communicate with their own Managers and each other. The Managers communicate with the Research Intern managers but not with members of the research team.

2. The Staff Managers interpret the original call for papers to the Staff Team. To avoid undue publicity prior to the speech, they cannot duplicate the original call for a white paper from the Senator's office. The Staff members confer and craft "want ads" for the bulletin boards read by the Research Interns. This procedure is the vital link in the simulation. If this original "call" is misinterpreted, the resulting research and the quality of the White Paper will be flawed.

3. The Staff Teams advertise calls for abstracts on the bulletin board accessed by all members of the simulation.

4. The staff members must pay for abstracts they purchase from the Research Teams.

5. Member teams each have a budget of $100,000 for the simulation. This sum must last for 10

rounds of the simulation. Of course, the wealth of each team may rise or wane depending on the fortunes of the simulation.

6. Teams receive the abstracts, paying for them by negotiating a price with the research team manager, and publish the White Paper by the deadline.

7. The staff members compute and keep their financial records and send a summary to the Senator at the end of each week.

8. The staff members are taxed for unused abstracts that are in their inventory and not needed in the preparation of the White Papers.

Research Intern Teams:

1. Research Team members read the Staff Members' advertisements for abstracts as bulletins posted by team members on the main board.

2. In uniform style, team members create abstracts of research gathered in the library. Each abstract contains title, author, and key search words and context information. Each team, in an organizing meeting, determines a style for its abstracts that is at once unique and also compatible with the work of the other teams of research interns.

3. The abstracts have an automatic "decay factor," a number and date that follow the citation line, Eg. (Smith, 1991) [1]. If the abstract is purchased and used on Monday, it would have a decay factor of "1;" on Tuesday, it would have a decay factor of "2"; Wednesday, "3"; Thursday, "4"; Friday "5."

4. Teams may advertise "samples" of their abstracts as offerings on the bulletin board. Of course, the "samples" do not disclose the full content of the abstracts. Once an abstract is sold, it cannot be duplicated and sold again. Once abstracts are offered for sale, they are not removed from the bulletin board until sold.

5. The sale price of abstracts may be any negotiated sum, but the general rule is that the abstract's value decreases $100 per day that it lingers on the bulletin board, to a low of $100 for a "dead" piece of information.

6. The abstract teams pay a toll of $10 per abstract ad to publish their wares on the bulletin board.

7. Team Members keep financial records of purchases and sales and send the information to the Senator at the end of each week. The Abstract teams fix the decay factor before the abstract is sold. Each team keeps lists of decay factors for future verification.

SIMULATION PROCESS

On Mondays, the beginning of a business week and the start of each new round, the Senator advertises the call for a white paper in confidential file to the staff member Team Managers. The managers may not share the information with the research interns.

The staff teams advertise the dimensions of the needed information on the bulletin board and are charged $100 for each billboard. The intern teams advertise their research "wares" on the bulletin board and are charged $10 per billboard.

The staff members purchase the abstracts and pay the negotiated price to the interns. The staff members write White Papers for the Senator by incorporating the abstracts—one white paper per team. They send the papers to the Senator.

The Senator sends the competing White Papers to the panel of judges, computes a decay factor, and announces the best "White Paper" for that week. The prize is awarded to the winning staff team on the basis of the best paper with the lowest decay factor.

The $500,000 prize is given to the winning team. One-quarter of this prize money is then divided among the abstractor research team who contributed to the winning White Paper. The team meetings are held at the conclusion of each round to help team members process their reactions to the simulations. The team meetings are a chance to adjust the rules to a more amicable fit for team members.

BENEFITS

The following benefits may be realized by employing simulations such as the one discussed in the article:

1. Such a team simulation enables students to gain a perception of how informational value may rise or fall depending on how "current" it is.

2. Students can attach a dollar value to information and observe their fortunes rising and waning-depending on the use made of their "critical mass" of data. They are taxed for unused information.

3. Students learn the importance of good "listening" skills to gain the full dimensions of the original assignment. They also learn the cost of mistakes made interpreting the original request.

4. Students also gain experience in small group communication as they attempt to clarify information given on the initial request.

5. They gain abstracting skill in quickly reading pertinent information and re-writing it in a form accessible and useful to others.

REFERENCES

Batson, T. (1989). *A selective national survey of ENFI real-time conferencing in the composition classroom.* Paper presented at the Fortieth Annual Meeting of the Conference on College Composition and Communication, Seattle, WA, March 16-18, 1989.

Bridwell-Bowles, L. (1989). Designing research on computer-assisted writing. *Computers and Composition*, 7, 79-91.

Bump, J. (1987). CAI in writing at the university: Some recommendations. *Computers and Education*, 11, 121-33.

Bump, J. (1985). Metaphor, creativity, and technical writing. *College Composition and Communication*, 36, 444-53.

McLean, R. S. (1989). Megatrends in computing and educational software development. *Education and Computing*, 5, 55-60.

Newman, J. M. (1989). Online: Dealing with information. *Language Arts*, 66, 58-64.

Patton, J. (1988). *Business computer applications I. Curriculum guide.* Commerce, TX: East Texas State University.

Wheeler, J. O. & Mitchelson, R. L. (1991). The information empire. *American Demographics*, 13, (March), 40-43.

* * * * * * * * * * *

Paul A. Fritz is an Associate Professor of Communication at the University of Toledo. Jeffrey D. Peters is one of Paul's research assistants. Besides being an "avid" Mudhens fan, Paul does research in critical thinking methods and in medical communication (aged/frail adults). He may be reached at (419) 537-2006 (office) or through BITNET--FAC0287@UofT01

THE CLIPBOARD

Ian Richmond

Better Than a Shoe Box

Are you still keeping your research notes on slips of paper stored in a shoe box? Or tacking notes to yourself on a bulletin board where they get covered by more notes to yourself? Or writing them on sticky little paper rectangles that are supposed to stick to anything, but usually drop off your desk lamp? If so, you're not alone. These traditional text storage and retrieval methods are still very much in favor despite the advent of the microcomputer age.

It's true: the popularity of wordprocessors among academics has led many of us to store research notes in wordprocessor files. Despite our enthusiastic acceptance of word processing, our profession has been less eager to embrace the benefits of text storage and retrieval software.

Most academics, it seems to me, need to store and retrieve two types of textual information. There are first of all the little, unconnected notes we write to ourselves and often have to keep for long periods of time—things like student counselling notes, reminders about meetings, subscriptions and deadlines for papers, addresses, telephone numbers, FAX and electronic mail addresses. The other type of text is related to our research and teaching: reading notes, bibliographical references, excerpts and even complete texts for analysis and study.

While you could keep all this in wordprocessor files, it would not be easy to locate a given text item or to use it once it was found. The limited sorting and retrieval capabilities of most wordprocessors usually means, in fact, that the file must be printed to be useful. Wordprocessors are wonderful, but they are better at creating text than at storing and retrieving it. (Yes, I do know *Nota Bene* has a built-in textbase feature, but it works only with *Nota Bene*.) Using a wordprocessor as a database really isn't much better than the old shoe box and the sticky notes. There are, however, several programs that are designed to make storing and retrieving these diverse bits of text as easy and as fast as possible. The drawback is that no single program does it all; you will likely need two to cover all your text storage and retrieval needs, just as, in the olden days B.C. (Before Computers) when you needed both a shoe box and sticky notes.

To meet the first type of textual need—the dozens (hundreds?) of bits of disconnected information that are the bane of our professional lives—there is a category of software that works very much like electronic sticky notes. These programs can be defined loosely as freeform databases. You type in any old piece of information you want to save for future use. You don't have to type it in any set format or order. Later on, when you want to find it again, you can search for it using any word or phrase that you happen to remember as belonging to the item you need.

You can also browse through the database one record at a time, in case you can't remember anything at all except that you saved some item of significance. The best of these "sticky notes" programs, such as Broderbund's *Memory Mate* (my personal favorite) and Micrologic's *Info Select*, do a number of nifty tricks that go beyond mere information storage. Both can be run in memory-resident (TSR) mode, for example. Thus, you can have all your bits of information readily available even when you are working in some other program, such as a wordprocessor , a database, or a communications program.

Not only will these programs let you see the information you need, but they also let you cut and paste it between themselves and whatever other application you are running. If you need an address for a letter you are writing in your wordprocessor , you can find it in *Memory Mate* or *Info Select* and then paste it, without retyping it, directly into your wordprocessor text.

This feature is invaluable for working with applications such as e-mail. I keep all my e-mail addresses in *Memory Mate*, for example, then just paste the one I need into my communications program when sending mail. When I want to keep an e-mail message I've received, I copy it directly from my PC's screen into *Memory Mate*.

For short messages, this is much faster than first writing the message to a file, then downloading it. Similarly, rather than mess around with the less than user-friendly editor on the mainframe, I usually write my e-mail messages in *Memory Mate*, then just paste them into the e-mail program. (*Info Select* works just as well for this.) This cut-and-paste feature alone is worth the price of either program, although *Memory Mate*, at less than $40 and with the more flexible cut-and-paste facility, is a clear winner here. *Info Select* will import or export only a full screen at a time, whereas *Memory Mate* lets you select and mark the text you need.

> ***. . . toss out your old shoe box and stop sticking colored rectangles on your desk lamp.***

When you add to this the ability to have messages pop up as reminders on specified dates, to create hypertext links between records, and to store specialized information in separate files, you have a couple of exceptionally useful programs. Of course, as I said earlier, no one program can meet all the text storage and retrieval needs of most academics. Although either of these programs could be used to take and store research notes, neither is suitable for bibliographical material because neither lets you sort your records into alphabetical order.

For research notes and bibliographies, my personal choice has been WordPerfect's *Notebook*, a simple, inexpensive, but enormously flexible program. Its main advantages for me have been its ability to handle and sort foreign-language text and its almost seamless interface with *WordPerfect*. Using WordPerfect Corp.'s *Office* integration program, I can keep *WordPerfect* and two copies of the *Notebook* in memory at the same time so that I can readily switch back and forth and transfer text among my article or paper text, my research notes and my bibliography file. It is also small enough (less than 100K) that I can easily use it for note taking on my hard-diskless laptop.

For the past year or so, I have also been using *askSam* from askSam Systems for bigger, more complex text handling jobs, such as juggling texts for a critical edition. It runs beautifully on my laptop and is compatible with WordPerfect's *Office* and *Library* programs, which enables you to transfer text between it and *WordPerfect* via the clipboard function.

To get the most from *askSam*, however, you must learn to program it. Although this takes a certain effort, it is well worth the trouble if you have heavy-duty text storage needs. You may not be able to find one single program that does all you would like it to, but, once you try out a couple of these text-oriented databases, you just might toss out your old shoe box and stop sticking colored rectangles on your desk lamp.

* * * * * * * * * * *

Ian M. Richmond is a member of the Department of French at the University of Western Ontario, London, Ontario, Canada N6A 3K7. He may be reached at 519-661-2163, Ext. 5703 as well as BITNET: IMR@UWOVAX.BITNET

TEXTechography

Arthur A. Molitierno

Abbreviations:

Countries

CN	Canada
CZ	Czechoslovakia
FR	France
JP	Japan
NL	Netherlands
SW	Sweden
UK	United Kingdom
WG	West Germany

Terms

DTP	Desktop Publishing
np	no page
n.	number

Months

JA	January
FE	February
MR	March
AP	April
MY	May
JE	June
JL	July
AU	August
SE	September
OC	October
NO	November
DE	December

[*Editor's Note:* TEXT Technology *welcomes bibliographic items of interest from the entire community of text producers. Issue 4 includes a representative sample from* Tokyo PC Newsletter, *published in English, courtesy of Peter Evans of Hosei University. Many thanks to Peter.*

Issue 4 also introduces a new feature for TEXT Technology. *Subsequent issues of TEXTechography will be published in a different style, according to associations or individual journals. This issue features the style of the American Anthropological Association, which is also the style of the publication,* American Anthropologist.

Although there will be some notable differences from the original style (italicizing the names of software or journals, for instance), "TEXTechography" will present the featured style as close as possible to the original's appearance. In the present case, the style omits all underlining, italicizing, and use of quotation marks in the citation proper.]

From the *Tokyo PC Newsletter*

Armour, A. 1991 — An Alternative to VD. Tokyo PC Newsletter [JP] 8(1, JA):19. [on *Reformat*, IBM program to change line length in text files]

Bechstein, W. 1990 — Writers' Little Helpers. Tokyo PC Newsletter [JP] 7(7, JL):5-10. [review article on IBM-standard software: *Funk & Wagnall's Standard Desk Dictionary*, *ABC Word*, *Writer's Handbook*, and *Associated Press Stylebook*]

Braem, A. 1990 — Translator's High. Tokyo PC Newsletter [JP] 7(3, MR):3. [on the sensation of word processing over a long period]

Evans, P. 1990 — Nobody Gets a Free Upgrade. Tokyo PC Newsletter [JP] 7(10, OC):3. [*XyWrite* upgrade from 3.55 to 3.56]

1990 — Tongue in Chip. Tokyo PC Newsletter [JP] 7(3, MR):10-12. [review of *Racter*, an interactive text-generating program, with tips for its use]

Fischer, C. 1990 — Words Through Windows. Tokyo PC Newsletter [JP] 7(5, MY):14-15. [review of *Ami Professional* and *Word for Windows*]

Gray, M. 1990 — Veuillez agréer, Monsieur, l'expression de mes sentiments dévoués. Tokyo PC Newsletter [JP] 7(11, NO):13. [*French Assistant*, English-to-French translation program for IBM]

Hochner, P. 1990 — Rapid Reading. Tokyo PC Newsletter [JP] 7(9, SE):8-9. [in French, review of *Speed Reading Tutor IV* (for IBM)]

Hodgson, W. 1991 — Memories Is Made of This. Tokyo PC Newsletter [JP] 8(1, JA):16-18. [review of *Memories!*, autobiography-writing program for IBM]

Parry, D., et al. 1991 — *Ventura* and *PageMaker*, *GEM* and *Windows*: A Symposium. Tokyo PC Newsletter [JP] 8(3, MR):28-30.

Alexander, G., J. Butler, P. Dyson, S. Edwards, D. Gertler, M. Moss, and M. Walter
1990 *PageMaker* 4.0, *DesignStudio*, *Xpress* 3.0 and *Ventura Publisher*. Seybold Report on Desktop Publishing 4(10):3-23. [four desktop publishing packages compared, includes designations for hardware requirements, features for editing and composition, with additional topics such as pagination and graphics performance]

Altman, R.
1991 Getting Smart with Ventura. Personal Publishing, MY, 20-24. [tutorial on various aspects of *Ventura 3.0* which runs in GEM plus Microsoft *Windows 3.0*, OS/2, and Macintosh; article includes productivity enhances for the newcomer to Ventura: cutting scroll and redraw time, using the underlying page, determining one's own defaults, loading multiple files together, tagging frames with color, using the Windows recorder, including original files with chapters, and using function keys]

Beckert, B.
1990 Technical Office Tools. CAE 9(12):68, 70. [overview of electronic mail and EDI (electronic data interchange)]

Bishop, P.
1991 Back to the Future. Personal Publishing, MY, 14-19. [overview of digitizing tablets, their styles, features, functions, advantages (replacing mouse with more natural feel for the artist), and drawbacks (pain in the wrist); includes comments from avid users; includes description of 4 tablets, plus information on price and manufacturer: *Kurta IS/ADB*, *SummaSketch*, *Wacom SD-420*, *Wiz*]

Churbruck, D.
1990 Desktop Color. Forbes 146:238-40. [comparison of output from Scitex and Mac for desktop publishing, with emphasis on color; includes information on scanners and programs such as *PhotoMac* and *Visionary* for publishing in-house; Scitex appears to have excellent compensation for color differences between video and print output]

Cohen, S. D.
1990 *FullWrite* Professional: Capable Wordprocessor for Macintosh. Electronic Library [UK] 8(4):285-86. [indicates many of the most common features of this wordprocessing package for Macintosh]

Crabb, D.
1990 The Place to Be for DTP: Is the Mac Really the Best Machine for Desktop Publishing? BYTE, SE, 87-89. [positive review of the ease of desktop publishing on the Mac, indicates high-resolution drawing and not page-layout or WYSIWYG is the most attractive feature; article includes discussion of *PageMaker*, *Quark Xpress*, and *On Location* (desk accessory for Mac)]

Diehl, S., and H. Eglowstein
1991 Tame the Paper Tiger. BYTE, AP, 220-226, 228, 230, 232, 234, 236, 238. [review of 14 OCR--optical character recognition--tools for converting printed text into electronic media]

Franke, H.
1990 Computer Art at a Glance. Chip [WG] 6:10-15. [in German, general article on the complete gamut of computer-generated art and images, including animation and printing]

Gorgens, A., and B. Kuhn-von Burgsdorff
1990 Multitude of Fonts. Personal Computer [WG] 7:102-03. [in German, reviews *Outline* 3.0, program to be used with wordprocessing programs to enhance the variety of fonts, along with such features as slanting, shading, and other special effects]

Harvey, D. A.
1991 Catch the Wave of DIP [Document Image Processing] BYTE, AP, 173-74, 176, 178-80, 182. [discussion of DIP, Document Image Processing, to reduce reliance upon paper; DIP includes input through a scanner, which is then recorded on optical media, with output through video displays, printers, or fax machines; costs of $25,000 per year for a 4 drawer paper file cabinet, plus $2160 to maintain the cabinet, make DIP an economically attractive alternative toward a paperless office]

Kuhn-von Burgsdorff, B.
1990 Tools for Publishing. Personal Computer [WG] 6:126, 128. [in German, review of 2 tools for desktop publishing with Ventura Publisher: 1) *VP Tools* 3.0 for print output; 2) *VP Saddle* for page layout]

Langley, F. W.
1990 Compiling an Old French-English Dictionary. University Computing [UK] 12(2):61-64. [discussion of the formation of CALATA--computer assisted linguistic and text analysis--in constructing an Old French (up to 1350) dictionary in 1 volume]

Lininger, S.
1991 PC Font Wrestling. Personal Publishing, MY, 38, 40. [introduction to loading fonts and making the fonts work with printers and software on IBM and compatibles]

Nielsen, J.
1990 The Art of Navigating through Hypertext. Communications of the ACM 33:296-310. [explanation of various means of navigating through the hypertext environment, including "timestamps" and "footprints" which provide contextual information]

Nishita, T.
1990 Present Situation of Education and Research on Computer Graphics in America. Journal of the Institute of Electronics, Information and Communication Engineers [JP] 73(2):171-73. [in Japanese, overview of computer graphics research in United States as it applies to calculation, educational training, commercial applications, with indication of present research trends]

Okadome, T., and H. Yamada
1990 A Comparative Study of Input Methods for Japanese Text Typing. Computer Processing of Chinese and Oriental Languages 4(4):275-94. [description of two experiments regarding transcription and description typing, with the conclusion that kana-to-kanzi causes decline in productivity and increase in stress]

Peyton, J. K.
1990 Technological Innovation Meets Institution: Birth of Creativity or Murder of a Great Idea. Computers and Composition 7:15-32. [discussion of real-time network as it applies to writing in the classroom, with examination of two case studies]

Pfeiler, K.
1990 Post-modernist Printing. Personal Computer [WG] 7:46-48. [in German, description of hardware and software for HP Laserjet Series II printers which allow them to work with Postscript language: *Hanzon Stellar PS* and *Destiny Pagestyler II*--both products contain circuit boards, expansion boards for the printer, cables connecting the two]

Quraishi, J.
1990 *HyperPAD 2.0* and *SmarText* Offer Two Different Metaphors for Creating HyperDocuments. PC Magazine, NO, 46. [review of two programs for creating hypertext documents: 1)

HyperPAD, character-based, requiring 640 K of RAM, uses extended memory and pull-down menus; 2) *SmarText*, *Windows* based, almost automatic creation of hypertext documents]

Rasmus, D. W.
1990 Guide 2.0 [hypertext program]. MacUser, AP, 70-71. [software review of *Guide 2.0*, Owl International's hypertext program for the Mac; creates links in text files for cross referencing; uses buttons (words that hide other words), notes (windows containing additional information), reference points (for getting from one section of text to another); program does not support multiple fonts and styles in single text mode, no automatic indexing available]

Reynolds, C. (reviewer)
1990 Review of *Hypertext Hands-On! An Introduction to a New Way of Organising and Accessing Information by* Ben Schneiderman and Greg Kearsley. New Scientist 126:83.

Ricciardi, S. P.
1990 Two Low-cost Boards Offer Graphics Workstation Power. PC Magazine, AU, 44. [review of 1) Hercules Graphics Station Card and 2) Desktop Computing AGA 1024, boards supporting VGA 800 x 600, including drivers for *Windows* and *AutoCAD*]

Robinson, P.
1990 Color on the Page. MacUser 6:see 56-69. [a guide for selecting color printers for the Mac for use in desktop publishing, for artists and designers; explanation of 1) dithering (mixing of colors) for output of numerous colors, 2) RIP, raster image processor; indicates Tektronix *ColorQuick* as the top-rated color ink-jet printer]

Rosenberg, S.
1991 Corel Draw Ups the Ante in Windows-Based Graphics. BYTE 16:289-90. [review of *Corel Draw* 2.0, capable of producing 3-D images, indicating ease of use and intuitive interface which make it more desirable than Micrographix's *Designer* 3.0]

Ryan, B.
1991 Coping with Diversity. BYTE, AP, 257-60. [discusses possible routes for connecting files between a number of different computers and operating systems: IBM, Macintosh, Atari ST, Amiga, Apple IIGS]

Sproat, R., and C. Shih
1990 A Statistical Method for Finding Word Boundaries in Chinese Text. Computer Processing of Chinese & Oriental Languages 4(4):336-51. [statistical grouping of Chinese characters into two-character words, based upon some 2.6 million Chinese characters in newspapers]

Van Tyle, S.
1990 A Whole New Way of Using Computers...[hypertext]. Electronics 63:70-71. [3 major developers of hypertext (Nelson, Engelbart, and van Dam) indicate the future of this new computer technology: accessing information from large databases, sharing of information, using computers as the major means for sharing information]

Wallia, C. J.
1990 Review of *Microsoft Word* 5.0. Computers and Composition 7(2):77-78. [positive review of features, indicating *Word* 5.0 is an embryonic desktop publishing program]

New Subscriber Information

To academic and corporate writers and teachers of writing, *TEXT Technology* brings analyses of microcomputer hardware and software, discussions of programming techniques (both in languages and in applications), book reviews, updates of significant events in computing around the world, bibliographic citations, and much more.

TEXT Technology, created by the editor of the *Research in Word Processing Newsletter*, also will become an information clearinghouse for subscribers' opinions and queries about personal computing during the '90s—and beyond.

Subscription rates for one year (6 bi-monthly issues—16 pages) of *TEXT Technology* are as follows:

US $20
Canada $27
Foreign $35

All prices are in US funds

Please fill out a copy of the form below and send it with your check or purchase order to

Subscriptions Department
TEXT Technology
Wright State University—Lake Campus
7600 State Rte. 703
Celina, Ohio, USA 45822-2921

Name _____

College or Corporation _____

Street _____

City _____

State or Province _____

Zip Code or Postal Code _____

BITNET _____

Affiliation: ACH ____ ALC ____ ALLC ____ 07/01/91