Automation of Administrative Systems in a Law School Teaching Clinic: Designing a Computer System to Process Case and Time Data for Management and Research (with James A. Sprowl)

Ronald W. Staudt
IIT Chicago-Kent College of Law, rstaudt@kentlaw.iit.edu

Computerizing Client Services in the Law School Teaching Clinic: An Experiment in Law Office Automation

James A. Sprowl and Ronald W. Staudt

In an earlier article (1979 American Bar Foundation Research Journal *1), James A. Sprowl described a new computational processor that enabled attorneys to automate repetitive client-interviewing and document-drafting tasks. In this article, Sprowl and coauthor Ronald W. Staudt describe the first experimental use of the new processor to serve the clients of a law school clinic. In the experiment attorneys and secretaries were trained to design automated practice systems. The authors analyze the results of this experiment and draw some preliminary conclusions about the future usefulness of automated law practice systems.*

James A. Sprowl is Research Attorney, American Bar Foundation, and Lecturer, Northwestern University School of Law and IIT/Chicago Kent College of Law. B.S.E., 1964, J.D., 1967, University of Michigan.

Ronald W. Staudt is Associate Professor of Law, IIT/Chicago Kent College of Law. B.A., B.S., 1967, Saint Joseph's College (Indiana); J.D., 1970, University of Chicago.

This article is the second in a series describing an ongoing American Bar Foundation investigation into the feasibility of designing a processor to automate the delivery of legal services. The first article is James A. Sprowl, Automating the Legal Reasoning Process: A Computer That Uses Regulations and Statutes to Draft Legal Documents, 1979 A.B.F. Res. J. 1.

The investigation is funded primarily by the American Bar Endowment. Northwestern University has contributed data processing services to the project, and both the Northwestern University School of Law and the IIT/Chicago Kent College of Law have actively participated in the investigation. The clinic field tests described here were generously funded by the Council on Legal Education for Professional Responsibility. Other participating institutions are the Chicago Bar Foundation, the M. J. Murdock Charitable Trust, Harper College, Mallinckrodt College, the University of Illinois College of Law, the University of Michigan Law School, and the Northwestern School of Law of Lewis and Clark College. The authors are grateful to all of these institutions for their continued interest and support.

We also wish to thank the following individuals for their personal contributions to the clinic experiment: Dean Lewis Collens and law professors Thomas L. Eovaldi, Gary S. Laser, and Robert F. Seibel; attorneys Carolyn Hughes, Timothy Hughes, and Mark Neil; law students Victoria Cazel, Charles Dennis, Caryl Leightman, Frank Owen, Ronald Schwartz, and Carol Vogel; and the Legal Services Center staff, particularly Dorothy Gadzinski, Kathy Morgan, Phyllis Powell, and Vina Sisney. Teresa Stahulak's contribution to the project was invaluable as a paralegal, self-trained computer expert, and legal secretary. We also wish to thank the computer scientists and staff of Northwestern University's Vogelback Computing Center, especially Lawrence R. Atkin, Charles G. Filstead III, David A. Mausner, Benjamin A. Mittman, John L. Norstadt, and Melvin A. Schwartz. We are grateful for the assistance of our secretaries, Connie Schroeder and Regina Zerna Smith, who labored over the manuscript, and for the support of Jessie Buhmann and Bette Sikes and the other staff members of the American Bar Foundation and of IIT/Chicago Kent who assisted with preparing this article for publication.

INTRODUCTION

Many law schools have recently established clinical education programs in which law students work on actual cases under the supervision of experienced attorneys.[1] The primary function of such clinics is to educate the law students. But just as teaching hospitals serve as testing laboratories for experimental medical techniques, law school clinics can serve as realistic, yet sheltered, testing laboratories for experimental law practice techniques.[2] Within the sheltered clinic environment, educators and researchers can study the lawyering process in action and compare different methods of practicing and of teaching law.[3]

We recently conducted an experiment in such a clinic to test a new method of delivering legal services using an experimental computer. Among other things, we wished to learn whether attorneys and their assistants can be trained to set up and use a law office computer to gather client data and

[1] The growth and development of clinical legal education has been reported in annual surveys published by the Council on Legal Education for Professional Responsibility, Inc., New York City, from 1971 to 1979, under the following titles: Survey of Clinical and Other Extra-Classroom Experiences in Law Schools: 1970–1971; Survey of Clinical Legal Education: 1971–1972, 1972–1973, 1973–1974; and Survey and Directory of Clinical Legal Education: 1974–1975, 1975–1976, 1976–1977, 1977–1978, 1978–1979. A recent AALS-ABA report gives the following historical overview:

> Clinical legal education is now a part of the curriculum of nearly all law schools. . . . Although only fourteen subjects in the law school curriculum were the subject of clinics in 1970, the range of subjects continues to grow; by 1978–1979, fifty-nine subject areas were involved in clinical fieldwork. The number of programs in which student fieldwork is completely supervised by the law school has also been increasing and now includes over one-half of the country's clinical legal courses. [Footnotes omitted]

Association of American Law Schools–American Bar Association Committee on Guidelines for Clinical Legal Education, Clinical Legal Education: Report of the Association of American Law Schools–American Bar Association Committee on Guidelines for Clinical Legal Education, at 7 ([New York: Association of American Law Schools–American Bar Association Committee on Guidelines for Clinical Legal Education], 1980) [hereafter cited as Guidelines Report].

[2] The analogy to the teaching hospital is by no means of recent vintage. See, e.g., Jerome Frank, Why Not a Clinical Lawyer-School? 81 U. Pa. L. Rev. 907, 917 (1933). But see Packer and Ehrlich's comment, "The analogy is not particularly apt, however, for in law there is no analogue to the hospital, and hence none to the teaching hospital," *in* Herbert L. Packer & Thomas Ehrlich, New Directions in Legal Education 39 (New York: McGraw-Hill Book Co., 1972). For a strongly worded reply to Packer and Ehrlich see Steven H. Leleiko, Clinical Education, Empirical Study, and Legal Scholarship, 30 J. Legal Educ. 149, 159–65 (1979).

[3] Leleiko, *supra* note 2, at 158–65. There is extensive discussion of the variety of clinical methodology in the Guidelines Report, *supra* note 1. The report reviews clinical teaching methods, including observation, role playing and simulation, performance of lawyering tasks by students on live cases, individual discussion and counseling sessions, classroom instruction, and video tape review and analysis. *Id.* at 14–24, 47–82. The following Consultants' Reports, printed as Part III of Guidelines Report, also analyze various clinical teaching methods: Joseph D. Harbaugh, Simulation and Gaming: A Teaching/Learning Strategy for Clinical Legal Education, at 191; Robert Bastress & Joseph D. Harbaugh, Examining Lawyers' Skills, at 223; Elliott S. Milstein, The Design of the American University Criminal Justice Clinic, at 238; Harry I. Subin, Clinical Pedagogy—the Educational Program of the New York University School of Law Criminal Law Clinic, at 254.

Milstein states:

> Any discussion of the design of an on-going clinical program necessarily represents work-in-progress rather than a finished product. The program is different every year, perhaps every day. Each change in personnel, in the legal environment, in the dynamics of the student group, in the teaching materials, and even in the office space available for the clinic affects the execution of its goals or necessitates their re-definition.

Guidelines Report, *supra* note 1, at 253.

to "assemble" form legal documents for clients.[4] We also experimented with using the computer in other ways: to gather information about activities within the clinic, to generate administrative reports, and to develop a research data base that may help others to study the operation of the clinic. The experiment was funded by the Council on Legal Education for Professional Responsibility and was conducted at the Legal Services Center of IIT/Chicago Kent College of Law. Computer facilities were provided by Northwestern University and by the American Bar Foundation.

We set up the experiment to test an experimental computer system developed by the American Bar Foundation and called the *ABF Processor.* It is a prototype of computer systems that are almost certain to come into widespread use in law offices of all sizes within the next 10 to 20 years. Automated legal practice computer systems will help attorneys gather client information and assemble legal documents, and they will also help attorneys unravel the complexities of the law.[5] A research team headed by coauthor Sprowl had developed the prototype processor in an effort to harness the power of the computer to assist with the daily work of practicing attorneys. Before this experiment, the prototype processor had not been tested in a law practice environment.[6]

This article describes the experiment and sets forth some of the things that we learned during the two and a half years that it continued. We experimented with a variety of ways to train attorneys and their assistants to use the computer, and this article explores both our successes and our failures. Ultimately, the experiment verified that a practicing attorney and a legal secretary working together can design a computer system that can assemble legal documents such as wills and trusts. We also verified that such a system can be used effectively by other attorneys working in the same clinic. During the experiment, five automated legal practice systems were designed: a divorce system, a guardianship system, a will system, a trust system, and a real estate system. The will system has produced more than 450 wills for clients since it was first put into use.

An article to be published later describes the companion accounting and report-generating system that we developed on the same computer.

[4] The computer "assembles" a document by piecing together parts of boilerplate text and items of client data to form a document that is customized to the needs of a particular client. The boilerplate text is extracted from form documents stored within the computer's memory, and the client data are gathered during an "on-line" interview. The entire process is controlled by the form documents themselves, and no computer programs are required. See *infra,* sec. II.

[5] See generally James A. Sprowl, Automating the Legal Reasoning Process: A Computer That Uses Regulations and Statutes to Draft Legal Documents, 1979 A.B.F. Res. J. 1. The ABF processor is also intended to be used for computer-assisted instruction, but this experiment tested its effectiveness only as an automated tool for drafting legal documents in a law school teaching clinic environment.

[6] See summary description of prior tests and training efforts in Sprowl, *supra* note 5, at 77–79. See also sec. IIIC, *infra.*

The reports produced by that system summarize the activities of the attorneys and students and also provide information about the clinic's cases and clients. The system also maintains a data base that is available to researchers who may wish to investigate the teaching and lawyering processes in a law school clinic environment.[7]

In sections I and II of this article, we briefly describe the computer technology used in the experiment. Section I presents a nontechnical introduction to digital computers, and section II introduces the ABF processor and explains how it can be set up by an attorney to gather client information and assemble documents. Section III identifies the research objectives of the experiment and the environment in which it was conducted, and section IV presents a narrative description of the experiment. Section V sets forth our conclusions and notes some recent developments that may soon lead to the practical deployment of automated practice systems of the type described here.

I. A Concise Introduction to Computers[8]

Digital computers are machines that accept numbers, move them about, perform computations upon them, and ultimately return them in a changed fòrm. Digital computers effectively differ from one another only in speed and in the quantity of numbers they can handle. The numbers that flow through a law office computer typically represent letters, digits, and punctuation symbols. For example, when the upper-case *A* key is struck on a keyboard attached to such a computer, electronic logic associated with the keyboard generates the number *65* to represent the letter *A* and sends this number to the computer.[9] When the same computer

[7] Boyer and Cramton noted in their 1974 survey of research on legal education that there is a need to learn more about how law professors and law students spend their time. Barry B. Boyer & Roger C. Cramton, American Legal Education: An Agenda for Research and Reform, 59 Cornell L. Rev. 221 295–96 (1974). Assuming that methods for controlling the accuracy of the data accumulation process can be implemented, computer management systems such as the one we have developed may prove to be useful vehicles for gathering nationwide data on the activities of clinical educators and students. Our computer management system is described in a companion article to be published later.

[8] A more extended introduction to computers as they relate to law can be found in James A. Sprowl, Evaluating the Credibility of Computer-generated Evidence, 52 Chi.-Kent L. Rev. 547 (1976).

[9] In the ASCII (American Standard Code for Information Interchange) code, every printable character and a number of commands are represented by numbers between zero and 127. The number *65* is assigned to *A*, the number *66* to *B*, and so on. See American National Standards Institute report ANS-X3. The ASCII code is adhered to by most terminal manufacturers and by many computer manufacturers, including Digital Equipment Corporation. IBM generally adheres to the EBCDIC (Extended Binary Coded Decimal Information Code) code in which printable characters and commands are represented by numbers between zero and 255 and in which *193* is assigned to *A*, *194* to *B*, and so on. Many terminals support the EBCDIC code as well as the ASCII code. Control Data Corporation has its own *extended display code* which includes the numbers zero to 63 and which represents printable characters and commands by sets of one or two of these numbers. In this extended display code, *1* is *A*, *2* is *B*, and so on. These three are the dominant code conventions at the present time. The incompatibility of these three code conventions is one of many reasons why it is difficult to move programs from the computers of one manufacturer to those of another.

sends the number *65* to its television screen, electronic logic associated with the screen writes the upper-case letter *A* on the face of the screen. Since nothing but numbers exists within the computer, the computer cannot "understand" anything. Computers are simply machines that respond to incoming numbers in a mechanical way.

The personality of a computer comes not from the computer itself but from the *programs* or sets of numeric instructions that control its actions. These numeric instructions tell the computer what to do with the numbers. A typical computer program may contain hundreds of individual numeric instructions, and it may take several hundred programs to enable a computer to perform the simplest useful task. A *dedicated* computer is one whose programs enable it to perform only one specialized task, such as word processing (text entry, revision, and printing). A *general purpose* computer is one that is controlled by one set of programs at one time and by a different set of programs at another time. A general purpose computer can be used for word processing and for law office accounting as well as for other tasks. The popular video games are general purpose computers supplied with cassette *modules* that are inserted to produce different games at different times. Each such module contains a set of related programs that together define a game. While law office general purpose computers do not typically include such cassette modules, they do contain sets of related programs that instruct the computer how to perform different tasks. By analogy to the game computers, we shall call such sets of related programs "modules." A typical law office computer in the 1990s probably will contain many modules that will enable it to perform tasks such as word processing, attorney time accounting,[10] cash disbursements accounting,[11] billing,[12] accounts receivable, litigation support,[13] automatic docketing and motion filing,[14] and conferencing.[15]

Instead of directly instructing a computer in the performance of a particular task, some modules enable the computer to translate English-like

[10] The computer will prepare reports summarizing how much time each attorney spent working on each case each month.

[11] The computer will report on the cash disbursements made for each case each month.

[12] Client bills will be written including varying amounts of detail from the attorney time and cash disbursement information.

[13] Searching through abstracts of litigation exhibits and through deposition transcripts, the computer will facilitate the management of large amounts of evidentiary information.

[14] The lawyer's computer will communicate with the court computer by telephone to make docket book entries and also to file motions and briefs.

[15] A *computer conference* is a discussion conducted by parties who leave each other messages within a computer. Much of our present-day motion practice could be carried out using a computer-conferencing arrangement established between the judge and the attorneys for both sides. One attorney would file a motion by submitting it to the computer, and the motion would be delivered to the opposing attorney the next day. The opposing attorney would then file a "response" with the computer. The next day, both the motion and the response would be placed before the judge who would type an "order" into the computer for delivery to the parties the next day. By avoiding court appearances and by never requiring the judge and the attorneys to physically meet such motion conference arrangements could significantly reduce the cost of pretrial practice.

instructions into the numeric instructions that control the computer's actions.[16] When controlled by such a module, a computer becomes a *processor* for the English-like language.[17] For example, a FORTRAN processor can accept instructions written out in the English-like language called FORTRAN, and a COBOL processor can accept COBOL language instructions.[18]

An ABF processor is a general purpose computer programmed to accept instructions written in the special *ABF language* described in section II. The program module that converts a computer into an ABF processor is a language translation module that must be used in conjunction with a suitable word-processing module, a document storage module, and a document-printing module. The ABF language simplifies the task of designing automated legal practice systems such as estate-planning systems, probate systems, and complaint-drafting systems. A primary object of this experiment was to test our hypothesis that attorneys and their assistants could use the ABF language to design such systems with minimal help from computer professionals. As this article relates, we ultimately verified our hypothesis with respect to legal form document assembly systems but only after extensively reprogramming the ABF processor and radically altering the way we instructed attorneys and legal assistants in its use.

II. A Concise Description of the ABF Processor and Language

The ABF processor is fully described in a recently published article, and those wishing a complete description of the processor will want to read that article.[19] The concise description of the ABF processor and language set forth below is only an introduction to this new technology. It is included here as background for those who have not read the previous article.

[16] Such a module is called a *compiler* if it actually translates an English-like language into numeric instructions. But often such a module merely "interprets" the English-like language and then tells the computer what to do, in which case it is called an *interpreter* or *interpretive processor* for the language.

[17] The term *processor* is often used interchangeably with the term *computer.* We use *processor* here in a limited sense. In this article, a computer is a machine that becomes a processor for a specific English-like language when the computer contains a module designed to translate the English-like language into the numeric commands that can control the computer.

[18] The most commonly used English-like languages have been COBOL (for business and accounting applications), FORTRAN (for scientific and engineering applications), PL/1 (supported primarily by IBM), BASIC (for time-sharing computers and microcomputers), PASCAL (widely used by computer scientists on microcomputers), ADA (a grandchild of PASCAL, sponsored by the United States Department of Defense, which may soon emerge as the most widely used language), APL (a very powerful, concise language used by many computer scientists), SNOBOL (a language for manipulating strings of text), and LISP (a language used in designing "artificial intelligence" systems that simulate human thinking). While standards exist for most of these languages, the versions supported by different computer manufacturers are not always compatible—another bar to moving programs from one computer to another. For more information on these and other computer languages see sources cited in Sprowl, *supra* note 5, at 14 n.23, and Peter Grogono, Programming in PASCAL (rev. ed. Reading, Mass.: Addison-Wesley Publishing Co., 1980); Laurent Siklóssy, Let's Talk LISP (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1976); and Peter Wegner, Programming with Ada: An Introduction by Means of Graduated Examples (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1980).

[19] Sprowl, *supra* note 5.

Digital computers are normally programmed *algorithmically.* An *algorithm* is a set of specific, step-by-step rules for the solution to a specific problem.[20] For example, the quadratic formula is an algorithm that defines a way to solve quadratic equations. The instructions that come with a nonassembled toy constitute an algorithm that explains how the toy is to be assembled. The detailed check list that guides many attorneys through the complicated steps of opening an estate is an algorithm that defines a solution to the problem of opening an estate. All of the commonly used computer-programming languages (including FORTRAN, COBOL, and numeric machine instructions) are algorithmic languages that enable the skilled computer professional to specify the exact sequence of actions the computer must take to solve a specific set of problems. Many computer professionals are so accustomed to programming computers algorithmically that they cannot conceive of computers being programmed in any other way. But a predesigned algorithm is only one of a possibly infinite number of different ways in which a particular problem may be solved.

The ABF language is not a conventional algorithmic programming language—it is a language that one may use to draft form documents, and also to draft the *statute-like procedures*[21] that specify how the form documents are to be filled in and assembled. When form documents and procedures drafted in the ABF language are fed into an ABF processor, they enable the processor to write its own algorithmic programs that can then guide the computer through the steps of asking questions, accepting the answers supplied, drawing legal and other conclusions, and assembling client-customized versions of the form documents. The ABF processor, guided by form documents and statute-like procedures previously drafted by law office personnel, programs itself to solve the problem presented by a specific client need. An automated delivery system is composed of form documents and statute-like procedures that control the ABF processor's assembly of client documents in one area of law practice. The ABF processor thus eliminates the need to have a computer specialist write lengthy programs embodying detailed, algorithmic solutions to the problems of automating the delivery of legal services.

[20] This definition of the term algorithm is essentially the meaning that is taught to students of computer science. A much more precise definition of the term can be found in Bryan Niblett, Legal Protection for Computer Programs 13 (London: Oyez Publishing Ltd., 1980), where Niblett, a barrister as well as a professor of computer science, defines an algorithm as

> a set of systematic rules or statements for solving a problem encountered in the design of a program. In his work on the art of computer programming, Knuth . . . says: 'The modern meaning for algorithm is quite similar to that of *recipe, process, method, technique, procedure, routine* except that the word "algorithm" constitutes something just a little different.' Knuth goes on to distinguish five important features of an algorithm: that it contain a finite number of steps; that each step be precisely defined; that it have zero or more input quantities; that it yield one or more output quantities; and that it be effective in the sense of consisting of basic steps which can be carried out exactly in a finite time.

[21] Discussed in Sprowl, *supra* note 5, at 36–37.

Form documents drafted in the ABF language closely resemble the form documents that are to be found stored within any computerized word processing system, and the form documents are typed into the computer in the conventional way.[22] But ABF form documents contain bracketed commands drafted in the ABF language that enable the computer to automate their assembly. At every point in the form where the computer must later insert client information, the attorney-drafter inserts a bracketed instruction that describes that client information. For example, the beginning of a form will document might look like this:[23]

> LAST WILL AND TESTAMENT
>
> OF
>
> [the name of the testator]
>
> I, [the name of the testator], of [the city and state where the testator lives], do make this my last will and testament.
>
> I leave all of my personal effects and household goods to my [the testator's spouse, a husband or a wife], [the spouse's name], if [he or she, the spouse's subjective pronoun] survives me, and otherwise to my children who survive me in equal shares.

Optional passages are preceded and followed by bracketed commands to the computer indicating the condition or conditions that are to control their inclusion or exclusion:

> [IF the death taxes ARE to be paid out of the *res* of the estate INSERT]
>
> I instruct my executor to pay all death taxes out of the *res* of my estate so that the legatees and devisees do not have to pay any such taxes out of their individual gifts.
>
> [END IF]

(END IF is an abbreviation for "end of the IF passage.") Passages that are to be repeated with different data insertions are similarly preceded and followed by bracketed commands so indicating.[24] Through careful use of these techniques—data insertion, clause selection, and repetitive

[22] The document is typed into the computer and is stored within the computer as a single unit, no matter how long it may be. Northwestern's computer stores the document on a spinning magnetic disk. At midnight, the Northwestern computer automatically produces two magnetic tape backup copies of each newly entered or altered document, and these backup copies are stored away from the computer. Backup copies of computer-stored material are essential. Many law firms are very negligent in not maintaining backup copies of the material they have stored on magnetic cards, floppy disks, or hard disks even though it would cost thousands of dollars to rekey the material if it were accidentally destroyed or erased.

[23] This is not an actual will suited for client use but only an example of how the ABF language is used. The will system designed by Seibel for use by the clinic staff is not reproduced in this article. Seibel plans to publish a separate article describing the will and trust systems.

[24] Passages that are to be duplicated are bracketed by [REPEAT] . . . . [END REPEAT]. Such passages must contain a bracketed statement that indicates the condition that is to stop the duplication of the passage. See Sprowl, *supra,* note 5 at 25–30.

clause insertion—an attorney-drafter may describe the structure and contents of even the most complex legal form document.[25]

Once a form is written in this way, a computer equipped with the ABF processor can automate its assembly. When an attorney or secretary needs to have a document prepared for a specific client, he or she calls on the computer to "process" the form document (with its embedded commands written in the ABF language) into a client-customized version of the form. The computer derives questions from the form, asks the questions, accepts the answers supplied, and assembles a client-customized version of the form that conforms to the answers supplied. For example, when the ABF processor processes the foregoing will excerpts the computer asks following questions:

Q: What is the name of the testator?
A: John Q. Smith
Q: What is the city and state where the testator lives?
A: Evanston, Illinois
Q: What is the testator's spouse, a husband or a wife?
A: wife
Q: What is the spouse's name?
A: Susan W. Smith
Q: What is he or she, the spouse's subjective pronoun?
A: she
. . . .
Q: Are the death taxes to be paid out of the *res* of the estate?
A: yes

After the interview is complete, the computer assembles a will for John Q. Smith conforming to the ABF instructions and to the data supplied during the interview. But the attorney is not locked in to the specifics of the form will, for the computer then converts itself into a word-processing system[26] and presents the will to the attorney, who may modify it in any way. Ultimately, the will is printed out at high speed on an automatic typewriter, ready to be signed.

The questions asked by the ABF processor during the "client interview"[27] are derived automatically from the instructions drafted (and em-

[25] For a complete explanation of the way documents are described in the ABF language, see *id.* at 17–30.

[26] Actually, the computer does not "convert" itself into anything. The ABF processor module simply turns control of the computer over to the word-processing module. The word-processing module then takes whatever action is necessary to bring the client-customized will into its "workspace" where it can be reviewed and revised. A separate printing module later controls the printing of the will. We found it essential to arrange the ABF processor to pass the completed will to the word processor automatically, because beginning trainees became confused if they were required to fetch the will into the word processing environment by issuing a FETCH command following the client interview.

[27] We use the term *client interview* here and elsewhere to refer to an interview such as the one just described. The exchanges are actually between the attorney (or secretary) and the computer *about* the client and the service needed. The client may be present, or the questions may be answered

bedded in the form document) by the attorney-drafter. If these automatically derived questions are not satisfactory, the attorney-drafter may improve the legal practice system by replacing these questions with new and better questions or directives. To create a new question or directive, the attorney-drafter writes another form document (optionally containing bracketed instructions such as those illustrated above) that the computer displays in place of the automatically derived question. This feature permits the attorney-drafter to modify the questions asked to match the sophistication of the person who conducts the client interview. If the interviewer is not skilled in estate planning, the questions can be written to include elaborate explanations of the possible choices together with statements of the legal consequences of each choice. If the interviewer is a skilled estate planner, the questions can be kept short and can be drafted in technical language. Alternatively, each question may include several different formulations preceded by bracketed IF . . . END IF commands that enable the computer to select the formulation appropriate to the needs and desires of the interviewer.[28]

In addition to form documents, an automated delivery system normally includes statute-like procedures that define how the documents are to be assembled. When designing a legal practice system, the attorney-drafter may use procedures to reduce the length of the client interview and to use the power of computer calculation in the form assembly. If the computer asks for information that can be "computed" from other information, the attorney-drafter can supply the computer with procedures that instruct it to perform the necessary computations. For example, in the interview set forth above the computer asks:

> What is the testator's spouse, a husband or a wife?

and

> What is he or she, the spouse's subjective pronoun?

Both questions would be unnecessary if the computer could compute the answers to these questions from the sex of the testator. To have these answers computed automatically, the attorney drafts and submits to the computer the following two procedures:

```
IF the testator IS a man
```

from information obtained earlier from the client and placed in his or her file. The interview process could be set up for client self-interviews if adequate safeguards were included to be sure that the questions were understood and answered completely and accurately. The automated practice systems described in this article were not designed to be used by unsophisticated clients who were not assisted by a lawyer or a paralegal.

[28] See Sprowl, *supra* note 5, at 54–56 for a more detailed explanation of how replacement questions and directives are actually created.

```
THEN the testator's spouse, a husband or a wife="wife";
OTHERWISE the testator's spouse, a husband or a wife="husband";
```

and

```
IF the testator IS a man
THEN he or she, the spouse's subjective pronoun="she";
OTHERWISE he or she, the spouse's subjective pronoun="he";
```

The computer then asks the single question

Is the testator a man?

and uses the answer supplied to compute whether *husband* or *he* or, alternatively, *wife* and *she* will be inserted in the client-customized version of the will. Other procedures can determine which passages are to be included or excluded.

More complicated procedures can bring the computational power of the digital computer to bear upon the problem of assembling a form legal document that precisely fits the needs and desires of the client.[29] In a sales contract, the contract price can be computed from the number of items sold, the price per item, the quantity discount, and the transportation charges. In a tax return, the adjusted gross income can be computed from the net income. In a real estate closing agreement, the prorated tax amount can be computed from the closing date and the annual tax figure. In a complaint or decree, the precise language can be "computed" from words and phrases picked by the interviewer from a "menu" of numbered alternatives. The possibilities are virtually endless.

Some procedures resemble statutes, and they enable the ABF processor to draw simple legal conclusions. For example, the following procedure might be used in a probate system to control the selection of the documents that are to be prepared:

```
IF
the decedent's children and the named beneficiaries ARE of sound mind
and over 20 years old
AND
the named executor IS a resident of the state of Illinois
AND
the decedent HAS waived security on the executor's bond in the will
AND
the decedent's children and the named beneficiaries HAVE consented to a
settlement out of court
THEN
```

[29] See *id.* at 35–50 and 56–60 for a complete explanation of procedures.

the estate may be probated without court supervision
OTHERWISE
court 'supervision is mandatory

This similarity of some procedures to statutes permits one to conceptualize the ABF processor as a computer that uses regulations and statutes to control the drafting of legal documents. The attorney drafts documents and procedures in the ABF language and thereby forms an image of the law in a particular area, including the applicable regulations, statutes, and cases. Whenever the law changes, the attorney-drafter corrects the computer-stored procedures and documents to reflect the changes. Since statute-like procedures and form documents written in the ABF language are conceptually more familiar to attorneys than are conventional computer programs, the designers of the ABF processor hypothesized that the ABF processor would enable attorneys and their assistants to work directly with the computer without a professional computer programmer. The ABF processor would eliminate the problems that can arise when an attorney who knows little about computers and a computer professional who knows little about the law attempt, as a team, to design a system.[30]

III. Field Test Preliminaries

A. The Research Objectives of the Field Test

The principal objective of the team that designed the ABF processor was to create a friendly environment within the computer that would permit law professors and practicing attorneys and their assistants to set up, maintain, and operate automated law practice systems with only occasional assistance from skilled computer professionals. It was thought that some attorneys would develop and maintain systems on office computers for their own personal use and for the benefit of their partners and associates. Other attorneys, particularly law professors and experts with highly specialized practices, would develop and maintain more sophisticated

[30] For a complete explanation of the rationale underlying the design of the ABF processor, see *id.* at 4–17.

Columbia University law professor Robert Hellawell, who learned computer programming to design a tax-planning computer system, analyzed the difficulties of using a computer specialist to design such a system:

> A possible obstacle to using computers for legal analysis will be that writing programs on complex legal subjects is a difficult and time-consuming job. Moreover it is doubtful that a nonlawyer professional programmer could take on as much of the work in legal programming as he can in many other areas. My work on CORPTAX, for which I did both the legal analysis and the computer coding, convinced me that an expert on the legal problem being programmed will need to be closely involved throughout. Most of the coding is the coding of legal decision points. The plan or algorithm of the program is almost wholly one of legal decisions. A legal expert, or a junior lawyer under expert supervision, will at least have to write the plan, and that task will be very time consuming. Indeed, it will be the bulk of the work. In contrast, much standard nonlegal programming may require a comparatively smaller component of professional decision-making.

Robert Hellawell, A Computer Program for Legal Planning and Analysis: Taxation of Stock Redemptions, 80 Colum. L. Rev. 1363, 1393 (1980).

For a comparative discussion of the ABF processor approach to system design and the approach taken by most computer professionals, see Sprowl, *supra* note 5, at 6–9, 12, and 15.

systems that would be shared by attorneys who lacked extensive experience in such areas but needed to provide services to their clients. Preliminary tests of the earliest version of the ABF processor indicated that these objectives might be feasible, but these preliminary trials did not test the technology in a law practice environment. A major goal of the ABF design team was to achieve a technology that could be mastered and used effectively by attorneys with heavy responsibilities to clients. This experiment, therefore, focused its attention on two key questions: Could busy practicing attorneys and their assistants be taught to set up and use the ABF processor? Could these attorneys and their assistants create automated legal practice systems that would be beneficial, convenient, and cost effective?

B. The Site of the Experiment

The clinical education center of IIT/Chicago Kent College of Law is called the Legal Services Center.[31] We selected the center as our test site because its staff and its director, Gary S. Laser, were enthusiastic about participating in the experiment, and because the center was experimenting with prepaid legal service contracts when other law school clinics in the Chicago area were generally providing services only to indigent clients. A significant number of the center's clients are property-owning middle-income persons whose legal needs resemble the needs of clients generally.[32] We therefore felt the center provided a realistic site for testing computer systems designed to be used by general practitioners. Also it offered the prospect of a large volume of work in certain legal areas. Any automated legal practice system must be used often so that the fixed cost of setting up the system can be spread over many individual uses. The center's contractual service arrangements appeared to provide enough repetitive work to permit us to estimate the cost effectiveness of this new technology.[33] The center also represented a sheltered environment in which this·new tech-

[31] The Legal Services Center, located on the first floor of the IIT/Chicago Kent College of Law building at Wacker Drive and Monroe Street in downtown Chicago, includes 16 private offices and 5 large rooms. Three of the private offices are small rooms designed to hold a computer terminal and, at most, 2 or 3 chairs. These offices are used for computer programming and for delivery of legal services using the computer. The 5 larger rooms are a library, 3 student workrooms, and a work area for 3 secretaries plus a computer terminal.

[32] "Demand for, and use of, lawyers' services by people of moderate means are both highly elastic. They are determined by certain elements of competition, including: the quality of the service; its price; its accessibility; and public knowledge and attitudes about law, lawyers, and lawyers' services." Barlow F. Christensen, Lawyers for People of Moderate Means: Some Problems of Availability of Legal Services 38 (Chicago: American Bar Foundation, 1970). The center's service of moderate income clients for fees was a new and growing part of its program. These client groups had produced some cases that would not usually occur in a poverty law office: for example, real estate closings, income tax return preparation, estate planning, evictions representing the landlord, probates, and divorces involving property settlement arrangements.

[33] For example, in 1979 the center opened approximately 20 new cases each month for clients seeking simple estate planning or a new will. During the experiment, the center normally received from 150 to 300 "new client matters" each month. Approximately one-half of these new client matters were resolved during the initial consultation; the remaining matters required additional legal services of one type or another.

nology could be tested without fear that a failure would cause catastrophic consequences.[34] The sheltered, educational character of the test site diminished the rigor of the experiment, for the experiment was not truly conducted "in the field." The test site was not a profit center supporting its staff with income derived solely from clients. The attorneys did not have to make hard economic decisions on when to use the computer and when to do things manually—such decisions often reflected the interest and enthusiasm of individuals rather than their business judgment. In addition, the ABF processor itself should not be regarded as an invariant element of this experiment because improvements and changes were constantly being made to the ABF processor and to other computer modules.

1. The Clinic Staff

The Legal Services Center faculty during the test period included Director Laser, Associate Director and coauthor Staudt, and four other attorneys[35] who supervised the work of students on actual cases and gave weekly lectures. The staff included several other attorneys who handled repetitive cases that the center was contractually obligated to accept.

2. The Clients

The great majority of the center's clients were accepted through a contract with the Mayor of Chicago's Office for Senior Citizens and the Handicapped. More than 100 clients were referred to the center each month from this source alone. While most of them were not charged a fee, a voluntary fee schedule was established for those willing and able to pay, and a significant number did pay. The clinic accepted other clients under a contractual prepaid plan established with a local garment manufacturer. Under this plan some legal services were provided without charge, and others, for set fees. Under an arrangement with the public guardian and the probate court, the center also accepted as clients indigent minors who needed to have appointed a local guardian other than a parent.[36]

[34] The "sheltered environment" was created by the generous financial support this experiment had from the Council on Legal Education for Professional Responsibility (CLEPR) and by support from the law school. CLEPR authorized $100,000 to support the ABF processor experiment and the other experiments in automating the center's activities. The center considered the CLEPR grant to be seed money funding the establishment of a self-supporting or partially self-supporting clinical education program, built on a clinic serving moderate-income clients and charging modest fees. It was hoped that the computer technology would form a cornerstone of this moderate-income clinical practice and would enable the clinic to provide services of higher quality at reduced costs. If the computer technology did not assist the clinic in providing efficient, cost-effective legal services, then its use might have to be discontinued after the end of the experiment. But to date, the computer is still in operation.

[35] During the experiment, there was a significant turnover of both professional and nonprofessional personnel at the center. So far as we can determine, the computer did not cause lawyers and legal secretaries to take jobs elsewhere. But this turnover did have an unexpected beneficial impact on the experiment: because new faces often appeared, we were provided with many opportunities to test different training strategies and different designs of the processor itself.

3. *Computer Resources*

At the outset of the experiment, the prototype ABF processor was already installed on a general purpose Control Data Corporation computer located on Northwestern University's Evanston campus. Access to the ABF processor could be gained by telephone from almost any standard computer terminal.[37] The Northwestern computer also included a word-processing module and a separate document-printing module that had to be used in conjunction with the Northwestern version of the ABF processor. The Northwestern word-processing module had been designed for use by skilled computer programmers and had been modified to perform text entry and revision tasks. While we initially had reservations about the suitability of this module, it proved to be adequate for the purposes of our experiment.[38] The word-processing module included a document storage module, but we were forced to design our own document storage module

[36] The center also accepted a variety of clients from other sources to provide a variety of lawyering experiences for the students. For example, Staudt was teaching a course on employment discrimination in the clinic, so he and his students frequently represented clients who claimed they were victims of job discrimination. Other cases requiring significant amounts of litigation were accepted to provide trial opportunities for student interested in litigation. In January 1979, the center began a criminal litigation program and added two criminal litigation experts to its faculty. The cases that originated from the sources just listed were not sufficiently repetitive to be suitable candidates for automated document processing within the confines of our small clinic. But we see the need to provide automated drafting systems for pleadings and motions for such cases in the future when it will be possible to market automated form systems to attorneys throughout a large county or a state or even throughout the country. For example, though the center did not have a sufficient volume of evidence-suppression motions in criminal cases to warrant automating the drafting of such motions, there is clearly a sufficient volume of such motions filed in Cook County to warrant the development of a county-wide motion-drafting system that could be shared by many defense attorneys.

The litigation faculty experimented briefly with using sorting and searching features of our computer to help them sort through abstracts taken from depositions originating in a criminal action involving multiple defendants. Our flexible command module, described in sec. IVB, *infra,* enabled us to build up a set of special commands to facilitate this work.

[37] See notes 41, 43, 44, and 45 *infra* for a description of the terminals we selected for this experiment.

[38] The modern trend in text entry and revision systems is toward *screen-oriented* systems that permit one to display a document on a viewing screen, move a *cursor* (or bright spot) to the point where a correction is to be made, and then type in new material or erase unwanted material (erasure is accomplished by passing the cursor over the unwanted material). Screen-oriented systems work best when only a small number of revisions are required and when they are run by full-time operators; they are therefore generally relegated to word-processing centers. Those most skilled at operating screen-oriented systems are frequently not as highly skilled in typing, spelling, grammar, and style as are many legal secretaries, who often prefer to use standard electric typewriters and who, by making fewer errors, can often outperform less skilled operators of the screen-oriented systems. The complexity of the screen-oriented systems is such that attorneys almost never attempt to operate them.

The Northwestern word-processing module was not screen oriented. It displayed the text of a document with a number assigned to each line. To delete material, one typed the letter *D* followed by the line number or range of line numbers to be deleted. To insert material, one typed *T* followed by the line number of the insertion point, and then typed in the insertion. In addition, the module could search single lines, sets of adjacent lines, or an entire document for words or phrases and could replace these words or phrases with others. To use the Northwestern module, one needed to master only four commands—insert, delete, search and replace, and display. The simplicity of this module made it possible for everyone in the clinic to understand its use. Attorneys could produce and revise wills in the evening without secretarial assistance.

to hold down the cost of using the computer.[39] The printing module was a product of student labor at the University of Calgary, and it contained a number of peculiarities that occasionally caused our trainees great distress.[40] Many of its shortcomings were overcome when we later attached it to the new command module that is described further on. The ABF processor, the word-processing module, the document storage module, and the printing module were not designed to be used together. Since we had to use all of these modules together, our trainees had to master each of them separately and then learn how to coordinate their independent operations. As this article relates, we were ultimately forced to integrate these separate modules into a unified system.

Several options were available to us other than using the Northwestern University word-processing and printing modules. We considered purchasing a dedicated word-processing microcomputer to serve as *front end* for the ABF processor, connecting the microcomputer to Northwestern's computer by telephone. Since such a microcomputer could have been used by the center staff for many purposes other than just for preparing ABF form legal documents, the clinic's legal assistants and secretaries then would have been testing the ABF processor in conjunction with a familiar word processor. But such microcomputers were quite expensive when this experiment was launched, and the additional complexity of such a distributed system might have created additional training problems. Accordingly, we chose to use the Northwestern modules despite our reservations.

Inexpensive terminals installed in the clinic permitted access to the computer by telephone.[41] Telephone line interference occasionally interrupted our communications, and the center staff had to be trained to dial again whenever such interference became a problem. All but one of our terminals operated at character transmission speeds of about 30 characters per second[42]—fast enough for computer-conducted interviews but too slow for general text entry and revision. We upgraded the center's ex-

[39] See note 53 *infra.*

[40] On one occasion, it insisted upon inserting a randomly selected character into the first page of a will. We never were able to trace down the source of this particular problem, but when we made a trivial change in the will, the problem disappeared and never returned.

[41] In addition to automated law practice systems, we also developed management and accounting systems during the course of the experiment. When we purchased the terminals to begin our work on the ABF systems, we still did not know what kind of technology we would use to perform the accounting and management functions. Therefore, we purchased general purpose terminals that could be used in a variety of ways. As we explain in an article to be published, our accounting and management systems were eventually built within the Control Data computer that also contained the ABF processor system. Access to both systems was obtained through the same terminals, using the same front-end command module (described in sec. IVB). The management and accounting systems utilized several special features of our terminals and printers such as the special-function pushbuttons, the automatic page advance, and the automatic fill-in-the-blank questionnaire devices. None of these were used by the ABF processor module.

[42] Exactly and technically expressed, the transmission speed was 300 baud.

isting magnetic card–controlled typewriter to print the legal documents in final form on bond paper and to capture them simultaneously on magnetic cards,[43] and we purchased a similar communicating typewriter having no magnetic card facilities.[44] We also purchased two terminals, each equipped with television screen and keyboard. One of these terminals transmitted and received data at a much faster 120-characters-per-second rate to facilitate text revision.[45]

Certain problems will arise in any project that relies upon a shared university computer or any other shared computer that is subject to intermittent heavy usage. At certain times during the day and at certain times during the school term, the heavy use by Northwestern University undergraduate students severely hampered the computer's ability to respond promptly to commands. When the ABF processor was conducting a client interview at such times, the delay between our response to one question and the display of the next would become excessive.[46] The Northwestern computer also malfunctioned occasionally, and there was no backup computer available.[47] By scheduling most of our computer activities in

[43] This unit was an A. B. Dick Magna I automatic "daisywheel" typewriter that is compatible with IBM Mag Card II equipment. We equipped the Magna I with an optional communications package that enabled it to communicate with the Northwestern computer. We then leased a special telephone that included a *modem* or computer-telephone interface. To avoid switchboard interference problems, we installed a dedicated telephone line for this terminal.

[44] This unit was an Anderson Jacobson automatic "daisywheel" terminal. We connected it to a ComData 300-baud modem that contained a cradle for a standard telephone handset. This terminal was equipped with a *forms tractor* that enabled it to print out lengthy administrative reports on accordion-fold paper with perforated tear-away edges. At an earlier point in the experiment, we had leased a Teletype 41 printing terminal from the telephone company. The Teletype 41 coupled directly into any phone line without the need for a separate modem. It was quite compact and had readable type, but its dot-formed characters were not suitable for court filing.

[45] These were microcomputer-controlled Teleray terminals, model 1061. One was connected to a ComData modem having a cradle for a standard telephone handset. This terminal operated at the slow 30-characters-per-second rate (300 baud). The other was connected to a special data telephone by a Vadic modem which operated at a 120-characters-per-second rate (1,200 baud). The faster data rate meant that this terminal was considerably more vulnerable to telephone static. The static problem was particularly severe when we had to dial from Chicago to Evanston. After Northwestern installed a Chicago entry port to its computer, we were able to contact the computer by a local call placed within Chicago, and then the static problem was much less severe. But occasional garbled lines were always a problem with this terminal, possibly because of some slight mismatch between our equipment and Northwestern's.

[46] Assembling a will for a client, a task that normally took 5 or 10 minutes, could take as long as 30 or 40 minutes in the late afternoon or at the end of a school term because of the much heavier student use of the computer at those times. This was largely due to the inefficient design of the prototype ABF processor, which needed 60,000 computer memory locations to function—a very large amount of memory compared to the less than 10,000 locations that Northwestern's word processor module required. At times when many small jobs were being run, the automated job scheduler on Northwestern's computer penalized the ABF processor because it required such a large block of memory.

[47] The Northwestern computer functioned flawlessly most of the time. But some weeks it would "crash" (cease operating entirely) several times during each day until some major technical problem was diagnosed and corrected. If such a crash occurred when one had just about finished a will interview, and if the "job" was not "saved" and resumed, then one had to start all over again.

the mornings and by simply not using the computer on days when the computer was malfunctioning or overloaded, we managed to minimize the adverse effects of these problems. We mention these difficulties only to illustrate the limitations under which this experiment was conducted.[48]

Design work on the ABF processor began in May 1976. One year later, when the processor was operational but unfinished, Sprowl arranged a modest preliminary trial of this prototype. A group of eight trainees, including paralegals, teachers from Harper Junior College, and paralegal trainees from Mallinckrodt College, attended a series of evening training classes. Each class session included an hour of lecture and an hour of supervised laboratory work. A 30-page primer on the ABF language had been prepared to assist these volunteer trainees in using the early version of the processor.

In spite of the enthusiasm of the trainees, some of whom drove 50 miles round trip to attend, the classes were not very successful. During the lectures, Sprowl placed too much emphasis upon teaching the trainees how to draft documents using the ABF command language, before the trainees had sufficiently mastered the word-processing system. Accordingly, the trainees' initial enthusiasm to create legal practice systems was quickly dampened by the difficulties they encountered when they first attempted to enter and revise the text of their documents. The ABF document-drafting language was also overly rigid at that time. For example, it required a THEN after every IF, and it assigned different meanings to EQUALS and =. The trainees frequently omitted THEN, freely interchanged EQUALS and =, and made other similar mistakes. Virtually no error-detecting logic was built into the ABF processor—it would attempt to process form documents containing serious errors, often with comic results. For example, the quotation mark and the apostrophe share the same key on many typewriters. If a trainee typed a quotation mark instead of an apostrophe, the ABF processor refused to process half of the document, giving no indication of what had gone wrong, much to the trainee's annoyance.

Despite these difficulties, the trainees did create a number of operative, albeit somewhat trivial, legal practice systems. Those who had previously used computer-based word-processing systems appeared to make better progress than did the others, and at the time we concluded that they were better prepared to master the ABF language. Accordingly, we selected trainees at the center who had prior experience in word processing. We

[48] The experiment was designed to last only a short time, and our resources were not sufficient to warrant the purchase of in-house computing equipment for the experiment. Nor do we wish to criticize the staff of the Vogelback Computing Center at Northwestern University. At the outset of this experiment we were aware of the limitations just described, and the fact that the experiment was quite successful in spite of these limitations is, in large measure, due to the support we received from the Vogelback staff.

were wrong, however, and now believe that general intelligence is a much better indicator of how easily one will master the ABF processor than is word-processing experience. College graduates make the best ABF trainees. It is encouraging to note that even the most primitive version of the ABF processor was apparently less difficult to master than was its companion word processor, an indication that the ABF technology is not too difficult for attorneys and their assistants to use effectively.

Using the lessons learned in these first training sessions, the ABF processor design team extensively revised the processor during the nine months from June 1977 through February 1978. The ABF language was made much more flexible and forgiving, and a mechanism was added that detected and explained any errors the processor could not forgive. Considerable time was spent redesigning the processor so that it would overlook harmless errors and would enforce rigid rules only when necessary to keep trainees out of serious trouble.[49]

The redesigned processor was tested in the spring of 1978 by a small group of attorneys and their secretaries at the center. This time the first few sessions dealt solely with word processing. Later lectures focused on drafting simple documents in the ABF language, and the trainees were assigned reading from an expanded and revised ABF primer. Each member of the class was asked to draft one article of a simple form will. With a good deal of assistance from Sprowl, the class created an operative legal practice system that produced a very simple will. (This simple system later served as the point of departure for the more sophisticated will system written by Robert Seibel.) This second experimental training class appeared to be successful. Although the will system was not sufficiently refined to be used for clients, it did produce a will from a client interview, and the clinic staff was quite proud of the system.

[49] We permitted EQUALS and = to be used interchangeably; we permitted THEN to be omitted; and we made numerous similar changes. There were also some language constructs we had permitted in the original prototype that we decided were too dangerous to be used by the trainees. For example, we learned that it was essential to have the computer check for spelling errors in the capitalized ABF language words such as IF and END IF. We learned that the trainees often misspelled these words, and the early version of the processor simply assumed the misspelled words were part of the statements within the brackets that described client data. No error indication was given, but error checking was clearly needed here to assist the trainees. To correct this problem, we forbade the use of capitalized words within brackets except for a limited number of special words such as IF and END IF. The computer was then instructed to assume that any other capitalized word encountered within brackets was misspelled, and it flagged all such words as errors. But we also determined that the computer could safely ignore any spelling errors that occurred beyond the first four letters in special words such as OTHERWISE. The computer could correct such spelling errors without drawing them to the trainee's attention. In many other small ways, we tightened up the error checking where it helped the paralegals and we loosened it up where it simply slowed them down. Feedback from users, essential to making such fine adjustments, was supplied by a group of five law student volunteers who constructed an intestate probate system and a short form 1040 income tax return preparation system during the fall of 1978. Northwestern law student Gregg Maryniak, now a Chicago attorney, was particularly helpful.

IV. Description of the Experiment

Because the preliminary test results were encouraging, we felt that the processor was ready to be tested by attorneys in a law practice environment. In the remainder of this article we present the results of an extensive field test of the ABF processor conducted at the center between 1978 and 1980. We describe our efforts to train the clinic staff to use the ABF technology, and we also describe the five automated systems they ultimately constructed. The divorce system is presented in considerable detail, since it typifies the kind of system we expect this technology to support in the future; our discussion of efficiency and cost effectiveness concentrates on the will system because it was used much more frequently than the others.

A. Early Failures

In the fall of 1978, we initiated our first formal training course for the attorneys and secretaries of the center. We decided to train all of the attorneys and most of the secretaries at the center in the ABF language and in the operation of the computer. We hoped to develop four automated delivery systems within a few months by allowing participating attorneys a reduction in caseload while each worked at creating a delivery system. In addition to the clinic staff, three law students enrolled in the class for credit.

Sprowl gave one classroom lecture each week, and the class was supplied with an extensively revised, 132-page ABF processor primer and other written materials that explained the use of the word processor module, the document-printing module, and a new document storage module.

Following the first lecture, which explored the nature of digital computers generally, the class members created and revised simple computer programs using the standard programming language BASIC and thereby gained familiarity with the terminals, the computer, and the word-processing module. After the second lecture, which focused upon word processing, the class created simple textual documents and learned the document-printing commands that controlled pagination, paragraphing, and centering. Following the third lecture, which introduced the ABF processor and language, the entire class began building a divorce system under the supervision of the clinic's family law specialist, Carolyn Hughes.[50]

[50] A short description of the concepts introduced at the first five classes is as follows:
Class 1: Use of the computer terminal; how to contact a remote computer by telephone; creation, modification and destruction of documents; introductory word-processing commands; introductory BASIC, including programming a computer to solve a simple compound interest problem; and introduction to repetitive operations in BASIC.
Class 2: More discussion of repetitive operations and terminals; additional text revision com-

The law students were required to complete their assignments and master the material presented before they could receive credit. Each was asked to write a portion of the divorce system under Hughes's general supervision. One student worked on the decree; another on an elaborate procedure that selected the appropriate grounds for divorce; and the third on a mechanism to select which of 10 to 15 possible documents were to be prepared for any given client.[51] The attorneys in the class were each carrying full loads of active cases and supervising 8 to 10 students, teaching classes in the clinic, and assisting with clinic administration. These heavy practice and teaching demands competed for their time, and they often were unable to complete their assignments. The attorneys other than Hughes gradually became less active in designing the divorce system. The loss of one staff attorney significantly increased the workload of the others, and the attorneys had even less time for the computer project. The secretaries considered their class assignments part of their normal workload. They were subject to pressures from attorneys with deadlines, and these more immediate pressures often interfered with their ability to complete class assignments on schedule. One secretary, Teresa Stahulak, with help from Hughes and Sprowl, managed to complete work on the divorce petition.[52]

Stahulak and the students ultimately built an operative divorce system that included a petition, a decree, a procedure to select the best ground for divorce, and another procedure to identify which documents should be

mands; introduction to temporary document storage; library storage of documents, and the commands needed to gain access to libraries and to manipulate documents stored in libraries; and introduction to the postprocessor and the commands that control centering, paragraphing, and the like.

Class 3: Introduction to the ABF language; the types of documents a library may contain; generating repetitive passages using the ABF language; explanation of the differences in notation between BASIC and ABF language; repetition-controlling commands; each class member assigned to write a portion of a divorce system, beginning with the complaint or petition.

Class 4: Additional instruction on ABF processor protocols; the formatting of numbers; altering questions; review of the text-editing commands, system commands, and library commands; further work on the postprocessor; and a quiz reviewing all commands and procedures.

Class 5: Discussion and examples of the difference between short names and file names: one being the temporary name that the computer gives a document or record in temporary storage, the other being the name the ABF processor gives a procedure or question; problems were raised by class members, particularly about the use of arrays and variable lists.

[51] The law students had mixed reactions to the project. One student who had previously worked with computers thought it was interesting and fun. She maintained her interest in the field test and later helped design the trust system under Robert Seibel's supervision. The second, who had not previously worked with computers, was neutral. The third, who had been a programmer and had gone to law school to get away from programming computers, reacted quite negatively to the project toward the end but ultimately completed his part. All the trainees found the commands frustrating, as we shall explain.

[52] Stahulak was the only secretary to attend all the class sessions with the attorneys and law students. The others were trained separately to do document entry and revision work, and they were not taught the ABF commands. They were supposed to help the attorneys enter and revise the stored documents.

prepared. Although Hughes had supervised the development of the system, she had given the class members considerable freedom to be creative. Hughes later found that the system was not well suited to serving the needs of her clients—the client interview was too long and time consuming, and many of the necessary documents had not yet been drafted. As a teaching exercise for Hughes, the students, and Stahulak, the project was a success. But when viewed as an effort to design a practical, useful system, it was a failure.

The project also failed to sustain the interest and commitment of the trainees. Most of the attorneys were drawn away from the computer by their responsibilities to serve clients and teach classes; some of them also viewed the computer as too difficult to use and not sufficiently helpful to their practice. Hughes and Staudt were the only attorneys who emerged from the lecture series fully trained. There was considerable frustration among all the trainees, caused by the confusing sets of commands they had to master. There were ABF processor commands, document storage commands for three separate computer storage areas, word-processing commands, and printer control commands. There was no logical structure to these command sets that the trainees could carry over from one set to the next. Each command set had to be learned in isolation, and all had to be mastéred before any automated system could be constructed. The trainees had great difficulty remembering the commands and keeping them straight, and they found their lecture notes and manuals cumbersome and slow as memory aids.

We were faced in December 1978 with frustration and loss of interest. We had completed one system, but it was not very satisfactory. There was a distinct possibility that the experiment would end in failure if nothing were done to reduce the trainees' frustration level.[53]

[53] A further explanation is needed of why those trained in the fall of 1978 had more difficulty with the commands than did those trained the preceding spring. The Northwestern word-processing module included an excellent mechanism for storing documents in separate "libraries." Accordingly, we designed the ABF processor to fetch documents directly from these word processor libraries. Form documents stored in any such word processor library could thus be processed by the ABF processor directly with no complications. We used this library mechanism in the spring of 1978, and the trainees had no difficulty with it.

Unfortunately, as the number of documents stored in the word processor libraries grew, our use of this library mechanism became very costly. The storage charge for a document stored in one of the word processor libraries was 25 cents per day, no matter how small the document was. Since our law practice system libraries often contained from 50 to 120 documents and since many of them were only one to three lines long, we soon found that our monthly charge for storing documents on the computer exceeded $500. Document storage charges were rapidly using up our rather limited project funds. The word processor's library mechanism also caused each document, if it was not used for several weeks, to be pulled off the computer and archived individually on a reel of magnetic tape. Thus, when an automated legal practice system had not been used for several weeks, its documents and procedures would all be archived, and it might take the Northwestern computer an entire morning to retrieve all the individual documents and procedures of the system from the many separate reels of magnetic tape, each of which had to be hand mounted on the computer.

We were thus forced to substitute a new document storage module of our own design for the word

B. Revising the Commands

In December 1978 and January 1979, Sprowl, with the assistance of computer scientists John Norstadt and David Mausner of Northwestern University's Vogelback Computing Center, designed an entirely new *command module* and interposed it between the trainees and all of the existing modules.[54] This new module did not change the underlying computer programs in any way. It simply presented the trainees with a much simpler and better organized set of commands that they could use to drive all of the underlying modules. The command module accepts a simplified command from a trainee and then prompts the trainee to supply any necessary additional information. After receiving the information, the command module generates the more complex commands that are needed to drive the other modules. The command module thus forms a protective shell within which the trainee can function using a very limited number of simplified commands. The trainee, gaining confidence and experience, gradually learns additional, more powerful commands that advance the work more rapidly and effectively.

To facilitate the learning, we also constructed a very simple computer-aided instructional module as a part of the new command module. This instructional module provides the trainee with lists of commands and

processor library mechanism. The new module stored all the documents relating to a single legal practice system together as if they were a single document. Hence, the 25 cents per day charge applied to an entire set of documents, rather than to each individual document. Then if a set of documents was not used for several weeks, all the documents in the set were archived together on a single reel of tape from which the entire set could later be retrieved in only 10 to 15 minutes. But the commands needed to operate the new document storage module were considerably more complex than were the commands associated with the word processor library mechanism. Those trained in the fall of 1978 were thus required to master a more complicated set of commands than those the trainees had used the preceding spring.

[54] Those with computer experience may be interested to know just how this command module functions: Norstadt was charged with maintaining Northwestern's word processor module at the time of this experiment. To increase its flexibility, he integrated that module with a separate interactive text revision module designed by Mausner to enable those using the word processor module to initiate the execution of tasks performed by other modules within the computer. Using this mechanism together with a separate mechanism within the word processor module that is normally used to initiate the translation of English-like computer programs into numeric form, Sprowl caused the text-editing command GO to prompt the operator by displaying COMMAND? and to accept whatever single-word command the operator supplied in response. The computer then checked to see if the single-word command corresponded to the name of a document in a special library of command documents. If a document was found whose name was the same as the command supplied, this document was delivered to Mausner's interactive text revision module. Questions were generated, and the answers supplied controlled the revision of the document. After it was revised, the document took over control of the computer and caused it to carry out a series of tasks. All one had to do to create a new command was to prepare a document and store it in the special library of command documents, assigning to the document a name that then became the name of the new command. Each document in the command library contained a mixture of text-editing commands, commands to the computer's operating system calling for control of the computer to be turned over to various modules, and special text revision commands that caused Mausner's program to ask questions and then use the answers supplied to control the revision of the document.

The command module was not particularly efficient, but it made an excellent breadboard structure for testing different arrangements of the commands as the experiment progressed.

brief explanations of each command, and it eliminates the need to refer to a written manual or to lecture notes. Lectures are valuable because they give the trainees an overview understanding of what they are doing, but lectures by themselves are not sufficient to teach the commands. We learned that trainees remembered a new command better when they were taught the command at a time when they needed it to solve a problem, and the computer-aided instructional module facilitated this method of learning.

When first installed, the new command module was empty in the sense that it contained no commands other than those associated with each of the other modules within the computer. But it required only minutes for a computer specialist to add a new command to the module or to modify any existing command. A single command could be designed to replace literally hundreds that might otherwise have to be used. New and simplified commands could thus be added to the command module whenever the need for them arose, and the new commands could be molded to the specialized needs of the center. We were thus not locked into a prepackaged set of commands designed by individuals who had no knowledge of the center's special needs. Within a few months, the command module contained almost 50 new commands. As the experiment progressed we frequently revised the commands and the accompanying prompts.

Consider, for example, the task of retrieving documents from storage. The Northwestern University computer has three storage areas: a *temporary* storage area, a *permanent* storage area, and a *library* storage area. Without the command module, separate commands had been required to pull documents from each area. To pull a document named TRUST from temporary storage, one typed READ TRUST; to pull a document named TRUST from permanent storage, one typed GET TRUST; and to pull a document named TRUST from a library named SMITH, one typed

```
$ ATTACH, LIBRARY, KENT-SMITH, PW=XYZ
$ EXTRACT. LIBRARY, TRUST
READ TRUST
```

All of these commands were replaced by a single FETCH command that elicits the following prompts for additional information:

Computer prompt:	DOCUMENT NAME?
User response:	TRUST
Computer prompt:	WHERE STORED (T=TEMP, P=PERM, L=LIB)?
User response:	L
Computer prompt:	LIBRARY NAME?
User response:	SMITH

Beginners now need only the single command FETCH. Persons who spend many hours working with the computer eventually learn to use the original commands GET and READ also, because these commands are faster than FETCH for obtaining documents from temporary and permanent storage. Experienced users thus use three commands where the beginning trainee needs only one.

Our command module forms a series of nested, protective spheres about the trainee. The beginning trainee remains within the innermost sphere where there are just a few simple commands. As the beginner becomes proficient in those commands, he or she is encouraged to learn additional, more complex and powerful commands as the desire for speed and flexibility becomes more important than the desire for simplicity and security.[55]

C. Later Successes

In early 1979, we decided to complete only two of the anticipated systems over the following five months—a divorce system and a will system. An attorney and a legal secretary were assigned to work together as a design team for each system. We arranged larger caseload reductions for these teams, and we initiated a new round of training to introduce the entire staff to the new command module and new staff members to the computer. We decided to try an entirely different training regimen. We offered the trainees two lectures covering the terminals, the simplified commands, and the ABF processor. Each trainee was also given an ABF language primer. The remainder of the training was conducted through informal, supervised laboratory sessions while the trainees worked in pairs on assigned projects. All instruction and assistance focused on the specific legal practice system the team was constructing, and there were no additional general lectures or assigned homework.[56]

[55] One legal secretary, Teresa Stahulak, ultimately used many sophisticated commands commonly used only by skilled programmers. She now opens and closes computer accounts, arranges for making tape backup copies of files, and generally functions as the computer expert for the center even though she has never received any formal training in computer science. The attorneys and legal secretaries who participated actively in the design of automated delivery systems reached intermediate levels of proficiency that reflected their enthusiasm, their talent, and the amount of time they spent working with the computer. The remaining staff members, who used the computer only to produce client documents, remained securely within the innermost protective sphere where they seemed to function very well. Our trainees thus sorted themselves out in a voluntary manner into a variety of proficiency categories, and the computer's command module was flexible and powerful enough to meet the needs of users within each category.

[56] During the first lecture, Sprowl explained how to use the terminals; introduced the new commands and the new computer-aided instruction facility; and discussed word processing in a very summary fashion, covering only the most basic word-processing commands. The trainees were then immediately sent to their terminals to begin their first laboratory session. One week later, Sprowl gave a slightly more elaborate explanation of word processing and of the commands for retrieving, revising, storing, and printing documents, and he introduced the class of the ABF processor and handed out copies of the ABF primer. Then once again, the trainees went directly to terminals to work on their assigned automated legal practice systems.

1. The Divorce System

When Hughes and her assistant Stahulak began a revision of the divorce system, they found themselves confronted with an entirely new, simplified set of commands. After having worked hard in earlier training sessions to master the old commands, both resisted using the simplified commands. They reacted much the way a legal researcher would react if the West key number system were replaced by an entirely new, unfamiliar system. Stahulak eventually began using these new commands that simplified her work with the processor. Hughes continued to supervise the development of the divorce system, but she never really tried to learn the new commands. She permitted her assistants to do the computer work, while she reviewed and corrected printed copies of the documents, procedures, and new questions that they created. She thus maintained substantive control over the creation of the divorce system without continuing as an operator.

The divorce system was completed successfully, but not within five months. Hughes studied the system created by the class the previous fall and used it as the departure point for the system she designed. She scrapped the entire grounds selection procedure, the elaborate document selection mechanism, and the student-drafted decree. She revised the petition to include provisions for marital property and for adopted children. Recognizing that many of the passages of the petition and the decree could be identical, she had those passages created as separate small documents that were incorporated by reference into both the petition and the decree. The large number of questions requiring a *yes* or *no* answer unduly protracted the client interview. She added computational procedures that substituted a smaller number of questions for those the processor delivered directly from the documents, without altering the divorce documents. The following is an example of the replacement questions generated by the new procedures:

> Please type the letter that corresponds to the statement most accurately reflecting the respondent's employment status:
>
> A. Respondent is employed;
> B. Respondent is self-employed;
> C. Respondent is retired;
> D. Respondent is unemployed;
> E. Respondent's employment status is unknown.

Hughes also added a number of additional documents that are often needed to complete a divorce case. Each form document is stored as a separate entity within the computer, and all the necessary documents are drafted by the ABF processor in a single interview session during which no question is ever asked more than once. One set of procedures guides the automatic insertion of the proper information for *petitioner* and *respondent* slots in some documents and *husband* and *wife* slots in other documents according to the petitioner's sex, and another set of procedures guides the automatic selection of personal pronouns and adjectives. All the computational procedures are shared by all the documents.

The divorce system exemplifies the type of automated legal practice system that we believe will soon become quite popular with attorneys who practice family, real estate, corporate, and tax law. Despite the complex function of the system, it is neatly structured and simple to audit. Its elements can be easily read and understood by attorneys and legal secretaries. Also revisions are quickly and easily accomplished by revising the procedures that guide the system or by revising the text of the documents, as the need arises. (Examples taken from this system can be found in the appendix.)

Although Hughes left the clinic in December 1979, her divorce system was finished to her specifications in May 1980, and it is now fully operational. The default divorce system is the most complicated system yet implemented using the ABF technology. Because Hughes had not worked with computers before this experiment, her experience demonstrates that a practicing attorney, with some support from a computer specialist, can master the ABF language well enough to supervise the design of an automated legal practice system. After Hughes left the clinic, none of the staff attorneys were interested in family law and divorce practice, so the clinic's caseload in this area fell off sharply. The system has been used only on rare occasions for clients.

2. *The Will System*

In January 1979 Robert Seibel, an experienced estate planner, joined the center staff.[57] He had never before worked with computers. From the inception of discussions about this experiment, we had intended to develop a simple estate-planning system to serve the needs of the center's senior citizen clients. Seibel came to the project enthusiastic at the prospect of designing an automated will-drafting system.

He began by revising the simple will system the center's staff had written the year before.[58] By April, he had written a much more sophisticated

[57] From 1972 through the end of 1978 Seibel was employed as an associate at Ropes & Gray, a large Boston law firm. His practice there was concentrated in estate planning for moderate and large estates. Mark Neil, a Chicago attorney with estate-planning experience, assisted Seibel in the early weeks to draft language appropriate for Illinois wills.

[58] The legal secretary assigned to work with Seibel on the will system was also a new addition and had not participated in the fall computer classes. Therefore, both members of the team assigned to set up the will assembly system started in January 1979 with no computer experience and no ABF processor experience. They worked as a team for several months until the secretary left the clinic to be married, after which time Seibel did his own work with the computer, assisted by a number of other legal secretaries.

system that includes the capacity to provide for single, multiple, or corporate executors; provisions for guardianships of minor children; provisions for a trust for minor children to hold property until they reach a certain age; provisions for cash gifts and specific gifts of tangible property; and a wide variety of incorporated available options for the disposition of tangible personal property and the residue of the estate. Seibel often modified and expanded the will system during the 14 months that followed, improving its operation and thus its suitability for the specialized needs of the clinic's clients. By May 1979, the system was drafting nearly 30 wills per month for clients of the center.

Seibel's system is still in active use at the center, even though Seibel left to become the director of the clinical law program at Drake University in June 1980.[59] Even attorneys who have received minimal training can sit down at a terminal, turn on the system, type in the answers to the questions asked, and supervise the printing of a will. The entire process can be accomplished within 15 minutes without the assistance of a typist. Seibel himself produced more than 35 wills in the evenings and on weekends when secretaries were not available to operate the system; other clinic attorneys have operated the system during off hours as well. But most of the wills are produced by Vina Sisney, a legal secretary who, after an hour of instruction on her first day of work, took over the clerical aspects of automatic will production.

3. *Other Systems*

Seibel, Chicago attorney Mark Neil, and law student Carol Vogel began to design a trust system in the spring of 1979. Sprowl eventually completed the trust system under Seibel's supervision in November 1980. The trust system matches the sophistication of the divorce system. For example, to aid the attorney in selecting which powers to grant the trustee, the system first displays a menu of the standard powers and asks the attorney to type out the numbers of those standard powers he or she does *not* want to include. Then it displays a menu of nonstandard powers and asks the attorney to type out the numbers of those nonstandard powers he or she *does* want to include. Elaborate mechanisms are provided to allow the same basic trust to be used with single, multiple, and institutional trustees, including procedures for selecting personal pronouns and verb forms to match the trustee or trustees. The trust system has not been used frequently, since most of the center's clients do not have sufficient assets to warrant the use of a trust in their estate planning.

Seibel believes he could combine his will and trust systems with additional estate- and tax-planning elements to provide an estate-planning

[59] In July 1981, Seibel moved to the clinical law faculty of the University of Maine.

system that would satisfy the needs of most attorneys in private practice. He suggests adding optional lessons to such a system to provide training for neophyte estate planners and refreshers for veteran estate planners. Compatible probate form and tax return preparation systems could also be designed and used in conjunction with such an estate-planning system.

Two other demonstration systems were written. Seibel designed a guardianship petition system that assembled the forms needed to have a guardian appointed for a minor child.[60] A paralegal trainee initially worked on the design of this system. It became apparent that automatic printing of the document heading for multiple parties required Sprowl's expertise. Otherwise, the system was relatively simple to write. The guardianship system was used successfully for more than 80 clients in 1979. In early 1980, however, a new probate judge expressed a strong preference for court-printed forms, which were difficult to prepare using the computer because of their irregular printing alignment and poor design. As a result, use of the guardianship system was discontinued when the new judge assumed control of the guardianship docket.[61]

Sprowl designed a real estate closing agreement system, which asks a series of questions and then prepares the closing statement for a sale of real estate. In addition to prorating taxes and rent automatically, this system produces a client file that summarizes the data supplied by the client. If the date of the closing changes, a word-processing command can correct the date in the client file, and the ABF processor can then produce an entirely new closing statement that reflects the revised closing date. A completely revised closing statement with accurate prorations can thus be produced in one or two minutes. A Northwestern law student later added several form deeds to this emerging real estate system.

D. Economic Considerations

It is not possible to determine from the data gathered during this exper-

[60] For a variety of reasons, low-income individuals in Cook County, Illinois, sought to obtain the transfer of legal custody of children for short periods of time. For example, such a transfer is necessary when a sole surviving parent wishes to enter military service and cannot take the child to the military base. The probate court in Cook County was unable to find lawyers to represent individuals who wished to have such uncontested procedures completed. Therefore, in conjunction with the local legal aid program, the Legal Services Center agreed to represent individuals seeking uncontested guardianships on referral from the probate court. During 1979 our automated guardianship system was used for 70 clients.

[61] This system was extremely simple and did not require sophisticated or complex clause selection or document drafting. The text of the documents was short and concise so that automated printing did not offer great advantages over manual typing. Only four documents were produced, none of which exceeded one page. In fact, as described in the text, the only difficult aspect of this system was designing instructions to the computer's document-printing module that prepared the document caption for multiple parties. The difficulty was not attributable to the ABF system but rather to the printing module, which had no provision for preparing such a caption. Sprowl used the ABF system to compensate for the deficiencies in the printing module.

iment whether the ABF processor reduces the costs of providing legal services to clients because we did not attempt to compare the cost of delivering the same services manually and with the assistance of the computer. The data we did gather illustrate our cost to prepare wills for senior citizen clients. The other systems were not used enough to generate meaningful cost data.

We used the computer to maintain records of the attorney time spent in developing the will system, but interviews with Seibel following the experiment revealed that his time records may not be very accurate.[62] However, even if we generously overestimate the time that it took Seibel to develop the system, the total did not exceed 200 hours. Accordingly, setting an arbitrary value of $50 per hour on his time, the attorney cost of developing the will system was less than $10,000. The system was used approximately 500 times during the 18 months following its creation. Amortizing the attorney cost over this period results in a base cost of $20 for each will produced during that period. Equipment rental and computer expenditures during the development of the system add approximately $3 or $4 per will. The cost for all computer operations necessary to produce one client-customized will is between $5 and $8 each time a will is produced. Adding these figures together, each use of the automated will system during the first 18 months cost the center between $28 and $32. Additional attorney time is necessary to maintain the system to reflect current legal developments. If this work can be done for $1,200 per year, and if 300 wills are drafted each year, then the cost per will in future years will fall to about $12 to $16 per will.

Some additional factors should be considered which might alter these cost figures. First, our computer was a not-for-profit computer, and our rates were less than a private firm might have to pay. But the prototype ABF processor is, in some respects, an inefficient prototype that caused computer charges to be larger than they would be with a commercial version of the same processor. These two factors balance out to some degree. If the computer used had been an in-house computer rather than the remote Northwestern computer, the computer cost figures would have been different.[63] But attorney costs are the dominant costs. Second,

[62] Seibel's time sheets indicate that from January through August 1979, he spent approximately 135 hours in writing, rewriting, testing, and debugging ABF computer systems. During February, March, and April much of this time was spent working on the will system. Some of his time in the 8 months was spent on the guardianship and trust systems. In May 1980, Seibel stated that he probably spent more time than was indicated on his time sheets working with the computer, but he agreed that 200 hours is a reasonable outer limit on the billable time he spent working on the will system.

[63] With an in-house computer, the costs could have been either larger or smaller, depending in part on how many other tasks the shared computer could be used to accomplish. The cost of computing power is dropping rapidly, and within three to five years it is likely that a computer as powerful (for our purposes) as Northwestern's can be purchased for $15,000 to $30,000 (in 1981 dollars).

the above figures do not include the attorney and secretarial time required to review and revise the final wills, counsel the clients, and execute the wills produced. To recover their full costs, the center would clearly have to charge more than the above amounts.

Finally, the above analysis does not reflect some important benefits that flow from use of the computer. The computer system enables newly hired legal secretaries to produce wills after less than an hour of training, and it has enabled clinic staff attorneys who are not estate-planning specialists to do quality estate-planning work (within the limits of the system) for moderate-income senior citizens. Seibel's system helps to ensure the quality of their product, even though Seibel is no longer in residence. It is difficult to assign a monetary value to these improvements in training speed and work quality. In addition, assuming the system has saved just one hour of attorney time for each will produced, it has already paid for itself twice.

V. CONCLUSION

During the experiment we learned many lessons about training lawyers in the use of computer technology; many of our preconceived ideas about training proved to be incorrect. We had assumed that the beginning trainees could use the same commands used by experienced system designers. We learned, however, that beginning trainees needed simplified commands and computer prompts for additional information. We initially gave our trainees lectures and supplied them with instruction manuals. The experiment demonstrated that trainees learned better, faster, and with less frustration in a supervised laboratory setting while working on the design of actual systems and when supplied with computer-aided instructional tools. Trainees should also be supplied with model systems to serve as sources of ideas and as departure points for their design efforts. The ABF language is so flexible that it gives the trainees no guidance on how to proceed; it leaves them in the same position as apprentice masons who understand bricks and mortar but do not know how to build a house.[64] The systems that were developed are an important product of this experiment, for they will serve as initiating models for future trainees.

We also developed a better understanding of how a law office processor should be designed. At the start we assumed that the ABF processor could be used in conjunction with any word processor; we discovered that the

[64] This phenomenon is illustrated by comparing Hughes's and Seibel's experiences in developing their systems. Seibel was supplied with a model system, and he successfully used that model as the departure point for his own successful design efforts. Hughes was not initially supplied with a model system, and her first attempt to design the divorce system was unsuccessful. Her second attempt succeeded partly because she had available the unsuccessful first system as a departure point. A number of student projects not described here also demonstrated the importance of a model system.

two must be carefully integrated into a single system that is responsive to a simple, logical set of commands. Many other technical revisions of the processor were inspired by this experiment. Additional revisions will be made in a new version of the processor now in the planning stage.

Three recent developments lead us to believe that systems of the general type described here will soon become widely available to the legal profession. First, West Publishing Company is now permitting manufacturers to design word-processing microcomputers that may be used as remote office terminals for the WESTLAW system of computer-assisted legal research. Encouraging the manufacturers of such systems to equip them with a standard telephone communication protocol,[65] West may have opened up a new way to market a wide variety of services to law firms of all sizes. Even a small law firm may soon be able to afford a West-compatible word-processing system and, by connecting it to a telephone, gain access to the local law school's ABF will assembly system, West's computer-assisted legal research system, Shepard's Citation Service, a commercial law office accounting system, the local courthouse's automated docket book and motion-filing system, and a variety of other comparable services.

Second, Mead Data Central has entered into a joint venture with publisher Matthew Bender & Company to establish an automated estate-planning system as part of Mead's LEXIS system of computer-assisted legal research. LEXIS may soon add automated law practice systems to its other services. Both Matthew Bender and Warren, Gorham, & Lamont, Inc., are now marketing estate-planning systems designed for magnetic card-controlled automatic typewriters. These magnetic card systems represent a step toward the development of fully automated systems similar to those described in this article.

Third, several projects are under way that parallel the ABF technology. A group headed by Fred D. Fagg, John Abele, Peter Bergsman, and Peter S. Nycum at Lewis and Clark College in Portland, Oregon, have recently completed a second-generation ABF processor considerably more efficient than our prototype processor, and they hope to install it soon on a microcomputer for in-office use.[66] Another group, led by Larry C. Farmer at

[65] The actual "protocol" or convention for transmitting data over telephone lines has been standardized for some time. West has adopted additional protocols defining commands that cause the remote terminal to perform different tasks such as writing on the terminal screen at selected locations or erasing at selected locations. West has adopted the command protocol of the Hazeltine Model 1520 terminal as its standard protocol. Perhaps a better protocol would be that recently promulgated by the American National Standards Institute.

[66] This is the PCUP project (Professional Computer Utilization Project) funded by the Murdock Charitable Trust. Sprowl spent the summer of 1979 in Portland training the PCUP team in the ABF technology. The PCUP software is written in the language PASCAL and is presently installed on a PDP 11/780 (VAX) minicomputer manufactured by Digital Equipment Corporation. The PCUP team hopes to install this system on a much smaller microcomputer in the near future.

Brigham Young University's J. Reuben Clark Law School in Provo, Utah, has designed a prototype law office support system accessible via telephone from word-processing systems.[67] The American Bar Foundation's prototype processor has recently been installed on a large, IBM-compatible computer at the University of Michigan by Layman E. Allen and Charles S. Saxon, where it will soon be used for instructing law students and for experimentation.[68] And a number of other commercial and academic efforts are now under way to develop automated practice systems or to develop computers that can support such systems.[69]

All of this leads us to predict that attorneys will soon be able to purchase prepackaged, fully computerized law practice systems of the kind we have developed and tested, and that these will help them serve their clients more efficiently and more competently than they can at present. And this experiment has demonstrated that it is possible for practicing attorneys to design such systems themselves.

[67] This excellent system is intended to demonstrate the "human engineering" principles that must be incorporated into the law office computer systems of the future. Farmer is a psychologist as well as a law professor.

[68] Saxon, a professor at Eastern Michigan University, is an attorney as well as a computer scientist. He rewrote the "machine language" portions of the Northwestern ABF prototype in FORTRAN and revised the remaining FORTRAN programs to be compatible with IBM computers. The Ann Arbor system is an all-FORTRAN system that runs under the MTS (Michigan Terminal System) operating system on an Amdahl computer. It is almost finished as of this writing.

[69] See, e.g., William E. Boyd & Charles S. Saxon, The A-9: A Program for Drafting Security Agreements Under Article 9 of the Uniform Commercial Code, 1981 A.B.F. Res. J. 637, where the authors describe how they used a commercial law office microcomputer marketed by LMS, Inc., Ann Arbor, Michigan, to automate the assembly of complex security agreements. See also Hellawell, *supra* note 30, where the author describes his computer program that helps analyze the tax consequences of corporate stock redemptions, taking into account the constructive ownership rules of Internal Revenue Code § 318. The program is written in BASIC-PLUS-2 to run on a Digital Equipment Corporation DECSYSTEM-20 computer.

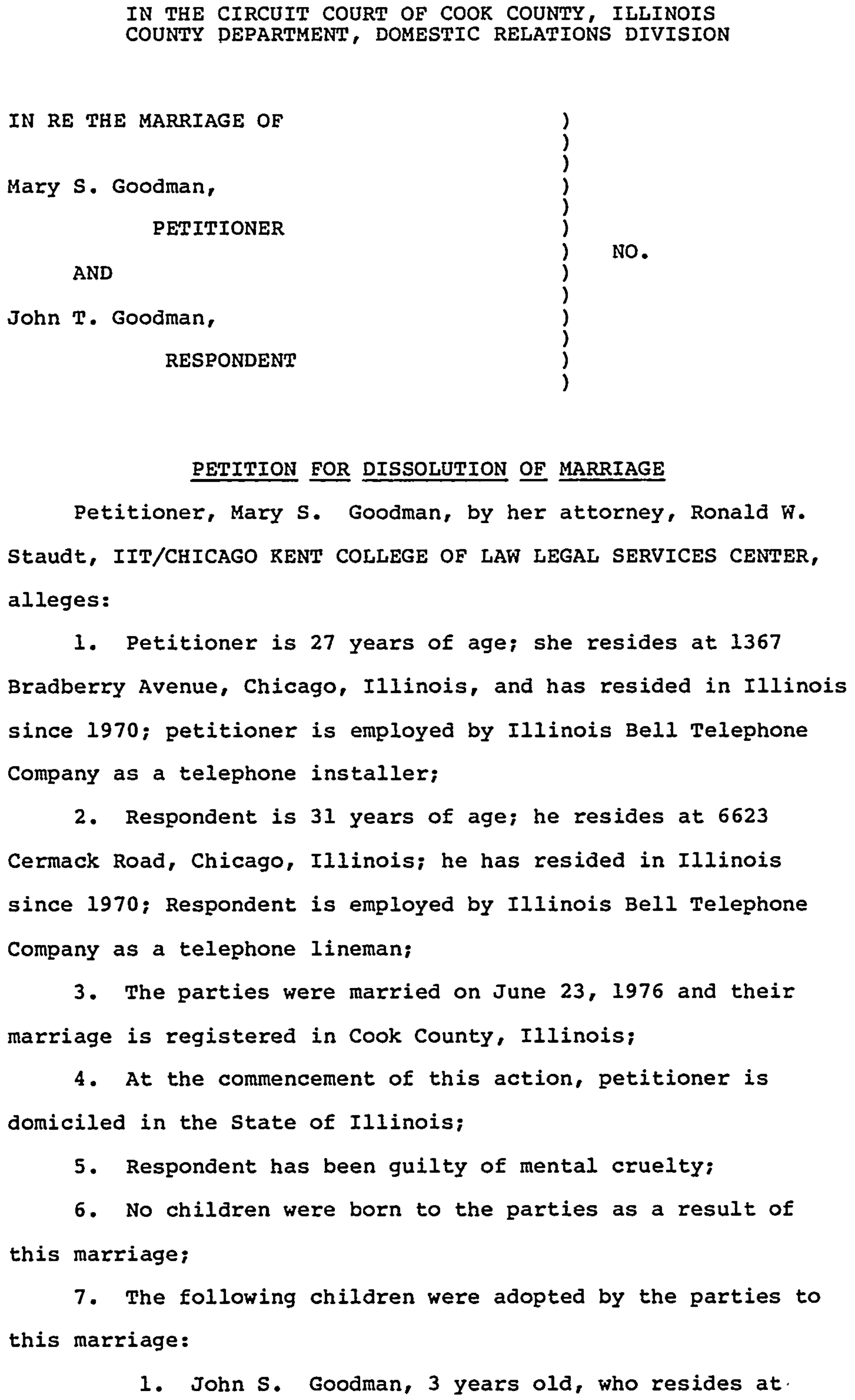
Appendix: Examples Taken from the Divorce System

The automated divorce document production system is the most complicated system constructed during the experiment. All the elements of this system are stored together in a document library named DIVORCE. Included in this library are 8 major documents, 18 passages that are incorporated by reference into 1 or more of the major documents, 14 "new question" documents, 63 computational procedures, 1 data file, and 1 document that controls the selection of which major documents are to be prepared for a given client. Following an interactive client interview, this system can produce a divorce petition, a divorce decree, a State of Illinois divorce certificate, an affidavit of nonsolicitation, an affidavit of military service, an order of default, and either an affidavit of service by publication or a summons, whichever is required. A complete printed copy of the entire divorce system is 42 pages long.

There is no point in reproducing the entire divorce system here. The examples presented below all relate to the assembly of the divorce petition, one of the eight major documents this system is designed to produce.

A. The Petition for Dissolution of Marriage

The petition that follows was produced automatically by the computer following a client interview during which the computer asked questions, and answers to the questions were typed back into the computer:



IN THE CIRCUIT COURT OF COOK COUNTY, ILLINOIS
COUNTY DEPARTMENT, DOMESTIC RELATIONS DIVISION

IN RE THE MARRIAGE OF)
)
)
Mary S. Goodman,)
)
PETITIONER)
) NO.
AND)
)
John T. Goodman,)
)
RESPONDENT)
)

PETITION FOR DISSOLUTION OF MARRIAGE

Petitioner, Mary S. Goodman, by her attorney, Ronald W. Staudt, IIT/CHICAGO KENT COLLEGE OF LAW LEGAL SERVICES CENTER, alleges:

1. Petitioner is 27 years of age; she resides at 1367 Bradberry Avenue, Chicago, Illinois, and has resided in Illinois since 1970; petitioner is employed by Illinois Bell Telephone Company as a telephone installer;

2. Respondent is 31 years of age; he resides at 6623 Cermack Road, Chicago, Illinois; he has resided in Illinois since 1970; Respondent is employed by Illinois Bell Telephone Company as a telephone lineman;

3. The parties were married on June 23, 1976 and their marriage is registered in Cook County, Illinois;

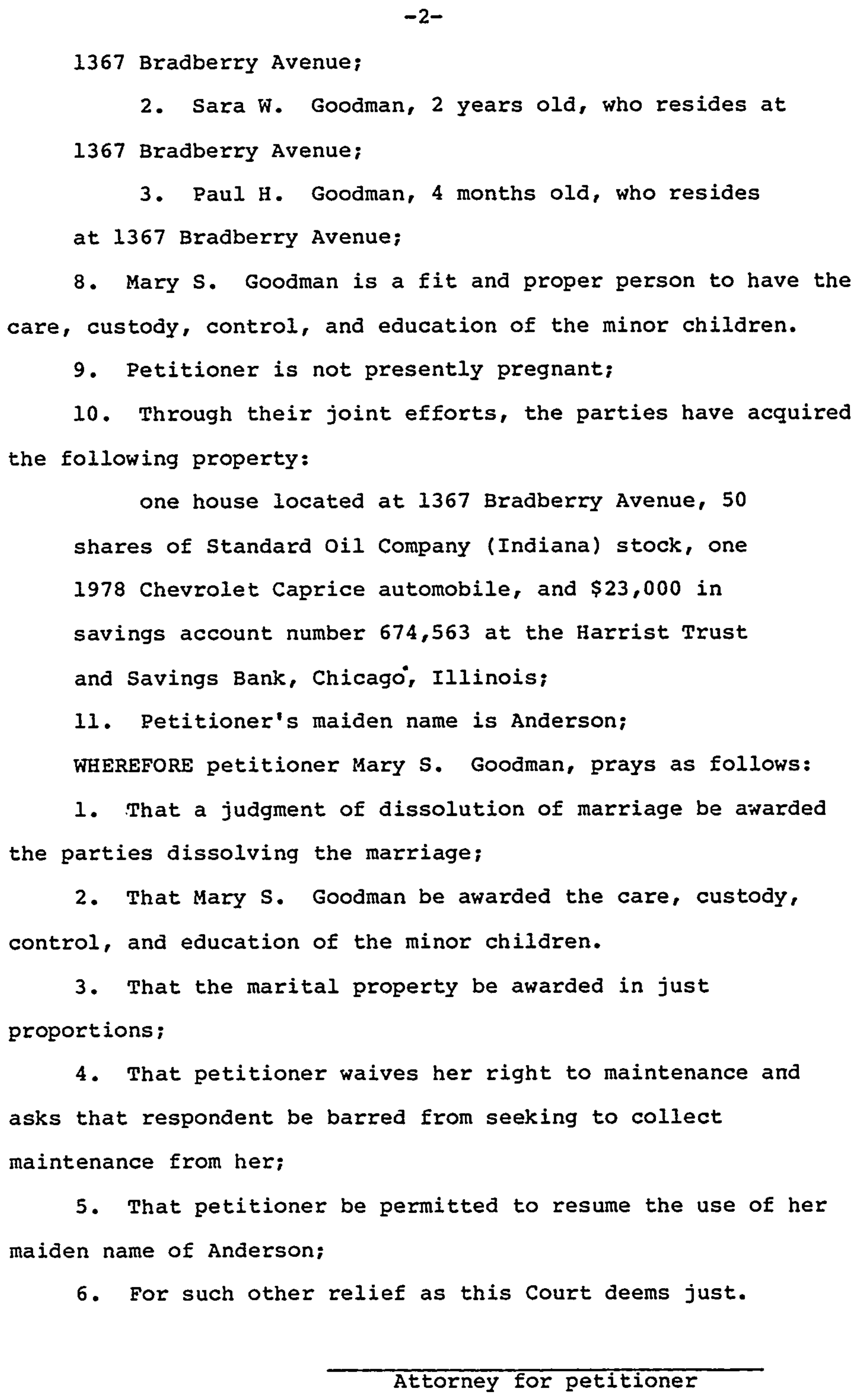
4. At the commencement of this action, petitioner is domiciled in the State of Illinois;

5. Respondent has been guilty of mental cruelty;

6. No children were born to the parties as a result of this marriage;

7. The following children were adopted by the parties to this marriage:

1. John S. Goodman, 3 years old, who resides at



-2-

1367 Bradberry Avenue;

2. Sara W. Goodman, 2 years old, who resides at 1367 Bradberry Avenue;

3. Paul H. Goodman, 4 months old, who resides at 1367 Bradberry Avenue;

8. Mary S. Goodman is a fit and proper person to have the care, custody, control, and education of the minor children.

9. Petitioner is not presently pregnant;

10. Through their joint efforts, the parties have acquired the following property:

one house located at 1367 Bradberry Avenue, 50 shares of Standard Oil Company (Indiana) stock, one 1978 Chevrolet Caprice automobile, and $23,000 in savings account number 674,563 at the Harrist Trust and Savings Bank, Chicago, Illinois;

11. Petitioner's maiden name is Anderson;

WHEREFORE petitioner Mary S. Goodman, prays as follows:

1. That a judgment of dissolution of marriage be awarded the parties dissolving the marriage;

2. That Mary S. Goodman be awarded the care, custody, control, and education of the minor children.

3. That the marital property be awarded in just proportions;

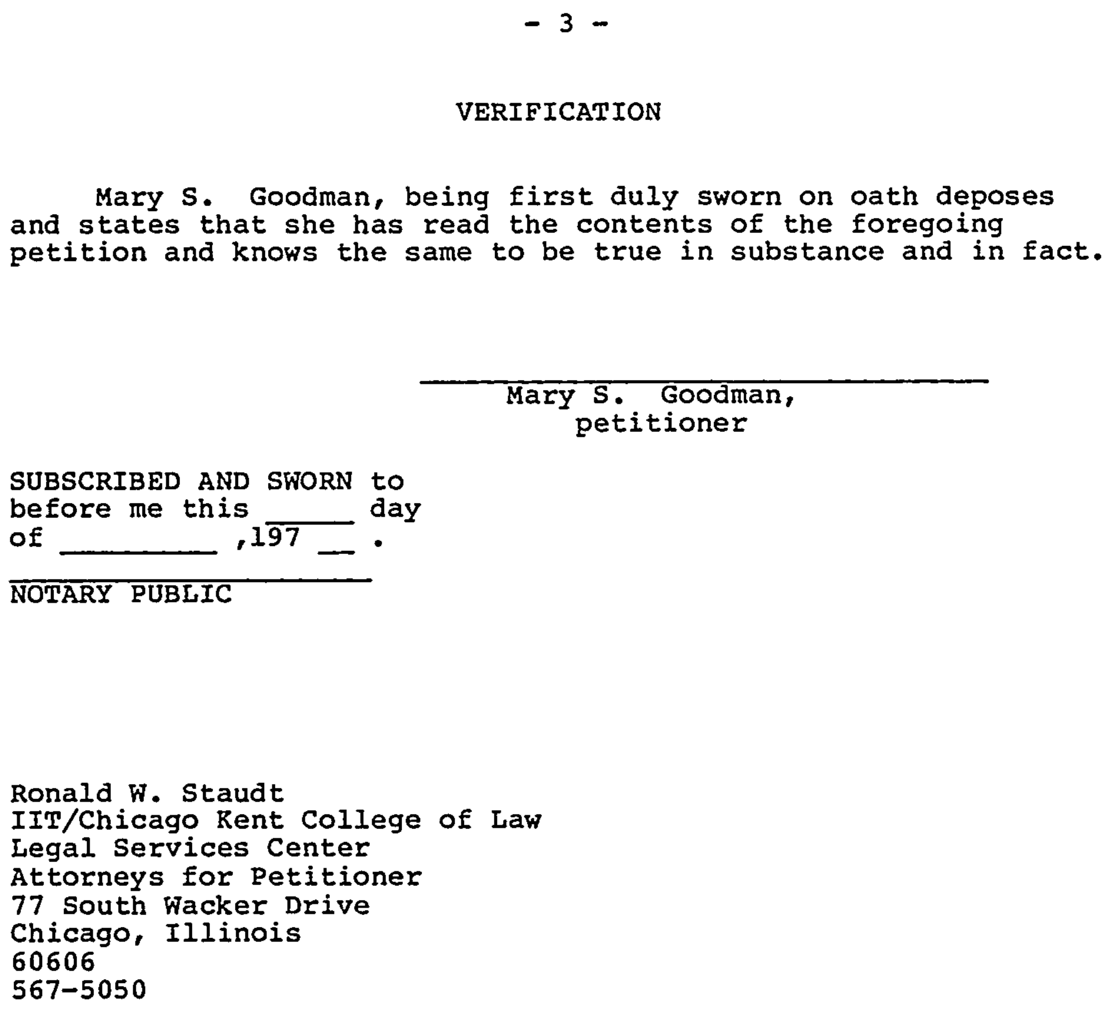
4. That petitioner waives her right to maintenance and asks that respondent be barred from seeking to collect maintenance from her;

5. That petitioner be permitted to resume the use of her maiden name of Anderson;

6. For such other relief as this Court deems just.

____

Attorney for petitioner



- 3 -

VERIFICATION

Mary S. Goodman, being first duly sworn on oath deposes and states that she has read the contents of the foregoing petition and knows the same to be true in substance and in fact.

____
Mary S. Goodman,
petitioner

SUBSCRIBED AND SWORN to
before me this _____ day
of _________ ,197 __ .

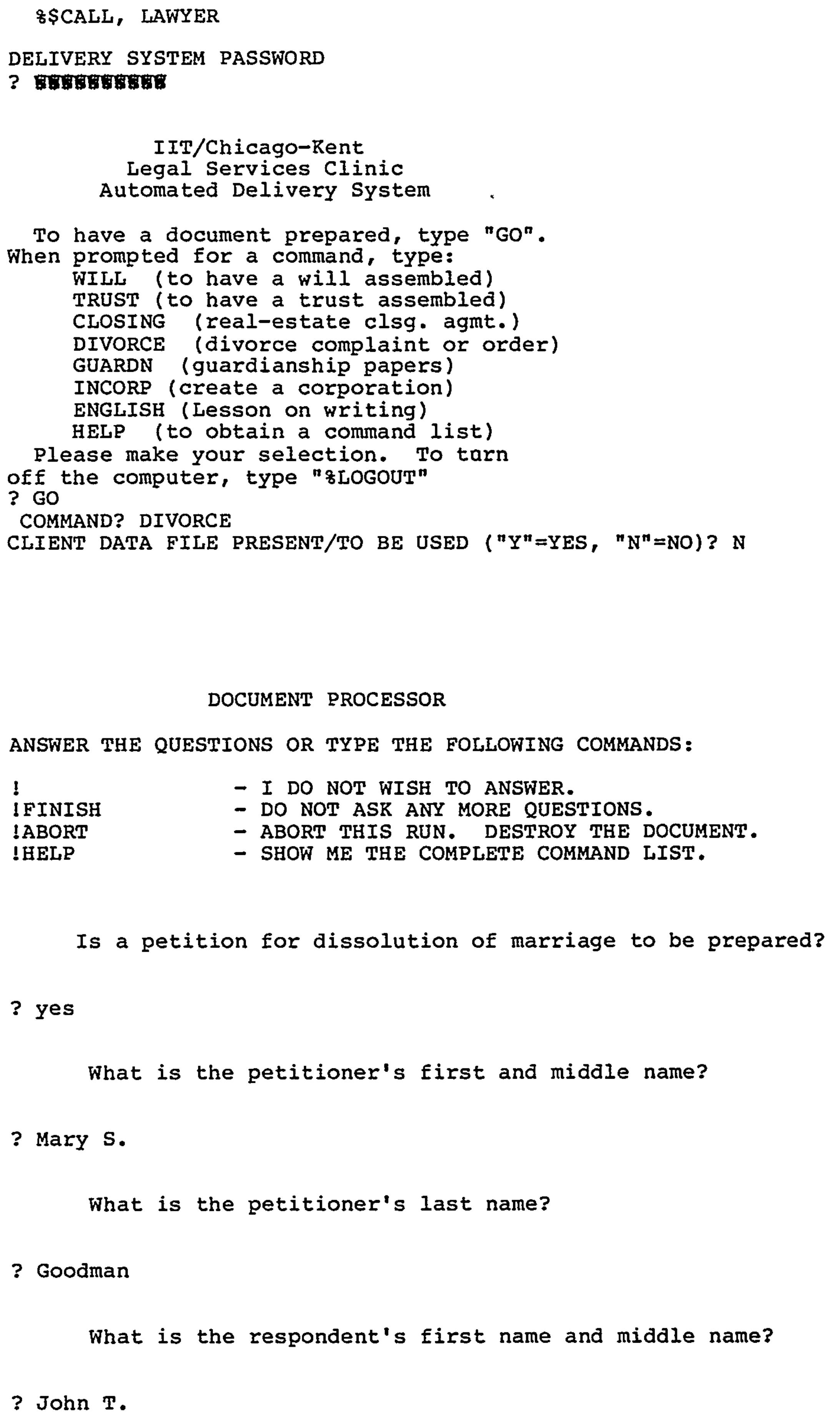
____
NOTARY PUBLIC

Ronald W. Staudt
IIT/Chicago Kent College of Law
Legal Services Center
Attorneys for Petitioner
77 South Wacker Drive
Chicago, Illinois
60606
567-5050

B. The Client Interview

The client interview that resulted in the production of the above petition is set forth below. The questions were typed out by the computer, and the answers to the questions were typed by a legal secretary. (Each answer is preceded by a question mark.)

To place the automated delivery system into operation, the legal secretary first types %$ CALL, LAWYER. The computer then asks the legal secretary to type in the DELIVERY SYSTEM PASSWORD, and the computer prints Xs over this password. Next, the computer explains to the legal secretary what options are available. In this instance, the legal secretary types GO, and, when prompted by the computer to supply a COMMAND, types the command DIVORCE to place the divorce system into operation. The computer asks if a client data file is present, and the secretary responds N for *no*. After giving the secretary a few additional instructions on such things as how to halt the client interview, the computer begins asking questions:



```
  %$CALL, LAWYER

DELIVERY SYSTEM PASSWORD
? [redacted]


            IIT/Chicago-Kent
          Legal Services Clinic
        Automated Delivery System

  To have a document prepared, type "GO".
When prompted for a command, type:
     WILL  (to have a will assembled)
     TRUST (to have a trust assembled)
     CLOSING  (real-estate clsg. agmt.)
     DIVORCE  (divorce complaint or order)
     GUARDN  (guardianship papers)
     INCORP (create a corporation)
     ENGLISH (Lesson on writing)
     HELP  (to obtain a command list)
  Please make your selection.  To turn
off the computer, type "%LOGOUT"
? GO
 COMMAND? DIVORCE
CLIENT DATA FILE PRESENT/TO BE USED ("Y"=YES, "N"=NO)? N




               DOCUMENT PROCESSOR

ANSWER THE QUESTIONS OR TYPE THE FOLLOWING COMMANDS:

!                - I DO NOT WISH TO ANSWER.
!FINISH          - DO NOT ASK ANY MORE QUESTIONS.
!ABORT           - ABORT THIS RUN.  DESTROY THE DOCUMENT.
!HELP            - SHOW ME THE COMPLETE COMMAND LIST.


     Is a petition for dissolution of marriage to be prepared?


? yes


      What is the petitioner's first and middle name?


? Mary S.


      What is the petitioner's last name?


? Goodman


      What is the respondent's first name and middle name?


? John T.
```

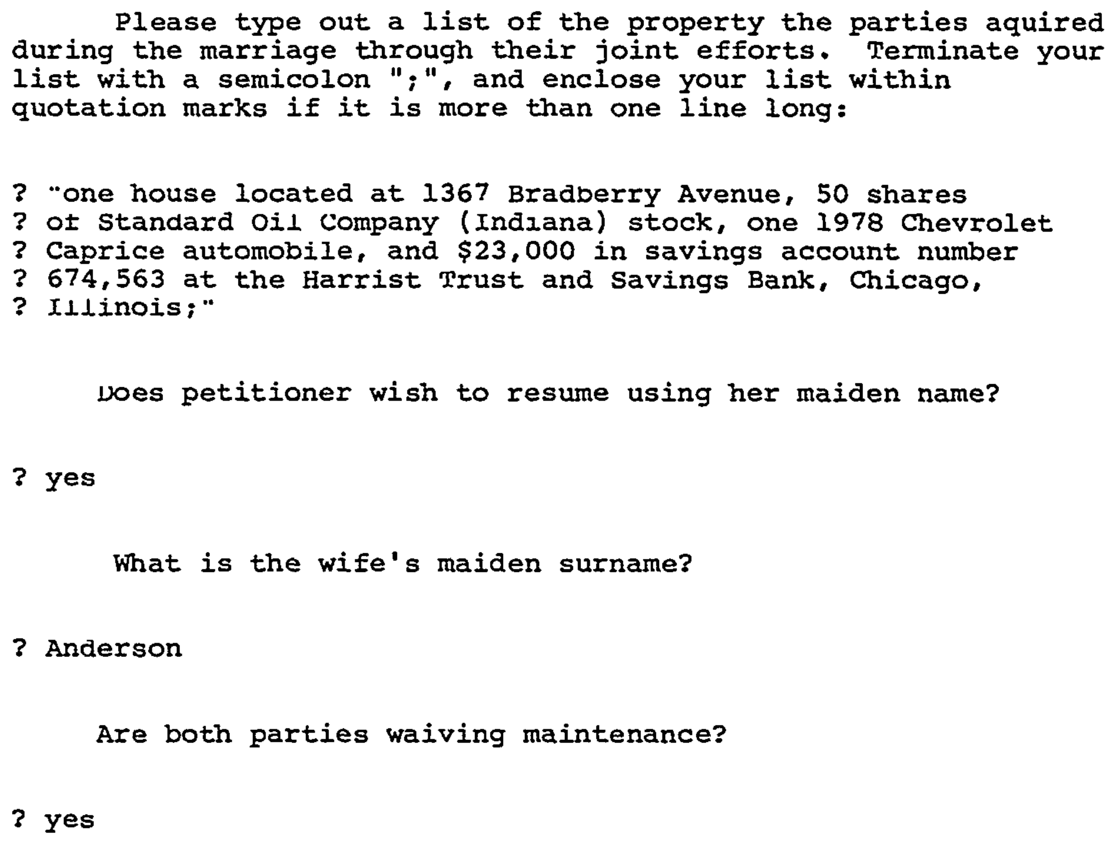
```
      What is the respondent's last name?

? Goodman

      Is the case to be handled pro se?

? no

      Is the petitioner female?

? yes

      Is a senior law student working on this case?

? no

      What is the name of the supervising attorney?

? Ronald W. Staudt

      What is the petitioner's age in years?

? 27

      What is the petitioner's street address?

? 1367 Bradberry Avenue

      What is the city in which the petitioner lives?

? Chicago

      What is the year petitioner began residing in Illinois?

? 1970

      Please type the letter that corresponds to the statement
that most accurately reflects petitioner's employment status:

    A.  Petitioner is employed;
    B.  Petitioner is self-employed;
    C.  Petitioner is retired;
    D.  Petitioner is unemployed.

? A
```

```
      What is the name of the petitioner's employer?

? Illinois Bell Telephone Company

      What is the petitioner's occupation?

? telephone installer

      What is the respondent's age in years?

? 31

     Do you know respondent's address?

? yes

      What is the respondent's street address?

? 6623 Cermack Road

      What is the city in which the respondent lives?

? Chicago

      What is the state in which the respondent lives or, if the
respondent lives outside the U. S., what is the country in which
he lives?

? Illinois

     Is the year respondent began residing in Illinois known?

? yes

      What is year respondent began residing in Illinois?

? 1970
```

```
     Please type the letter that corresponds to the statement
that most accurately reflects respondent's employment status:

    A.  Respondent is employed;
    B.  Respondent is self-employed;
    C.  Respondent is retired;
    D.  Respondent is unemployed;
    E.  Respondent's employment status is unknown.


? A


      What is the name of the respondent's employer?


? Illinois Bell Telephone Company


      What is the respondent's occupation?


? telephone lineman


      What is the date of the marriage? (Example: "November 12,
1974")


? June 23, 1976


     Were the parties married in the U. S.?


? yes


      What is the county where the marriage took place?
       (e.g.: "Cook", "Lake", etc.)


? Cook


      What is the state where the marriage took place?


? Illinois
```

```
      Please type the letter that corresponds to the statement
that most accurately reflects petitioner's grounds for
dissolution:

    A.   Desertion
    B.   Constructive desertion
    C.   Mental cruelty
    D.   Physical cruelty
    E.   Conviction of a felony
    F.   Adultery
    G.   Attempt on life of spouse
    H.   Habitual drunkenness
    I.   Drug addiction
    J.   Previous living spouse (bigamy)
    K.   Impotency
    L.   Infecting spouse with venereal disease
    M.   Don't know


? C


      What is the number of children born to the parties?


? 0


      What is the number of children adopted by the parties?


? 3


      What is the first adopted child's full name?


? John S. Goodman


      Please type in John S. Goodman's age, in years or months:
          (for example, "7 years" or "11 months")


? 3 years


     Do the children all live at the same address?


? yes


      What is the address, including city and state, where all
the children live?


? 1367 Bradberry Avenue
```

```
      What is the next adopted child's full name?

? Sara W. Goodman

      Please type in Sara W. Goodman's age, in years or months:


? 2 years

      What is the next adopted child's full name?

? Paul H. Goodman

      Please type in Paul H. Goodman's age, in years or months:


? 4 months

      What is the number of minor children of the parties?

? 3

      What is the name of the person to whom petitioner wants
child custody awarded?

? Mary S. Goodman

     Are the parties expecting a baby?

? no


      Please type the letters that correspond to the statements
which describe the way property was acquired during this
marriage. For example, you may answer "B", or "A B C", or " " if
none are applicable.  If you type more than one letter, please
separate them by spaces or blanks:

     A.  The parties, through their joint efforts, acquired
            property during the marriage.

     B.  The petitioner, through her own efforts,
           or by gift, has acquired property.

     C.  The petitioner has acquired property since the time
            the parties separated.

? A
```



```
     Please type out a list of the property the parties aquired
during the marriage through their joint efforts.  Terminate your
list with a semicolon ";", and enclose your list within
quotation marks if it is more than one line long:


? "one house located at 1367 Bradberry Avenue, 50 shares
? of Standard Oil Company (Indiana) stock, one 1978 Chevrolet
? Caprice automobile, and $23,000 in savings account number
? 674,563 at the Harrist Trust and Savings Bank, Chicago,
? Illinois;"


     Does petitioner wish to resume using her maiden name?


? yes


     What is the wife's maiden surname?


? Anderson


     Are both parties waiving maintenance?


? yes
```

C. The Client File

The computer automatically assembles all the answers supplied during an interview into a client file that may be saved and used to cause the automated assembly of other documents at a later time, or it may be revised and used to produce a set of revised documents without the need for a second client interview. A portion of the client file that was produced during the production of the above petition is reproduced below:

```
petn: a petition for dissolution of marriage IS to be prepared ;

LET pfmn: the petitioner's first and middle name =
"Mary S." ;

LET pln: the petitioner's last name =
"Goodman" ;

LET petname: the name of the petitioner =
"Mary S. Goodman" ;

LET rfmn: the respondent's first name and middle name =
"John T." ;

LET rln: the respondent's last name =
"Goodman" ;

LET resname: the name of the respondent =
"John T. Goodman" ;

pro se: the case IS NOT to be handled pro se ;

pet is fem: the petitioner IS female ;

LET his or her =
"her" ;

student: a senior law student IS NOT working on this case ;

LET atty: the name of the supervising attorney =
"Ronald W. Staudt" ;

LET petage: the petitioner's age in years = 27 ;

LET he or she =
"she" ;

LET pstadr: the petitioner's street address =
"1367 Bradberry Avenue" ;

LET pcty: the city in which the petitioner lives =
"Chicago" ;

LET pet adr: the petitioner's address, including city =
"1367 Bradberry Avenue, Chicago" ;

LET pet yrl: the year petitioner began residing in Illinois =
1970 ;

LET pet work: the petitioner's employment sentence = DOCUMENT
]petitioner is employed by [ pet boss: the name of the
petitioner's employer ] as a [ pet job: the petitioner's
occupation ];[ END ;


LET pet boss: the name of the petitioner's employer =
"Illinois Bell Telephone Company" ;

LET pet job: the petitioner's occupation =
"telephone installer" ;

LET resage: the respondent's age in years = 31 ;
```

D. The Form Divorce Petition Document

Reproduced below is the form document that produced the interview set forth above and that controlled the automated assembly of the petition also set forth above. Some portions of the final petition, such as the standard heading containing the name of the court and the names of the parties, are actually separate form documents that are incorporated by reference into this document, and accordingly such portions do not appear below. For clarity, we have deleted from this form petition the postprocessor commands that control such things as paragraphing, title centering, underlining, page break points, and single and double spacing.

The numbering of the paragraphs is postponed until just before the petition is printed. In the place of paragraph numbers the special commands #R0 and #R1 appear below. When the document is printed, the sequence of numbers *1*, *2*, *3*, and so on replaces the #R0s. Another sequence of numbers *1*, *2*, *3*, and so on replaces the #R1s. Postponing this numbering makes it simpler to delete or insert a paragraph at the last moment.

```
 [ lgdbhdr: the heading for legal size, double spaced documents ]

   [ divhdg: the standard heading for divorce documents ]

          PETITION FOR DISSOLUTION OF MARRIAGE

      Petitioner, [ petname: the name of the petitioner ], [
IF pro se: the case IS to be handled pro se
   INSERT ] PRO SE, [
OTHERWISE
   INSERT ]  by [ his or her ] [
   IF student: a senior law student IS working on this case
      INSERT ]  attorneys, [ atty: the name of the supervising
attorney ] and [ stdnt nm: the name of the senior law student ],
certified senior law student, [
   OTHERWISE
      INSERT ]  attorney, [ atty: the name of the supervising
attorney ], [
   ENDIF ]  IIT/CHICAGO KENT COLLEGE OF LAW LEGAL SERVICES
CENTER, [
ENDIF ]  alleges:

      #R0.  Petitioner is [ petage: the petitioner's age in
years ]  years of age; [ he or she ] resides at [ pet adr: the
petitioner's address, including city ], Illinois, and has
resided in Illinois since [ pet yrl: the year petitioner began
residing in Illinois ]; [ pet work: the petitioner's employment
sentence ]

      #R0.  Respondent is [ resage: the respondent's age in
years ]  years of age;  [
IF r add kn: you DO know respondent's address
   INSERT she or he ] resides at [ resaddr: respondent's
address, including city and state or country, if outside the
United States ]; [
OTHERWISE
   INSERT ]  respondent's address is unknown; [
ENDIF
IF r st kn: you DO know in which state or country resopndent
lives AND r ill res: respondent DOES reside in Illinois
   IF resyrbun: the year respondent began residing in Illinois
IS NOT known
      INSERT ] respondent resides in Illinois; [
   OTHERWISE
      INSERT she or he ] has resided in Illinois since [ resyr
l: year respondent began residing in Illinois ]; [
   ENDIF
ENDIF ]
[ res work: the respondent's employment sentence ]

      #R0.  The parties were married on [ wedate: the date of
the marriage ]  and their marriage is registered in [ wedcount:
the county and state in which the parties were married, if U.S.,
otherwise, the city and country ];

      #R0.  At the commencement of this action, petitioner is
domiciled in the State of Illinois;

      #R0.  Respondent has been guilty of [ grounds: the
grounds for divorce ];

  [ child par: the child paragraph for the petition and decree ]
  [
IF no mi ch: the number of minor children of the parties IS
GREATER THAN 0
   INSERT ]

      #R0.  [ guardnm: the name of the person to whom
petitioner wants child custody awarded ] is a fit and proper
person to have the care, custody, control, and education of the
minor [ child or children ]. [
ENDIF ]

  [ preg par: the pregnancy paragraph for the petition and
decree ]

  [ ppy cls: the petitioner's property clauses ]

  [ mdn nm par: the maiden name paragraph for the petition and
decree ]

      WHEREFORE petitioner [ petname: the name of the
petitioner ], prays as follows:

      #R1.  That a judgment of dissolution of marriage be
awarded the parties dissolving the marriage;

  [
IF no mi ch: the number of minor children of the parties IS
GREATER THAN 0
   INSERT ]

      #R1.  That [ guardnm: the name of the person to whom
petitioner wants child custody awarded ]  be awarded the care,
custody, control, and education of the minor [ child or children
].  [
ENDIF ]  [

IF pp: the petitioner HAS acquired property through petitioner's
own efforts or through gift OR asp: the petitioner HAS acquired
property since the separation
   INSERT ]

      #R1.  That petitioner be awarded as [ his or her ] sole
and exclusive property, free of all interest of respondent, the
nonmarital property described above; [
ENDIF ]  [

IF jp: the parties DO have marital property
   INSERT ]

      #R1.  That the marital property be awarded in just
proportions; [
ENDIF ]  [
IF both w mnt: both parties ARE waiving maintenance
   INSERT ]

      #Rl.  That petitioner waives [ his or her ]  right to
maintenance and asks that respondent be barred from seeking to
collect maintenance from [ him or her ]; [
OTHERWISE

   IF mnt for p: petitioner DOES want maintenance
      INSERT ]

      #Rl.  That petitioner be awarded a reasonable sum as
temporary and permanent maintenance; [
   OTHERWISE
      IF res mnt: petitioner DOES want to reserve maintenance
         INSERT ]

      #Rl.  That petitioner reserves [ his or her ] right to
maintenance; [
      ENDIF
   ENDIF
ENDIF ]

  [
IF pet is fem: the petitioner IS female AND mdnnm: petitioner
DOES wish to resume using her maiden name
   INSERT ]

      #Rl.  That petitioner be permitted to resume the use of
her maiden name of [ maidname: petitioner's maiden name ]; [
ENDIF ]

      #Rl.  For such other relief as this Court deems just.


                              _________________________________
                                   Attorney for petitioner


                         VERIFICATION

      [ petname: the name of the petitioner ], being first
duly sworn on oath deposes and states that [ he or she ]  has
read the contents of the foregoing petition and knows the same
to be true in substance and in fact.


                              _________________________________
                               [ petname: the name of the petitioner ],
                                   petitioner


 SUBSCRIBED AND SWORN to
 before me this _____ day
 of _________ , 198___.

 _____________________
 NOTARY PUBLIC

  [ trailer: the atty name and add blk ]
```

E. An Incorporated-by-Reference Document in the Child Paragraph Passage

The form document reproduced below is incorporated by reference into the form petition document reproduced above. The numbering of the paragraphs is postponed, and the special commands #R0, #R2, and #R3 are inserted into locations where paragraph numbers will later be placed. Many other similar form document passages are incorporated by reference into both the petition and the decree.

```
 [
IF no chldrn: the number of children born to the parties IS
GREATER THAN 1
   INSERT ]

      #R0.  The following children were born to the parties as
a result of this marriage:  [

   USING child counter
   REPEAT ]

          #R2.  [ #child:  the child's name ], [ #age: the
child's age, in years or months ] old, who resides at [
#address: the child's address, including city and state ];
  [
      UNTIL child counter EQUALS no chldrn: the number of
childrerr born to the parties
   ENDRPT ]  [

OTHERWISE

   IF no chldrn: the number of children born to the parties
   EQUALS 1
      INSERT ]

      #R0.  The following child was born to the parties to this
marriage: [ child:  the child's name ]  [ age: the child's age,
in years or months ] old, who resides at [ address: the child's
address, including city and state ]; [

   OTHERWISE
      INSERT ]

      #R0.  No children were born to the parties as a result
of this marriage; [
   ENDIF
ENDIF ]  [

IF no ad ch: the number of children adopted by the parties IS
GREATER THAN 1
   INSERT ]

      #R0.  The following children were adopted by the parties
to this marriage:  [

   USING child counter
   REPEAT ]
```

```
          #R3.  [ #a ch:  the adopted child's name ], [ #a ch
age: the adopted child's age, in years or months ] old, who
resides at [ #a ch add: the adopted child's address, including
city and state ];
  [
      UNTIL child counter EQUALS no ad ch: the number of
children adopted by the parties
   ENDRPT ]  [

OTHERWISE

   IF no ad ch: the number of children adopted by the parties
   EQUALS 1
      INSERT ]

      #R0.  The following child was adopted by the parties as a
result of this marriage: [ a ch:  the adopted child's name ]  [
a ch age: the adopted child's age, in years or months ] old, who
resides at [ a ch add: the adopted child's address, including
city and state ]; [

   OTHERWISE
      INSERT ]

      #R0.  No children were adopted by the parties to this
marriage; [
   ENDIF
ENDIF ]
```

F. A New Question Document

The following new question document is one of 14 such documents in the divorce system. Without this new question document, the system would ask, "What is the adopted child's name?" over and over again. This new question document causes the system to ask, "What is the adopted child's full name?" if there is only one such child. If there are two or more adopted children, the system asks, "What is the first adopted child's full name?" to obtain the first child's name, and then it asks, "What is the next adopted child's full name?" to obtain the names of the remaining children. Simply adding this new question document to the system thus changes the way this question is asked. Of course, the reformulated question could be much more elaborate than this simple example:

```
?     What is the [
UNLESS  no ad ch: the number of adopted children = 1
   IF child counter EQUALS 1 INSERT]first [
   OTHERWISE INSERT]next [
   ENDIF
ENDIF]adopted child's full name?
```

G. Computational Procedures

Reproduced below are 2 of the 74 procedures that form an important part of the divorce system.

The first procedure sets the address of each adopted child equal to the address where all the children live if there are two or more adopted children and if they live together. This procedure thereby blocks the computer from asking for the address of each child separately when they all live at the same address:

```
/
IF  no ad ch: the number of children adopted by the parties
IS GREATER THAN 1 AND sameadd: the children all DO live at the
same address
   THEN
   LET #a ch add: the address of this adopted child = allchadr:
the address, including city and state, where all of the children
live
```

The second procedure causes a list of four alternative responses to be presented to the legal secretary. Each of these responses is assigned a letter A through D. The legal secretary types one of these letters, and then the computer selects an appropriate petitioner's employment sentence to be incorporated by reference into the petition in accordance with the letter selected. If the response is not a letter falling within the range A to D, the computer displays an error message, and the entire procedure is repeated. The legal secretary is permitted to EXIT from this procedure only after he or she has supplied an appropriate response to the question.

```
/
REPEAT

   ASK ]     Please type the letter that corresponds to the
statement that most accurately reflects petitioner's employment
status:

    A.  Petitioner is employed;
    B.  Petitioner is self-employed;
    C.  Petitioner is retired;
    D.  Petitioner is unemployed.  [

   END

   IF reply EQUALS "A"
      LET pet work: the petitioner's employment sentence = pet
empl: the employment sentence for an employed petitioner ;
      EXIT

   OTHERWISE
      IF reply EQUALS "B"

         LET pet work: the petitioner's employment sentence =
pet s emp: the employment sentence for a self employed
petitioner ;
         EXIT

      OTHERWISE
         IF reply EQUALS "C"

            LET pet work: the petitioner's employment sentence =
pet ret: the employment sentence for a retired petitioner ;
            EXIT

         OTHERWISE
            IF reply EQUALS "D"

               LET pet work: the petitioner's employment
sentence = "Petitioner is unemployed;"
               EXIT

            OTHERWISE

               DISPLAY ]     You typed [ reply ] which is an
invalid answer. You must type "A", "B", "C", or "D".   Please
try again. [
               END

            ENDIF
         ENDIF
      ENDIF
   ENDIF

ENDRPT
```