ICON: A CASE STUDY IN OFFICE AUTOMATION AND MICROCOMPUTING

Ronald M. Lee

December 1980
WP-80-183

INTERNATIONAL INSTITUTE FOR APPLIED SYSTEMS ANALYSIS
A-2361 Laxenburg, Austria

ACKNOWLEDGMENTS

I am especially grateful to Alan Krigman and David Ness, key characters in the case to follow, for their *detailed* comments and criticism of an earlier draft of this paper.

TABLE OF CONTENTS

ICON: A CASE STUDY IN OFFICE AUTOMATION AND MICROCOMPUTING

Ronald M. Lee

INTRODUCTION

The following is a story about a small company in Philadelphia called Information Concepts, Inc., or, more colloquially, ICON.

ICON's business is doing technical writing on a consulting basis for various firms in the production instrumentation and computer industries. ICON also serves as editor for several trade magazines and newsletters relating to instrumentation and process control, as well as providing market research studies for this industry. The basic product of ICON is therefore information—in the form of *text*.

The focus of this study is on the technologies ICON uses to produce this text. Over the past four years, ICON's operations have shifted from being basically typewriter-based to one that now uses among the most sophisticated of word processing and office automation technology.

The purpose of this study is to examine the factors which enabled and encouraged this evolution. Of particular interest is the fact that ICON is an extremely small company, ranging variously from five to ten people, and so had very little capital to spare. Thus this remarkably rapid development took place under severe budget constraints, with payback periods ranging from six to 18 months.

As in many case studies, one of the most important aspects is the personalities involved. In this case, the central figure is the president of ICON, Alan Krigman. Several other major actors were William Latimer, then director of computer services at the Wharton School in the nearby University of Pennsylvania; David Ness, Professor of Decision Sciences and Management at this same school, and myself, then a doctoral student in Decision Sciences. Thus, another theme throughout this story will be the successful interaction between a pragmatic (though open-minded and not risk-averse) businessman and members of a research-oriented, academic environment.

Obviously, because of my personal involvement, this account will not have the tone of indifference and objectivity that case studies are supposed to have. On the other hand, I think the biases thus introduced are in this instance offset by my direct familiarity to many of the personality factors and events that influenced this firm's development.

Al started ICON, Inc. in 1973, after serving some five years as a trade journal editor at a major publishing house.

I met Al rather coincidentally. Another of his financial interests is in real estate around the university area and, upon arriving in Philadelphia, I ended up renting an apartment from him.

On that encounter, I mentioned I was joining the Decision-Sciences department and, during subsequent conversation, I happened to describe some of the word processing software in use at the Wharton School. Al became interested. Some ten years earlier, he had done some programming in FORTRAN II, but for the intervening time had no contact with computing. His image of computing was thus mainly of rather narrow engineering applications, and hadn't really considered that machines could be of use in handling text.

So, one Friday I borrowed a portable terminal from the department, and let Al use it for the week-end. I started him out with about a half-hour tutorial on the use of the text editor and document formatter, and left him some additional introductory documentation.

Two days later, when I next saw him, I realized he was hooked: he had pages of questions for me about moving text around, making footnotes, etc. etc. Indeed, for the weeks to come, he would speak of nothing else.

Like many successful business people, Al is decisive and moves quickly once he has made a decision. In the week following, Al began calling time sharing houses and terminal suppliers. Soon he had narrowed down the choice of timesharing vendors to two: Bowne Inc. and the Wharton School. (Wharton sells a limited amount of time to local commercial users.)

Both sources offered substantial word processing software. Bowne had an advantage over Wharton in that they dealt only with commercial customers and so had established and well developed customer services.

At Wharton, on the other hand, usage was mainly for internal purposes—i.e., for the various department offices, and by graduate students and faculty. Thus, documentation was not as well developed. Also, because much of this software was developed as part of ongoing research in office automation, the software was not necessarily as stable as that at Bowne.

On the other hand, while experimental, Wharton's software was by the same token more sophisticated than Bowne's. This software had been developed over several years by faculty (chiefly David Ness) and graduate students as part of Wharton's Office Automation Project (OAP). It thus included numerous features and options that a commercial enterprise like Bowne would find overly complicated or fanciful.

Price-wise the two services were comparable. Bowne was slightly more expensive on an hourly basis, and had a minimum charge of $300 per month, while Wharton had no minimum. It was this last fact that convinced Al to choose Wharton: he assumed that he would not have enough usage to exceed the minimum. As it turned out, this was the right choice for the wrong reason.

For hardware, Al bought a Delta Data Systems CRT for $1300 and a floor-sample daisy wheel printer for $3500, both used. The printer produced high quality output, so that finished documents could be done on the machine.

The immediate impact of this equipment was on the turn around of document drafts. Al, the principal writer of ICON, would often go through numerous drafts of a given report. Previously, this meant that a secretary had to entirely re-type the report each time, which for longer reports often meant a several day delay.

Al taught his secretary enough about the system to permit input of documents with minor proofreading. He had originally intended to proofread printed copies, but started editing at the terminal as a means of getting started. This meant that to use the system Al had to invest his own time in making corrections, etc. This might have been a problem because he can't touch-type and basically just uses two fingers. (His typing has since improved, but still isn't up to a secretarial level.)

Even with these limitations, Al found it efficient to continue doing work on the machine himself. The documents are of article length, i.e., usually from two to ten pages. The key difference between computer word processing vs using manual typewriters is of course that keystrokes are never repeated; i.e., one types the original text and subsequent modifications only once.

Thus, the usual routine in preparing a document was that having a draft re-typed with revisions usually imposed an interruption of at least one to two days, often longer since the typists had other responsibilities as well. Thus the continuity of the paper's development would be broken.

With the word processing system, this problem was vastly improved. Most changes to the documents involved typing relatively little text—e.g., they were changes in wording or re-arranging parts of the text. These Al could do himself in a fairly short time and print out a final copy. Thus turnaround on revisions was reduced from several days to one to two hours.

Other people were soon trained to use the word processing system. Everyone in the office now does at least some original text entry, and, as appropriate, text correction.

The reason for having individuals make certain corrections themselves is that it is often easier to do so than explain it to others. Also, working on the CRT, a writer could try out a certain wording or sentence or paragraph restructuring, and if he didn't like it, change it again on the spot.

Good writing, it should be remembered, typically involves a great deal of revision and tuning. Thus, the importance of this immediacy of response was especially important in Al's case.

SUBSEQUENT DEVELOPMENT

Within several months it became clear that another CRT would be needed. This time Al bought a new Datamedia, which had a better designed keyboard than the DELTA CRT..

Basically, this second terminal was used by secretarial personnel to input original texts, while Al kept the first terminal in his office for making corrections.

By this time, ICON had become one of the major word processing users at Wharton. Furthermore, Al was using this software in a much larger variety of ways than most other users. Thus, he was often the one to discover the need for improvements and extensions. Also, as a user in a production environment, Al had a more practical view than the typical academic user.

Most of the word processing and other office automation related programs had been written by David Ness. Among these was an electronic mail system, and an adaptation of it for reporting bugs and other comments about particular programs.

So it was that Dave began getting electronic mail messages from a certain A. Krigman whom he had never met but who seemed to be using the software to its limits judging from the suggestions.

I, of course, knew both parties and suggested they meet—though this didn't happen for a couple months. However, by that time they had already become well acquainted through electronic mail. Also during this period of increasing use, Al also began to think of other, non-text related computer applications for ICON and his real estate management activities (which were run out of the ICON office). Through his numerous interactions with the Wharton computer center, Al had become acquainted with Bill Latimer, its Director. An MBA graduate from Wharton, Bill was as well a competent programmer, and Al was able to recruit him on a consulting basis to develop some of these other applications. This work included preparation of name and mailing directors for clients and a system to manage the accounting and billing of Al's rental properties. More importantly for the theme of this study however, Bill, by virtue of his activities as computer center director, was well acquainted with a large variety of hardware and software vendors and products. It was Bill, therefore, that provided Al with an introduction to the computing marketplace. Well beyond his consulting involvement at ICON to develop software, Bill continued to be an important influence in notifying Al of new products and innovations, advising him of the quality of dealers and helping him in purchase decisions.

DAVE NESS AND MIDDLE-OUT RESEARCH

Dave Ness came to Wharton in 1973 after having been on the faculty of the Sloan School at MIT. Nearly anyone acquainted with him from either school would acknowledge him to be one of the most prolific programmers they have met.

A year or two after coming to Wharton, Dave became interested in office automation; i.e., to develop improved technology to assist office staff and managers.

Obviously, one of the key problems in any research area is methodology. In the area of office automation, methodology is particularly problematic. Clearly, a person like Dave has the ability to invent all sorts of software--but the question is, what do office staff and managers need? Because these people don't have a perspective of the technological possibilities, it is difficult for them to make useful suggestions. Conversely, a technically trained academic has little feel for the day to day problems these people face.

One alternative to this situation is a "bottom-up" approach of building bits and pieces of software to cover small, well structured tasks, later hoping to synthesize these into larger units covering more substantial problems.

Another approach might be characterized as a "top-down" one: doing a careful and comprehensive study of what activities these people perform, where the problems are, and what an ideal office automation package would be.

Dave criticizes the first approach, in that the tiny building blocks never seem to fit together to form more substantial systems. The top-down approach he also finds deficient because the analysis never is complete, and so the invention of new technology never begins. Dave's notion of a "middle-out" approach is something of a compromise between these two, but also adds the additional dimension of *evolutionary design*. Dave's view is to build a system for some "middle size" office related task based on the designer's best guess of what is needed. Then, put it in an office environment and see how people use it, and *collect their reactions and suggestions* once they have used it. Then go back and modify the system to meet these suggestions and once again see what the users think. The key element here is that the user becomes an active agent in the design process. (Clearly, to use this as a research strategy, one has to be a competent programmer.)

At Wharton, Dave basically has had four user groups to use his OA software.

The first of these is Wharton Word Processing--the typing pool for the Wharton faculty and administration. Here the usage is mainly for preparation of academic papers and administrative reports.

The second is the Dean's office at Wharton, where the usage is to prepare more standardized documents such as faculty publication lists, Dean's mailings to alumni, etc.

The third is the departmental office of the Decision Science department. Usage by this group is somewhat more varied, including scheduling programs, electronic mail, etc.

The fourth user group is the faculty and students at Wharton. Because there is no course in the use of word processing or the other OA software, this usage was initially confined to the Decision Sciences and Management departments (where Dave teaches), though faculty and doctoral students from other departments are now picking it up as well.

Al's appearance on the system thus offered Dave a different type of target user: a commercial writing professional. Of special interest was the fact that Al had begun to use the system as an integral part of his writing activities, rather than simply use it as a speedier replacement for a typewriter.

MOVE TO MICROCOMPUTERS

While an energetic and imaginative businessman, Al is also extremely cost conscious ("cheap" has been used to describe his approach to investment).

In its usage of the system, ICON had long since exceeded the would-be $300/month minimum. The typical usage was now around $800/month for text processing—and sometimes more than $2500/month with related tasks considered.

While each advance in expenditures was in Al's mind cost justified, he began to wonder whether he could somehow do better. So he began to look into microcomputer technology.

With typical thoroughness, Al, with the help of Bill Latimer and Dave Ness, investigated the various vendor literature and local outlets. After several weeks of investigation, Al decided to buy a CROMEMCO Z-80 microprocessor and dual floppy disks. This machine supported a CPM operating system, COBOL, FORTRAN, BASIC (later PASCAL, C, etc,) and most importantly a text editor very similar in syntax as well as power to the TECO editor ICON had been using on

Wharton's DEC-10. The micro also had a document formatting program which was adequate, though not as powerful as the RUNOFF package on the DEC-10.

Al had originally intended to move about half of his work from the DEC to the micro and use telephone communication to exchange files between the machines. But the phone link did not work as planned, so he decided to try and see how much could actually be done on the micro. Within a short time, all routine work in document preparation was being done on the small system. Only the real estate package and some special text capture routines continued to run on the DEC 10. About 80% of the load was therefore transferred in-house. Roughly, this was a savings of about $600-800 per month. Since the terminal and printer which had been purchased for time sharing purposes would work with the new equipment, the only incremental costs were the micro computer itself and a dual disk drive, which together cost about $6,400, so at a conservative estimate the micro paid for itself in about 10 months.

This in itself certainly made the micro a satisfactory investment. However, there was another side benefit that turned out to be decisive in ICON's technical development: the BASIC language. The Wharton machine of course contained a wide range of language compilers and interpreters. However, most of these were oriented towards quantitative applications--e.g., statistical packages, FORTRAN, APL. (A primitive version of BASIC was also available on the DEC 10, but was clumsy and limited in capability and so seldom used.)

For the text-oriented applications in the OA work, the language used was the DEC assembly language, MACRO. This, unfortunately, requires considerable sophistication and detail level understanding of the machine to use effectively. So, while using the DEC, Al was never tempted to try to write any programs himself.

The market for microcomputers has been largely from hobbyists, or at least non-professional programmers. Thus BASIC, a very simple language, has become popular on these machines and the commercial versions have come forth with useful extensions, which still maintain the simplicity of the language.

Coincident to this period, Al had hired an editorial assistant, Ms. Leslie Tierstein. Leslie had a graduate degree in linguistics, and as part of that work had done some modest amounts of computer programming.

Thus, some time after the arrival of the micros, she and Al tried writing some programs.

One of the earliest attempts was a letter writing program. The idea for this came from the dissertation work of Michael Zisman, then in progress at Decision Sciences, which contained several such programs among the modules of a larger OA system.

The letter writing programs at ICON were extremely simple yet effective. The program simply prompted the user for the name and address of the recipient and the text of the letter. The program's task was basically only to provide the proper formatting commands--e.g., how far to indent for the date and closing, proper vertical spacing, etc. The recipients address was also copied on a separate page, for the envelope. This text was then output as a file and passed to the document formatting program.

Few programmers would be impressed by the technical sophistication of this application, yet, even for these inexperienced programmers it was relatively easy to do and it solved a small but persistent problem—the cosmetics of letter layout.

While Al soon became able to program in BASIC, Leslie became the primary programmer. A multitude of other small applications programs were developed in the succeeding months. Again, each was not particularly sophisticated or complicated from a programming standpoint, but each nonetheless addressed some specific need. for instance, a common type of program is one to assist in data entry--e.g., for a numeric report or table. Such programs were often developed for just a single use. Other programs were developed to manage mailing labels, format statistical tables, etc.

The experience at ICON with micro-computers thus shared many of the advantages argued for them in other organizational contexts. In particular, since computer time on an owned, personal machine is essentially free, one is not greatly concerned with the efficiency of program operation, and hence inefficient but easy-to-use languages like BASIC are feasible. This in turn enables non-expert programmers to develop their own application programs, which has the particular advantage that people *more familiar* with the application *problem* can do the programming.

MULTI-LEVEL TECHNOLOGIES

As mentioned, most but not all of ICON's text processing work could be moved to the micros. Some projects, e.g., a mailing lists of several thousand names, a 300-page book, were too large scale to be stored on the small diskettes.

On the other hand, Al realized, the data entry for these projects could just as well be done on the micros if only the telephone lines could be implemented so data could later be moved to the larger machine.

The preference for doing work on the micros was several-fold. First, of course, was that time on the micro was free whereas that on the DEC-10 had a per-hour charge. But also, the micros had a more comfortable interface. In using the DEC-10, the terminal used an acoustic couplers with a line speed of 300 baud (30 characters per second). With the micro, however, one could run at

1,200 or 9,600 baud. While this doesn't make much difference for straightforward data input, it is a tremendous advantage when editing a text where you often want to print out the preceding paragraph, etc.

But perhaps the most appreciated aspect of interacting with the micro as opposed to the time-shared DEC-10 was *constant response time*. That is, while the DEC-10 was certainly much faster in execution, the response time felt by the user varied depending on how many other users were on the system. When inputing and editing text, this can be very distracting.

Apparently in such interactions, one builds up a certain *rhythm* in exercising the editing commands. When this rhythm is broken by delays in the system, one's attention is interrupted from the text contents to the system itself. While the micro was in many cases slower than the DEC-10, it was at least consistent in the time it took, allowing the user to maintain this rhythm.

Thus there was an interest to keep as much of the interactive computing as possible on the micros, and use the DEC-10 in a batch mode for the heavier processing. Some thought was given to getting a floppy disk reader for the DEC-10 so that the data could be entered locally at ICON and hand carried to Wharton to be read in. However, the more economical alternative that was adopted was to improve on telephone interface for the micro so that it could dial out and appear to the DEC-10 as a terminal.

Al had by this time built up a good working relationship with Mr. Tom Dinella, owner/operator of the Computer Store where Al purchased his micro equipment. Tom knew the micro hardware and operating system software in detail, and already had done some minor systems programming for ICON. It was thus Tom who developed the first "transceiver" program that allowed the micro to access the DEC-10. Basically, this program would call the DEC-10 and enter the login commands. At that point, the user's view was exactly as if he/she had simply called in using an ordinary remote terminal. However, by using a special control character, the communication would switched back to the micro and similarly back again, special commands were also included for sending files from the micro to the DEC-10 and vice versa. A limitation was that these transmissions all took place at 300 baud, so they were rather slow--hence they were usually left until after hours. Nonetheless, the method proved effective and became a regular practice at ICON.

One of Al's responsibilities was to serve as editor of INTECH, the trade magazine of the Instrument Society of America (ISA). As part of this contract, ICON received access to the Lockhead DIALOG system, a bibliographic database. DIALOG is designed to print out bibliographic references at the user's dial-up terminal. However, with the above described feature of contacting other machines through the micro, Al was able to obtain these bibliographic references at the terminal, and print only those of interest.

This enabled Al to prepare retrieved collections in a form more directly useful than ordinarily possible. Indeed, one of the regular features of the INTECH journal, the text for which was all done on ICON's micros, was a listing of references on some special topic taken from DIALOG.

NANO-COMPUTING

As mentioned earlier, Al interacted frequently with Dave Ness regarding changes and further directions of development for the text processing software at Wharton.

Dave, until that time, had worked primarily on the DEC-10. However, when Al made the move into micro-computing, Dave too became curious about its possibilities. Because of their now strong working relationship, Al gave Dave a free hand in using and experimenting with ICON's micros. Dave in turn was impressed with the power and flexibility of the small machines, especially for text processing applications, and began to explore the range of languages and other software available for the CPM operating system supported by the micros; e.g., languages like PASCAL, C and a variety of text editors and document formatters.

One thing that served to seal Dave's cooperation was another result of Al's entrepeneurialism. Seeing the developments emerging at ICON having applicability to other small firms of comparable characteristics, Al proposed starting a separate company to develop and market this technology. This company was later called Office Automation Concepts (OAC). The investors in this company were Al, Dave, and Gerald Hurst, another faculty member in decision sciences and mutual friend of both Dave and Al. The strategy was that applications would not be developed for ICON specifically, but rather for more generalized uses, for which ICON would serve as a test bed. To get things rolling, Al arranged to sell a packaged text processing personal scheduling system to the main office of the ISA *(op cit* Instrument Society of America). located in Pittsburgh. Dave's role was to develop a personal scheduling system for the micro.

A further step was taken when Dave decided to buy a microcomputer of his own. This was an Exidy "Sorcerer." While it had a comparable size memory to the Cromemcos at ICON, it did not have a CPM operating system nor floppy disks. Language interpreters (e.g., BASIC) were called up by inserting a plug-in ROM (read only memory). Programs and data were stored on an ordinary cassette tape recorder.

Cost-wise, this complete system was about $1,600, including the CRT screen. The micro computer itself is built into the keyboard.

As may be apparent from this description, the Sorcerer is basically designed as a stand-alone personal computer, e.g., for small scale calculations or computer games.

Dave however saw it is potentially more useful as a smart terminal, i.e., for communicating with other machines.

One of the attractive features of the Sorcerer is its flexible design. For instance, by adding a so-called "extender box," one can add a variety of additional hardware—e.g., disks, clocks, music, voice generators, etc.

Also, one is given a great deal of software control over the machine. For instance, the CRT screen image is a "memory map," i.e., each screen position corresponds to a location in memory as opposed to being a serial transmission. Thus, not only is the screen speed literally instantaneous, but several alternative screen images can be maintained in memory, and switched back and forth by program software.

Dave took advantage of these features to modify his Sorcerer to be able to call up the DEC-10 and interact with it, similarly as had been done with ICON's micros. (While said in a few words, this is no mean feat.) Thus, after loading his "terminal program" from cassette tape, Dave could type the command "DEC-10," and the terminal would proceed to dial the phone and execute the login protocols of the Wharton machine--i.e., account number, password, etc.

Dave later refined this procedure by investing in a "PROM burner" (PROM = programmable read only memory), which allowed him, effectively, to make his own ROM cartridges. Thus, to turn the Sorcerer into a terminal, one simply plugged in the memory cartridge.

It is to be remembered that Dave is a premier programmer, capable of working feverishly and who understands computing at a detailed level. He is the major reference person at Wharton for questions about applications programs or the operating system. He was thus able to exploit the capabilities of his home machine in ways that few hobbyists could.

One major example of this was that Dave wrote an assembler for the Sorcerer, *on the DEC- 10*. He could thus make use of the robust software on the Wharton machine to edit/assemble Sorcerer programs and then "down-load" them to his home machine.

Recall that the micros at ICON had a call *out* facility so they too could communicate with other machines. The next step in this development was to develop a call *in* facility for these machines. This was another of Dave's inventions. After that, Dave could thus call up the micros at ICON using his Sorcerer at home.

Two other of Ness' developments deserve mention. One was what he called "Label-Basic." This was a pre-processor that would convert a slightly abstracted BASIC syntax to that required by any of the DEC-10, the Cromemco micros or the Sorcerer.

More consequential, as suggested by several extended discussions with Al, Dave took the object code for the text editor, TECO, (versions of which were used on the ICON micros—on the DEC10), and to "dis-assemble" it, through a series of iterations of substituting symbolic names, etc. He used this as a model to create a new version compatible with the Sorcerer and later burned it too onto a PROM. Thus, the SORCERER could then be used as a stand-alone word processing machine, at a cost of roughly 1/3 that of the Cromemco micros. The same version of the editor was later substituted for that purchased for the Cromemco system.

Dave and Al's thinking was thus: Much of the time spent on the micros was for simple text entry and minor correction. This was primarily an in-core task, requiring disk access basically only to make back-up copies. By using the Sorcerer's for this purpose, ICON could have three work stations for about the price of one micro+CRT. ICON has thus bought three Sorcerers for this purpose.

Thus, most original text entry is done on the Sorcerers. Also one of the micros uses a Sorcerer as its CRT. Thus, after the text has been entered on one of the stand alone Sorcerers, it is dumped onto cassette tape. The tape is then taken to the Sorcerer connected to the micro, read in, and transferred to floppy disk. Alternatively, using the Sorcerer terminal program, text can be transmitted via phone from remote locations.

There are thus now three scales of technology now in use at ICON:

"nano" level technology	—	i.e., the Sorcerers
"micro" level technology	--	the Cromemco Z-80 micros+CPM floppy disk operating system
"mini" level technology	—	i.e., the Wharton DEC-10, used on a time-sharing basis.

The technological development at ICON is of course still continuing, and it will be interesting to watch its future growth.

CONCLUDING REMARKS

This has been a brief account of how a small, closely held company went from a completely manual mode of office operations to not only a highly automated one but actually to the level of making new innovations in this technology. Moreover, this development took place in a span of less than four years, and always with very limited capital. One very important factor, I think, was the personalities involved. As can perhaps be felt in the previous pages, Al is an extremely gregarious and outgoing personality. On the streets around his office he knows literally everyone. Likewise, he has become better known around Wharton than many faculty are.

More importantly, however, he was able to build a working relationship with various academic oriented people that seldom emerges in ordinary consulting relationships. He was able to present his problems in a way that stimulated their research interests. ICON (and OAC) thus became the focus of creative, innovative talent that otherwise could not be bought by companies ten times as large as ICON.

But more than simply stimulating interest, Al also was tolerant and open-minded, accepting that academicians tend to be somewhat like children: they often do not stay fixed on the assigned objective, but wander into other ideas that prick their fancy. They work irregularly—often with great intensity, other times not at all. They are relatively unintimidated by deadlines. Few business managers, especially those running small companies with shoe string budgets and tight controls, can put up with this. Ergo, they end up buying off-the-shelf software and turn-key systems.

Correspondingly, the academicians that Al met were fairly applied in orientation. At Wharton there is a strong emphasis that research, while future oriented, must have some visible practical consequence, albeit perhaps on a fairly distant horizon. This is especially true at the Decision Sciences Department where Al made most of his contacts.

Another important factor in this case was timing. At the time when Al first came into contact with Wharton, micro-computing was just beginning to emerge commercially. Many new systems and software products were coming onto the market. However, the industry was (and still is) very young, lacking standards to aid compatibility in its products and largely populated by small engineering oriented manufacturers unused to providing customer support and assistance and often lacking in financial stability. Mortalities were and are high. Thus, while the emergent technology was at the right scale for a small business like

ICON, it was nonetheless still risky for all but very standardized applications. The academic influence at ICON was what allowed it to overcome these risks, by providing counsel on hardware purchases and the expertise to create software to really exploit it in novel ways. In many cases, technical problems were overcome that a commercial software house would never venture to touch.

Lastly, a key enabling factor for all this was the coincidence of ICON's location near the university, hardly a 10-minute walk. Had ICON been located in another part of the city, this development would probably never have taken place. It was thus easy for the university people involved to drop in at ICON during the day and conversely for the ICON people to go to the university to pick up output, ask about a bug in a program, etc. Also, in addition to the people mentioned here, Al also hired various university students on a part-time basis, further reinforcing his connections there.

ICON is thus a happy example of an industry-academia relationship where both sides visibly benefited. Enabled by the coincidences of timing and location, I nonetheless believe the key factor was the attitude the various people brought to this relationship. If case studies can indeed have "lessons," I think that would be the lesson here.

REFERENCES

Krigman, Alan S., and Ronald M. Lee. 1978. "Automatic Text Generation: Commercial Application." Working Paper 78-07-02, Department of Decision Sciences, The Wharton School, University of Pennsylvania, July.

Lee, Ronald M., and Alan S. Krigman. 1978a. "Generating Semi-Structured Text: Representation." Working Paper 78-08-03, Department of Decision Sciences, The Wharton School, University of Pennsylvania, August.

Lee, Ronald M., and Alan S. Krigman. 1978b. "Generating Semi-Structured Text: Specification Language and Generalized Implementation." Working Paper 78-08-04, Department of Decision Sciences, The Wharton School, University of Pennsylvania, August.

APPENDIX A: AUTOTEXT

My own involvement at ICON was not so much in developing technology for the day-to-day business of ICON, but rather for various special projects and joint ventures with Al. Among the more interesting of these was a system we came to call AUTOTEXT.

My own research at Wharton involved applications of Artificial Intelligence to management. Thus, as a programmer, I had a reputation for working on rather "far fetched" problems like natural language query parsers, etc.

One day Al suggested a problem he thought might fit my interests.

Another of ICON's regular jobs was to produce a newsletter called Pollution Equipment News. This is a bi-monthly digest of announcements of new products for pollution control. The procedure for producing this tabloid magazine is as follows:

On an ongoing basis, vendors submit flyers, brochures, spec sheets, etc. about their new products to ICON. There, a person—call him/her the "writer"—assembles this material and condenses it to a one or two paragraph summary descriptions of the product.

These "profiles," as they are called, prove to be fairly structured in content. They contain the vendors name for the product, then its generic type, functions that the product performs, special features of the product, unusual applications if any, tolerances and other specifications, and finally ordering information.

On the other hand, there is also substantial variation from one product class to another--e.g., one may have a long list of features, the next may be interesting primarily for its unusual applications, etc.

The structure of these profiles is thus only "semi-regular." Indeed, the structural similarities between them are seldom apparent to the reader; and were only recognized by Al himself after writing and editing large numbers of these for several years.

The profiles had to be complete, yet as brief as possible. Also, since 200-300 of these may appear together in a single issue, a certain amount of purely stylistic variation is needed.

In order to write such profiles, the writer has to have a certain amount of technical expertise, to recognize what is important and interesting about the product. As well, the writer needs to have a certain degree of writing skill to meet the above editorial requirements. The problem, as Al put it, was that anyone smart enough to have both these skills was typically smart enough to do something more challenging and so quickly gets bored with the task with a consequent decline in quality.

Al's proposal was to automate the writing part of this activity--i.e., have the computer ask questions about the product and then compose the 1-2 paragraph profiles itself.

This, as it turned out, proved to be relatively easy from the computer standpoint. The main difficulty was in the specification--i.e., deciding what the computer should ask and, from the responses, what text it should generate.

After one or two additional meetings with Al, I had sufficient idea of what was needed to code a prototype program. I happened to write this in SNOBOL since I was learning that language at the time, though this was really a more robust language than was actually needed for this problem.

In this case as in many experimental applications, once Al had a working prototype, his thinking sharpened considerably as to how it should ideally perform. In the several weeks to follow, I made numerous modifications and extensions to the program. The end result was the first version of AUTOTEXT, which generated satisfactory pollution equipment profiles.

Not long after, Al began to recognize similar application opportunities in other areas. For instance, the aforementioned INTECH trade journal also had a section on new products. I thus ended up writing several variants of this original program for other similarly restricted subject matter.

Repetition of this rather straightforward programming task led me to consider whether the problem could be generalized. The core of this problem was to arrive at a specification language for describing the structure and style of any one of these writing tasks. Deciding on this language again involved numerous interactions with Al: I would propose a syntax, he would text it to see if it could describe the described textual patterns, offer criticisms and suggestions, etc.

Once we settled on a specification language, I wrote an interpreter for it, this time in LISP because of its recursive capabilities and flexible control structure. This too was on the DEC-10. The AUTOTEXT specification language and the LISP implementation are described in Lee and Krigman (1978a, 1978b) respectively.

Thus in this version, what we have since called "meta" Autotext, the editor him/herself describes the structure or "grammar" of the type of text to be generated. This is done originally in a graphical syntax we developed and then translated to a linear form.

The LISP program reads this grammar and, based on it, asks certain questions of the writer and subsequently generates the text.

As it turned out, this LISP version--while an elegant generalized solution to the problem--was used only once or twice at ICON. The problem was cost. Unlike most of the text processing software ICON used at Wharton, which are usually small, highly efficient programs, LISP is itself an interpreted language and consequently takes up a lot of core and CPU time--for which ICON had to pay directly.

This was for me a real disappointment . As an academic, I took pride in logically elegant solutions to challenging problems, but was not accustomed to recognizing cost constraints as part of that. To me it was the job of vendors, software houses, etc. to take the additional step of making such solutions commercially feasible and efficient.

However, in this situation this clearly was not going to happen. Motivated more by indignation than anything else, I managed to re-work the meta-Autotext program as a BASIC pre-processor for the micro machines. As a pre-processor, it accepted a similar grammar specification as its LISP counterpart, but rather than interpret this directly it generated another BASIC program which was then compiled and run.

This, as might be imagined, was a very large and complicated program, not at all suited to a simple language like BASIC. Indeed the program had to be broken into several parts to fit into the 64K core. A more appropriate language would probably have been either PASCAL or C, but these only became available sometime later.

This micro version of meta-Autotext has proved to be successful. It has since been used to create Autotext programs now in regular use for other periodicals.

Typical of Al's entrepreneurial influence, we are now taking steps to market this to other companies with comparable writing applications.

Our sales efforts to date have, however, met with definite customer resistance, which serves to highlight the open-minded atmosphere are ICON relative to other similar enterprises. Writing is generally considered to be an essentially human, creative activity and therefore not subject to automation. Despite the fact that these people repeatedly are unable to distinguish the computer-written profiles from the manual ones, there seems to be a deeply entrenched, almost moral skepticism about applications of this sort. There is, of course, no magic involved, merely editorial control carried to a mechanical extreme; in effect "meta-writing," specifying how an entire class of documents should be written. The actual writing following these rules, could as well be done by a human clerk with the same result.

For those interested, further discussion about AUTOTEXT and its commercial applications are available in Krigman and Lee (1978). Additional technical details are to be found in Lee and Krigman (1978a, 1978b).