

ARTICLES

THINKING LIKE A LAWYER: EXPERT SYSTEMS FOR LEGAL ANALYSIS

BY RICHARD GRUNER †

The statute of limitations was about to run and attorney John Smith was faced with a highly complex problem in a very difficult field of law. Fortunately, Smith had access to LEX, a legal expert system containing knowledge in this field of law based on the accumulated expertise of top attorneys from the past three generations. In addition, LEX was supported by a research staff that kept it constantly abreast of the latest developments within its area of expertise.

Based on the information Smith was able to obtain during the early phases of his consultation with LEX, Smith began to identify the primary causes of action available to his client, including a possible cause of action under an obscure statute that Smith had initially overlooked. Further, LEX knew its limitations—it was able to identify potential causes of action outside its primary area of expertise, suggesting that further consultations with appropriate attorneys or expert systems might be desirable.

Once Smith determined the claims he would pursue, LEX generated a draft complaint by combining information that it had already obtained about Smith's client with standard "boiler-plate" provisions for such claims. Additionally, LEX gave a detailed description of the normal procedures for bringing such claims, and generated a working draft of initial interrogatories that Smith might want to use in discovery. LEX also proposed a scheme for organizing information obtained in discovery.

After completing this consultation, Smith felt much more at ease about how he would handle his client's problem. Although he would have to do some research to confirm LEX's analysis, he now had a conceptual starting point in the case, a draft complaint, and a tentative discovery plan. Best of all, Smith was able to obtain a preliminary analysis of his client's problem with minimal effort and maximum efficiency. LEX did the work more quickly

© 1987 High Technology Law Journal

† Associate Dean and Associate Professor of Law, Whittier College School of Law; LL.M. 1982, Columbia University School of Law; J.D. 1978, University of Southern California Law Center; B.S. 1975, California Institute of Technology. The author would like to thank David A. Battaglia and Eulalio J. Garcia for their assistance.

and at less cost than would have been incurred if the work had been done manually by attorneys on Smith's legal staff.¹

Expert computer systems represent the future of legal practice. The attorney who best understands the benefits and limitations of such systems, while remaining undaunted by nagging fears of computers, will have a significant advantage in completing a wide variety of professional tasks.

An expert system is a type of computer program designed to emulate the analytic skills of human experts² — hence the name "expert

1. While the scenario described here is pure fantasy, it is not too far removed from present reality. For example, the JUDITH expert system assists attorneys in developing lines of legal reasoning by systematically reviewing the elements of various types of civil claims. See *infra* text accompanying notes 166-176. CORPTAX and Tax Advisor are systems that help attorneys recognize civil tax liability arising under narrow, but complex, provisions of the federal tax laws. See *infra* text accompanying notes 138-165. Another system estimates the settlement value of a given products liability case, taking into account not only the strength of the legal claims made, but also surrounding factors like the characteristics of the litigants and the identity of the judge in the case. See *infra* text accompanying notes 210-224. The ABF Processor and A-9 are two existing systems that assist in the assembly of legal documents based on user input. See *infra* text accompanying notes 226-243. LRS is a recently developed expert system which assists in the organization and retrieval of documents in a specialized legal area. See *infra* text accompanying notes 253-266.

LEX's primary advance over existing legal expert systems is that it combines more advanced forms of today's computer capabilities in a single well-integrated package. In the near future, the technology will develop to the point where a system can perform all of the functions mentioned in this "Computer Fantasy." Indeed, Donald Waterman, a scientist with the Rand Corporation and a leading analyst of expert systems, projects that new expert systems for legal document preparation, interpretation and prediction, scheduling and monitoring, and case management have great potential for success and will lead to revolutionary changes in the way legal work is performed. See D. WATERMAN, A GUIDE TO EXPERT SYSTEMS 224-26 (1986). Other leading computer researchers have also projected important roles for expert legal systems as both lawyers' tools and as research devices for studying how lawyers work and think. See R. SCHANK, THE COGNITIVE COMPUTER 220-21 (1984); Buchanan & Headrick, *Some Speculation About Artificial Intelligence and Legal Reasoning*, 23 STAN. L. REV. 40, 62 (1970).

2. See, e.g., Duda & Shortliffe, *Expert Systems Research*, SCIENCE, Apr. 15, 1983, at 261; Hayes-Roth, *The Knowledge-Based Expert System*, IEEE COMPUTER, Sept. 1984, at 11.

Expert systems differ from conventional computer programs in the way they are organized, the way they incorporate knowledge, the way they execute, and the impression they give system users. See Hayes-Roth, *The Knowledge-Based Expert System*, IEEE COMPUTER, Sept. 1984, at 11. The most significant of these differences is the separation of general knowledge about a class of user problems (stored in a system's "knowledge base") from information about a current problem (contained in input data) and from methods for applying the general knowledge to the problem at hand (incorporated in a reasoning module sometimes called an "inference engine"). These separations not only permit system designers to achieve maximum performance in each of these areas, but also allow a system to be easily changed by adding a new knowledge or reasoning base. See Gevarter, *Expert Systems, Limited but Powerful*, IEEE SPECTRUM, Aug. 1983, at 39.

Expert systems and the technology underlying them are described in detail in P. HARMON & D. KING, EXPERT SYSTEMS (1985); F. HAYES-ROTH, D. WATERMAN & D. LENAT, BUILDING EXPERT SYSTEMS (1983); C. KULIKOWSKI & S. WEISS, A PRACTICAL GUIDE TO

systems."³ Such systems are designed to undertake the same work as human experts, at least in narrow ranges of analysis.⁴ Examples of existing expert legal systems include a system that uses client information to advise attorneys on the merits of alternative forms of bankruptcy filings⁵ and a program that explores tax planning options based on information about a client's property holdings and tax goals.⁶

Less ambitious computer systems, sometimes called "knowledge systems," attempt to emulate only a small fraction of an expert's analytic powers, such as the ability to answer one complex question or to interpret a single type of specialized information.⁷ Examples of current

DESIGNING EXPERT SYSTEMS (1984); D. WATERMAN, A GUIDE TO EXPERT SYSTEMS (1986); Duda & Shortliffe, *Expert Systems Research*, SCIENCE, Apr. 15, 1983, at 261; Michaelsen, Michie & Boulanger, *The Technology of Expert Systems*, BYTE, Apr. 1985, at 303; Forsyth, *The Anatomy of Expert Systems*, in ARTIFICIAL INTELLIGENCE: PRINCIPLES AND APPLICATIONS 186 (M. Yazdani ed. 1986); Gevarter, *Expert Systems, Limited but Powerful*, IEEE SPECTRUM, Aug. 1983, at 39; Hayes-Roth, *The Knowledge-Based Expert System*, IEEE COMPUTER, Sept. 1984, at 11; Kinnucan, *Computers That Think Like Experts*, HIGH TECHNOLOGY, Jan. 1984, at 30.

3. See Duda & Shortliffe, *supra* note 2, at 261; cf. P. HARMON & D. KING, *supra* note 2, at 5 (the label "expert systems" reflects the use of knowledge drawn from human experts as a basis for system programming).

4. The degree of intelligence exhibited by expert systems is a matter of some debate. While admitting that computers have qualities that make them superior to human analysts in certain contexts, Professors Hubert and Stuart Dreyfus of the University of California argue that many intelligent tasks undertaken by humans involve complex "intuitive" insights that computers cannot match. They claim that true expert behavior depends heavily on unconscious processes whereby experts recognize that new situations are similar to remembered ones for purposes of analysis. See Dreyfus & Dreyfus, *Why Expert Systems Do Not Exhibit Expertise*, IEEE EXPERT, Summer, 1986, 86, 88.

While a valid warning about the limitations of early expert systems, the Dreyfus' thesis may be overstated. The types of intuitive analyses the Dreyfus' consider emblematic of expert behavior may soon be accomplished by computers through increased information processing capabilities available in the form of computers called "parallel processors." These computers will be able to simultaneously test for multiple similarities between new situations and circumstances stored in databases. With these new pattern recognition capabilities, the parallel processors should permit computer systems to approximate the "intuitive" linkages experts use to relate new problems to old solutions. Cf. D. MICHIE & R. JOHNSON, *THE KNOWLEDGE MACHINE* 29-34 (1985) (discussing the potential importance of parallel processors in artificial intelligence applications involving pattern recognition). Even with these technical advances, the implementation of the computer equivalent of intuition will depend on careful descriptions of what experiences experts typically bring to their work and what types of paradigm problem solutions those experiences have taught the experts to apply. See Dreyfus & Dreyfus, *supra* note 4, at 86-88.

5. See Boyd, *Choosing Between a Chapter 7 and a Chapter 13 Bankruptcy: An "Expert System" to Assist an Attorney in Making the Choice*, in COMPUTING POWER AND LEGAL REASONING 699 (C. Walter ed. 1985).

6. See Michaelsen, *An Expert System For Federal Tax Planning*, 1 EXPERT SYSTEMS, Oct. 1984, at 149.

7. For purposes of this Article, I will use the term "expert system" to refer to both traditional expert systems and knowledge systems. Knowledge-based systems apply a large body of human knowledge to solve particular problems, but also sometimes per-

knowledge systems include a program that analyzes corporate reorganizations to determine if they qualify for special income tax treatment under federal law⁸ and a system designed to examine the tax consequences of operating foreign corporations as either a subsidiary or a division of a parent firm.⁹

Expert systems offer the promise of new accuracy and completeness in legal analyses.¹⁰ Expertise captured in an expert system will be more permanent and comprehensive than its human counterpart; not only will it survive its original author, but it may also reflect the aggregate wisdom and experience of numerous experts in a particular specialty. The expertise contained in an expert legal system will also be easily transferable and reproducible, often through means as simple as copying a computer program or database. Further, where analysis is heavily dependent on numerical calculations or requires boring repetitions of a simple reasoning process, the tireless mechanical operation of expert systems may produce significantly more reliable results than human experts in a shorter amount of time. Relying on expert systems to perform these tedious tasks also frees human workers to perform more interesting and useful duties. Finally, the legal analysis performed by expert systems will be easy to evaluate, since computers can easily document their analytic steps and are not limited by human impatience with paperwork.

Although expert legal systems have many advantages, they also possess several drawbacks.¹¹ In general, current expert systems have a

form applications for which there are no qualified experts or which are only a small part of a human expert's tasks. See Duda & Shortliffe, *supra* note 2, at 267 n.6; P. HARMON & D. KING, *supra* note 2, at 5. The discussions in this Article, although referring only to "expert systems," apply to "knowledge-based" computer systems as well.

Many analysts refer to large knowledge-based systems emulating the performance of human experts as "expert systems" and smaller, less capable systems as "knowledge systems." E.g., *id.*; cf. Duda & Shortliffe, *supra* note 2, 267 n.6. In much of the technical literature, however, this distinction is not observed and the terms "expert systems," "knowledge-based systems," and "knowledge systems" are used interchangeably. E.g., P. HARMON & D. KING, *supra* note 2, at 5; Hayes-Roth, *supra* note 2, at 11; Kinnucan, *supra* note 2, at 30.

8. See McCarty, *Intelligent Legal Information Systems*, 9 RUTGERS COMPUTER & TECH. L.J. 265, 267-76 (1983); McCarty, *Reflections on TAXMAN: An Experiment in Artificial Intelligence and Legal Reasoning*, 90 HARV. L. REV. 837 (1977).

9. See Hellawell, *CHOOSE: A Computer Program for Legal Planning and Analysis*, 19 COLUM. J. TRANSNAT'L L. 339 (1981).

10. "[K]nowledge systems do not display biased judgments, nor do they jump to conclusions and then seek to maintain those conclusions in the face of disconfirming evidence. They do not have 'bad days'; they always attend to details, and they always systematically consider all of the possible alternatives. The best of them, equipped with thousands of heuristic rules, are able to perform their specialized tasks better than a human specialist." P. HARMON & D. KING, *supra* note 2, at 7.

11. For an evaluation of the weaknesses of expert systems when compared with human experts, see D. WATERMAN, *supra* note 1, at 13-15.

limited ability to translate complex factual information into ideas and concepts, meaning that information input into such systems must often be limited in scope. This consequently limits the scope of the program's analysis. Expert systems also tend to be overinclusive in that they blindly undertake useless or wasteful analyses which human experts are able to avoid through the use of common sense knowledge about everyday life. Expert systems also run the risk of being underinclusive because they are typically designed to take only specified factors into account in their analyses and to ignore some tangential factors that may affect those analyses. In addition, such systems are often unable to "learn" new information through their own efforts and are therefore heavily dependent upon human assistance to keep abreast of changing legal expertise. Finally, current expert legal systems lack the means to develop novel approaches to new problems through the use of analogies and other creative reasoning processes.

Primitive expert systems and knowledge systems have already been developed to assist with diverse legal tasks, ranging from law office automation¹² to estimation of the settlement value of products liability claims¹³ to the formulation of criminal sentences.¹⁴ Overall, however, the development of expert systems for legal analyses has lagged behind the extensive development of such systems in other professions,¹⁵ such as medicine.¹⁶ This Article evaluates expert systems for legal analysis and how they will affect legal practice. It is written in hopes of sparking new interest in expert legal system development. Consequently it

12. See Sprowl & Staudt, *Computerizing Client Services in the Law School Teaching Clinic: An Experiment in Law Office Automation*, 1981 AM. B. FOUND. RES. J. 699; Sprowl, *Automating the Legal Reasoning Process: A Computer That Uses Regulations and Statutes to Draft Legal Documents*, 1979 AM. B. FOUND. RES. J. 1.

13. See Waterman & Peterson, *Models of Legal Decisionmaking*, RAND CORP. INST. FOR CIV. JUST. REP. 2717-ICJ (1981) [hereinafter *Models of Legal Decisionmaking*]; Waterman & Peterson, *An Expert Systems Approach to Evaluating Product Liability Cases*, in COMPUTING POWER AND LEGAL REASONING 627 (C. Walter ed. 1985) [hereinafter *Evaluating Product Liability Cases*]; Waterman & Peterson, *Evaluating Civil Claims: An Expert Systems Approach*, 1 EXPERT SYSTEMS 65 (1984) [hereinafter *Evaluating Civil Claims*]; Waterman, Paul & Peterson, *Expert Systems for Legal Decision Making*, 3 EXPERT SYSTEMS 212, 217-18 (1986) [hereinafter *Legal Decision Systems*].

14. See DeMulder & Gubby, *Legal Decision Making by Computer*, 4 COMPUTER L.J. 243 (1983).

15. Several observers have been surprised by the relatively small amount of recent research done in regard to expert legal systems given their vast potential for success. See D. WATERMAN, *supra* note 1, at 224; Hellawell, *A Computer Program for Legal Planning and Analysis: Taxation of Stock Redemptions*, 80 COLUM. L. REV. 1362, 1365 n.5 (1980). For an analysis of the problems facing designers of expert legal systems, see *Legal Decision Systems*, *supra* note 13, at 213-15.

16. For a broad review of expert systems in medicine and the development efforts behind them, see READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE (W. Clancey & E. Shortliffe ed. 1984).

describes features which future expert legal systems should possess to meet the special needs of attorneys. The extent to which attorneys should rely on such systems is analyzed as well.

The Article first briefly summarizes the history of expert systems and describes in lay terms how a typical expert system operates. The Article then analyzes the needs of attorneys and formulates the characteristics of an "ideal" expert legal system with these needs in mind. Finally, existing expert legal systems are analyzed in light of the characteristics of this ideal expert system.

I. A BRIEF HISTORY OF EXPERT SYSTEMS

Efforts to program computers to act more like humans are almost as old as computers themselves. Since World War II, computer decision-making systems, robotics devices, and speech recognition techniques have been extensively studied by computer scientists.¹⁷

Collectively, these and other computerized systems designed to emulate intelligent human behavior are called "artificial intelligence" devices. The term "artificial intelligence" refers to the goal of enabling computers to behave in ways that humans recognize as "intelligent."¹⁸ Designing artificial intelligence programs typically involves two types of problems. The first is determining the nature of intelligent behavior in a particular context. The second is analyzing how to program a computer to approximate that behavior.¹⁹

A key breakthrough in artificial intelligence research occurred in the early 1950's with the creation of computers having sufficient information processing capacity to allow them to solve problems posed in symbolic rather than mathematical terms.²⁰ This capacity was significant because it permitted computers to emulate a broader range of human analyses in fields where humans work by manipulating symbols and expressions.²¹ Expert tasks based on symbol processing are important in many fields:

17. The early history of artificial intelligence research is recounted in P. JACKSON, INTRODUCTION TO EXPERT SYSTEMS 2-10 (1986) and P. MCCORDUCK, MACHINES WHO THINK (1979).

18. See E. FEIGENBAUM & P. MCCORDUCK, THE FIFTH GENERATION 312 (rev. ed. 1984); R. SCHANK, THE COGNITIVE COMPUTER 25-38 (1984); Lenat, *Computer Software for Intelligent Systems*, SCIENTIFIC AMERICAN, Sept. 1984, at 204, 204-207; cf. Buchanan & Headrick, *supra* note 1, at 40-41 (artificial intelligence research includes efforts to design computer information processing to perform legal reasoning).

19. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 34-38, 81-91.

20. Duda & Shortliffe, *supra* note 2, at 261; see generally COMPUTERS AND THOUGHT (E. Feigenbaum & J. Feldman eds. 1963).

21. See B. RAPHAEL, THE THINKING COMPUTER 4-5, 18-19 (1976); Newell & Simon, *Computer Science as Empirical Inquiry: Symbols and Search*, 19 COMM. ACM 113, 119 (Mar. 1976).

for example, editors work with words, a city planner sketches diagrams of roads or utility lines, and a lawyer works with both words and legal principles.²²

A number of specialized computer programming languages have been developed to perform symbolic manipulations.²³ The most frequently used is LISP (LISt Processor), a language which facilitates computer handling of information lists.²⁴ For example, using a list of facts about a given object, the presence of that object can be determined by testing for the presence of the listed attributes. PROLOG (PROgramming in LOGic) is another symbolic programming language often used to create artificial intelligence programs.²⁵ Unlike most programming languages, which require a programmer to tell the computer how to perform its tasks, PROLOG permits a programmer to define facts and rules about a given problem and then allows the computer to attempt to find solutions to the problem on its own. Since the heart of a PROLOG program is a series of declaratory statements about factual and inferential expertise governing a given type of problem, such a program is often much easier to understand than a program in a general-purpose programming language where expert knowledge may be hidden in numerous procedural specifications.

Initial expectations for the potential power of artificial intelligence programs were extremely high. Experts tried to develop a machine that would think and reason like a human being, thereby making it adaptable to any type of analysis in any particular field. Some initial successes led artificial intelligence researchers to expect that computers could extend the abilities of problem-solving machines to most, if not all, challenging mental activities.²⁶ However, further attempts to provide computers with general analytic powers—and thereby create intelligent machines with broad problem-solving capabilities—were dismal failures. Most of these programming efforts were premised on the expectation that a few formal rules underlie most intelligent activity and analysis, and that these rules, once identified, could form the basis for general problem-solving by a computer.²⁷ However, these broadly applicable analytic rules were

22. See B. RAPHAEL, *supra* note 21, at 18-19.

23. For analyses of several symbolic programming languages and their advantages in artificial intelligence projects, see E. RICH, *ARTIFICIAL INTELLIGENCE* 389-408 (1983).

24. See J. ALLEN, *ANATOMY OF LISP* (1978); E. RICH, *supra* note 23, at 393-95; P. WINSTON & B. HORN, *LISP* (1981). For an assessment of how LISP can be used to create expert systems, see P. HARMON & D. KING, *supra* note 2, at 85-87.

25. W. CLOCKSIN & C. MELLISH, *PROGRAMMING IN PROLOG* (1981); E. RICH, *supra* note 23, at 401-02 (1983). Applications of PROLOG in expert system development are discussed in P. HARMON & D. KING, *supra* note 2, at 87-89.

26. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 36-37; Duda & Shortliffe, *supra* note 2, at 261.

27. See, e.g., E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 36-37.

seldom found, rendering useless many early studies premised on the existence of such rules.²⁸ The reason for these failures lay in the inability of computer scientists to realize the paramount importance in problem-solving of "domain-specific" analysis rather than general analytic powers. Researchers often discovered that general methods of analysis, as opposed to methods tailored to a particular kind of problem, could not address the complex combinations of information necessary to complete useful analyses. General problem-solving techniques designed to apply to imprecisely stated problems, uncertain facts, and unreliable axioms produced results that were meaningless or erroneous.²⁹ Consequently, contemporary researchers began to place new significance on "domain-specific" analysis as a basis for problem solving. "Domain-specific" analysis concentrates the computer's powers on solving a specific problem given a limited range of facts, rather than on trying to design a general program to solve a diverse set of problems. New understanding of human problem-solving in narrow expert domains formed the basis for early efforts to develop practical expert systems.³⁰

In developing "domain-specific" systems, researchers carefully studied the manner in which human experts approached and solved various types of problems. These studies revealed that experts applied two fundamentally different types of knowledge in fashioning a solution to a given problem. They often relied on not only substantive principles from their field of expertise, but also on practical knowledge and "rules of

28. Roger Schank of Yale University observed this disappointment in another artificial intelligence field—research involving natural language processing.

Work on natural language processing began in the 1950's, but those initial attempts bear almost no resemblance to our approach today. These researchers, whether they happened to be in computer science or linguistics, started out by attempting to find the formal rules that might underlie language itself. . . . Early research concentrated on the outer *form* of language rather than on the *content* of communication, and focused on getting a computer to break a sentence down into its parts of speech rather than into its elements of meaning.

These early researchers first attempted to build a *parser*. They wanted the computer to *parse* or convert sentences into grammatical structures according to standard grade-school rules. . . . These investigators hadn't really given any thought to what would come after that. They never succeeded in writing down all the grammar rules, and found themselves with a lot of unanswered questions about how to get the computer to use language effectively.

R. SCHANK, *supra* note 18, at 135.

29. See Duda & Shortliffe, *supra* note 2, at 261; Stefik, Aikins, Balzer, Benoit, Birnbaum, Hayes-Roth & Sacerdoti, *Basic Concepts for Building Expert Systems*, in BUILDING EXPERT SYSTEMS 59, 66 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983) [hereinafter *Basic Concepts*].

30. See Duda & Shortliffe, *supra* note 2, at 261-62; cf. Buchanan & Headrick, *supra* note 1, at 45-46 (noting the importance of studies determining how lawyers break problems down into subproblems and identifying the decision rules attorneys apply in solving problems as prerequisites to the development of artificial intelligence devices capable of legal reasoning).

thumb.”³¹ For example, in solving problems in specialized areas like law, engineering, or medicine, experts must apply “know-how” about their given field. Such know-how is comprised of detailed substantive knowledge of their specialty and a certain amount of common-sense knowledge usually gained through experience. For lawyers, know-how consists of a combination of formal substantive training in law school and a history of accumulated experience obtained in actual practice. In other fields, such as medicine, clinical or apprenticeship methods of training are a standard part of a minimum education. These clinical or apprenticeship experiences are designed to help students obtain expertise in the practical or common-sense aspects of their field—expertise that is essential to successful work as an expert, but difficult if not impossible to teach in a classroom.

Application of expert knowledge or know-how is essential in programming successful expert systems because of the characteristics of the problems such systems are expected to solve. Most of the difficult and interesting problems addressed by experts do not have algorithmic solutions; they originate in complex social or physical contexts and therefore resist precise description and mathematical analysis. Consequently, expert systems designed to solve these types of problems must go beyond algorithmic solutions and apply detailed substantive and procedural know-how.³² Thus, artificial intelligence “researchers learned that high IQ doesn’t make a person expert; specialized know-how does. To make a fast and consistent symbol processor perform as well as a human expert, someone must provide it specialized know-how comparable to what a human expert possesses.”³³

The discovery of techniques for capturing expert knowledge and for representing it in forms that can be used by computer systems has consequently become a primary goal of expert systems research.³⁴ Such

31.

An expert is a person who, because of training and experience, is able to do things the rest of us cannot; experts are not only proficient but also smooth and efficient in the actions they take. Experts know a great many things and have tricks and caveats for applying what they know to problems and tasks

Johnson, *What Kind of Expert Should a System Be?*, 8 J. MED. & PHIL. 77 (1983). See also E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 67; P. HARMON & D. KING, *supra* note 2, at 32-33; Hayes-Roth, *supra* note 2, at 12.

32. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 36-37, 82-83; Duda & Shortliffe, *supra* note 2, at 219-20.

33. Hayes-Roth, *supra* note 2, at 12.

34. *Basic Concepts*, *supra* note 29, at 59; Stefik, Aikins, Balzer, Benoit, Birnbaum, Hayes-Roth & Sacerdoti, *The Architecture of Expert Systems*, in BUILDING EXPERT SYSTEMS 89 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983) [hereinafter *Architecture of Expert Systems*]; Buchanan, *Expert Systems: Working Systems and the Research Literature*, EXPERT SYSTEMS, Jan. 1986, at 32.

Expert systems development is a peculiarly commercial branch of artificial intelligence

"knowledge engineering"³⁵ has been successfully undertaken in a number of fields to produce expert systems that perform a variety of important tasks. The following are some examples of tasks performed by expert systems:

Chemistry

Identifying Chemical Compounds from Laboratory Data³⁶

Decoding and Synthesizing DNA Sequences³⁷

Medicine

Diagnosing Bacterial Infections³⁸

Recommending Cancer Treatments³⁹

research which contrasts with an earlier emphasis on theoretical studies. These earlier studies were initiated to test theories of mental processes and were close in spirit to prior philosophical, linguistic, and psychological inquiries into cognitive behavior. Most recent expert system research has instead been aimed at producing computer system products with problem solving abilities to meet the demands of particular industries. See R. SCHANK, *supra* note 18, at 32-34.

35. Stanford Professor Edward A. Feigenbaum defines "knowledge engineering" as follows:

The knowledge engineer practices the art of bringing the principles and tools of [artificial intelligence] research to bear on difficult applications problems requiring experts' knowledge for their solution. The technical issues of acquiring this knowledge, representing it, and using it appropriately to construct and explain lines-of-reasoning, are important problems in the design of knowledge-based systems.

Feigenbaum, *The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering*, 5 PROC. INT'L JOINT CONF. ON ARTIFICIAL INTELLIGENCE 1014, 1014 (1977). For more detailed descriptions of the work of knowledge engineers, see E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 86-91; Buchanan, Barstow, Bechtal, Bennett, Clancey, Kulikowski, Mitchell & Waterman, *Constructing an Expert System*, in BUILDING EXPERT SYSTEMS, 127, 127-67 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983). For an analysis of practical problems arising in knowledge engineering of an expert legal system, see Newquist, *The Making of a Tax Expert*, AI EXPERT, Mar. 1987, at 57.

36. DENDRAL identifies the molecular structure of unknown compounds based on mass spectral and nuclear magnetic resonance measurements. See R. LINDSAY, B. BUCHANAN, E. FEIGENBAUM & J. LEDERBERG, APPLICATIONS OF ARTIFICIAL INTELLIGENCE FOR ORGANIC CHEMISTRY: THE DENDRAL PROJECT (1980).

37. MOLGEN assists geneticists in designing genetics experiments by taking the user's gene-splicing goals and refining them into a set of specific laboratory procedures. See Stefik, *Planning With Constraints (MOLGEN: Part 1)*, 16 ARTIFICIAL INTELLIGENCE 111-139 (1981); Stefik, *Planning and Meta-Planning (MOLGEN: Part 2)*, 16 ARTIFICIAL INTELLIGENCE 141-69 (1981).

38. MYCIN assists physicians in treating blood and meningitis infections. It uses information about patient symptoms to first suggest the likely identity of an infection and then to recommend a corresponding program of drug therapy. See B. BUCHANAN & E. SHORTLIFFE, RULE-BASED EXPERT SYSTEMS: THE MYCIN EXPERIMENTS OF THE HEURISTIC PROGRAMMING PROJECT (1983).

39. ONCOCIN aids in the development of chemotherapy treatments for cancer patients. It recommends a particular patient's therapy by analyzing information about a patient's diagnosis, previous treatments, and laboratory tests and comparing this information to chemotherapy models based on past experience with therapeutic benefits and toxic side effects of similar treatments. See Shortliffe, Scott, Bischoff, Campbell, Van

	Diagnosing Internal Disorders ⁴⁰
<i>Industry</i>	Configuring Computer Systems ⁴¹
	Identifying Computer Malfunctions ⁴²
	Diagnosing Oil Drilling Problems ⁴³
	Analyzing Mineralogic Data ⁴⁴

II. AN INTRODUCTION TO EXPERT LEGAL SYSTEMS: THE BASIC COMPONENTS

There is no one way to program a computer to act like a human expert. However, because such systems are intended to undertake the same types of analyses as do human experts, the mental steps that human experts go through in solving problems or giving advice delineate

Melle & Jacobs, *ONCOCIN: An Expert System for Oncology Protocol Management*, in PROC. SEVENTH INT'L JOINT CONF. ON ARTIFICIAL INTELLIGENCE 876-81 (1981).

40. CADUCEUS and INTERNIST are the first and second versions respectively of an expert system that uses patient histories, symptoms, and laboratory results to suggest diagnoses of internal medicine disorders. See Miller, Pople & Myers, *INTERNIST-1: An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine*, in READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE 190 (W. Clancey & E. Shortliffe eds. 1984).

41. XCON, previously called R1, is an expert system developed by the Digital Equipment Corporation ("DEC") to configure computer systems for customer locations. The system uses customer orders and information about the characteristics of DEC computer products to identify additional components that must be added to the orders to create operational systems. It also lays out the spacial relationships for placement of the ordered equipment upon installation. See P. HARMON & D. KING, *supra* note 2, at 155-59 (1985); McDermott, *R1: A Rule-Based Configurer of Computer Systems*, 19 ARTIFICIAL INTELLIGENCE 38-88 (1982); McDermott, *R1: The Formative Years*, 2 AI MAGAZINE 21-29 (1981).

42. A proprietary expert system developed at Stanford University for the IBM Corporation uses information about a computer failure to identify which major component of the computer is malfunctioning. Once identified, the malfunctioning component (typically a circuit board) can be replaced by field repair personnel and brought to an IBM facility for a more complete diagnosis. See S. WEISS & C. KULIKOWSKI, *A PRACTICAL GUIDE TO DESIGNING EXPERT SYSTEMS* 63-67 (1984); E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 75; Kinnucan, *supra* note 2, at 39.

43. DRILLING ADVISOR helps drilling supervisors diagnose and remedy the most likely causes of oil drill sticking in boreholes. It uses information about the geologic formations at a drill site and observed drilling symptoms to identify likely causes of sticking and to recommend actions to prevent reoccurrences. Hayes-Roth, *supra* note 2, at 21-23.

44. PROSPECTOR analyzes geologic data about a region and estimates the probability of finding specific types of minerals there. See Gaschnig, *PROSPECTOR: An Expert System for Mineral Exploration*, 10 MACHINE INTELLIGENCE 301-23 (J. Hayes, D. Michie & Y.H. Pao eds. 1982); Duda, Gaschnig & Hart, *Model Design in the PROSPECTOR Consultant System for Mineral Exploration*, in EXPERT SYSTEMS IN THE MICRO-ELECTRONIC AGE 153-67 (D. Michie ed. 1979). PROSPECTOR is reportedly responsible for a molybdenum find in the State of Washington worth as much as \$100 million. E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 78-79.

the basic capabilities a complete expert system should possess. Consideration of the mental processes of human experts also aids expert system designers in understanding what makes expert analyses difficult, thereby focusing design attention on critical analytic steps.⁴⁵

A. Components of Problem-Solving by Human Experts

Although techniques used by human experts to complete analyses and solve complex problems vary widely depending on the type of analyses or problems involved, all such techniques involve some combination of perception, memory, information processing, and response.⁴⁶ These elements of human analyses serve as a basis for subsequent discussions of their counterparts in expert legal systems.

Every expert begins his analysis by first discovering the facts which have caused a problem to arise. Information can be retrieved from a variety of sources such as the inspection of a scene or article, special measurements of the physical features of an object, or verbal descriptions by a client. Once information about a problem is obtained, it is stored in the analyst's memory. Two major types of memory activities are generally involved in problem-solving—short-term memory and long-term memory.⁴⁷ Typically, little of the detailed information an expert collects and manipulates will stay in the analyst's memory after the problem is solved. Thus, memory activity in the immediate course of the investigation and resolution of a problem is sometimes called short-term memory to distinguish it from an expert's long-term retention of basic expertise.

An example from legal practice should make the distinction between short-term and long-term memory clear. If an attorney were to examine a client's affairs for tax planning purposes, factual information about the client's business and personal circumstances would be assembled in the attorney's short-term memory as she examined different planning options. The attorney would also draw knowledge of relevant tax law principles from her long-term memory in order to analyze the client's activities. Once recalled, these principles would be kept in the attorney's short-term memory and manipulated in combination with the client's factual information to produce an expert analysis.⁴⁸

45. See *Basic Concepts*, *supra* note 29, at 82.

46. For a brief overview of the human problem solving processes underlying expert behavior (and, hence, defining the basic objectives of expert system development), see P. HARMON & D. KING, *supra* note 2, at 22-33. See generally P. LINDSAY & D. NORMAN, *HUMAN INFORMATION PROCESSING: AN INTRODUCTION TO PSYCHOLOGY* (1972); A. NEWELL & H. SIMON, *HUMAN PROBLEM SOLVING* (1972).

47. P. HARMON & D. KING, *supra* note 2, at 22-25 (1985).

48. Cf. Buchanan & Headrick, *supra* note 1, at 52-53 (observing that legal reasoning typically involves a combination of direct applications of memorized legal rules and rea-

Information processing by human experts involves the manipulation of factual information to make rational inferences which can then be used in further analyses.⁴⁹ This inference process typically involves repetition of the following pattern of information manipulation in an expert's short-term memory: (1) information regarding facts and relevant principles is gathered and grouped in short-term memory; (2) when enough information is present, an inference or conclusion will be made based upon these facts and principles.

An expert's response to a client or other party after completing an analysis involves the communication of the results of that analysis in a form that is useful to the recipient. Rather than simply describing the primary conclusions reached in the analysis, the expert's response might include an explanation of how the conclusion was reached, as well as what factors would have produced a different outcome. In addition, where the outcome of an expert analysis forms the basis for related work, such as the creation of legal documents necessary to carry out an attorney's recommendation, the expert's response to her client may include this related work-product.

B. Components of Expert Systems

An example drawn from legal practice will illustrate how cognitive tasks undertaken by an attorney might be replicated in the components of an expert legal system. In assessing the legal implications of a client's activities, a lawyer typically does the following:

- (1) consults the client and gathers facts about the client's problem;
- (2) recalls substantive knowledge in the appropriate legal field and performs legal research to identify relevant legal standards and precedents;
- (3) makes inferences, based on these precedents and standards, which predict probable legal implications of the client's problem;
- (4) explains this outcome to the client and recommends appropriate client action.

An expert system capable of useful legal interpretations of client behavior would need to accomplish all of these tasks. Typically, the computer programming steps necessary to complete each of these separate tasks are grouped into distinct segments or "modules." A modular approach allows a system developer to easily change one portion of the system without the need for altering other modules.⁵⁰ Under this

soning by analogy from memorized rules or cases).

49. See P. HARMON & D. KING, *supra* note 2, at 25-26.

50. See Duda & Shortliffe, *supra* note 2, at 266; Gevarter, *supra* note 2, at 39.

approach, an expert legal system designed to accomplish the legal tasks described above would include four modules.⁵¹ The input module would gather information about the problem from the attorney-user and translate the attorney's responses into a form that the expert system could utilize. The knowledge-base module would store expert knowledge, such as judicial precedents or other expert know-how relevant to a client's problem. The inference module utilizes this knowledge to formulate legal conclusions. Finally, the output module communicates to the attorney the system's conclusions and explanations of how they were reached.

These system components are examined individually in the remainder of this Section to give the reader a better sense of how a typical expert legal system might operate. This framework is also used in the following Section of this Article to analyze the goals of expert legal system design.⁵²

1. *Input Modules*

Attorneys usually begin their analyses by gathering a detailed set of facts about a problem to be solved. They initially ask extensive questions of the client concerning the problem and may interview or depose many individuals in order to discover more facts. Later, as preliminary steps in the analysis are completed, attorneys will come back to the client with further questions dictated by the particular direction the analysis has taken.

Unfortunately, difficulties in understanding human languages have prevented computers from gathering information from spoken or written

51. While the four expert system modules described here are certainly not the only units that an expert legal system might be divided into, they are typical of the types of programming modules found in most expert systems. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 81; cf. Forsyth, *The Anatomy of Expert Systems*, in *ARTIFICIAL INTELLIGENCE: PRINCIPLES AND APPLICATIONS* 186 (M. Yazdani ed. 1986) (four essential components of a fully-fledged expert system are a knowledge base, an inference engine, a knowledge-acquisition module, and an explanatory interface); Hayes-Roth, Waterman & Lenat, *An Overview of Expert Systems*, in *BUILDING EXPERT SYSTEMS* 3, 16-19 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983) [hereinafter *Overview of Expert Systems*] (ideal expert system would contain seven modules: knowledge base, "blackboard" for holding intermediate system results, interpreter, analysis scheduler, consistency enforcer, result justifier, and language processor).

52. Not every system will contain all these system modules. The wisdom of dividing a system into these or other modules depends greatly on factors like the programming techniques used to create the system and the types of legal analyses the system will perform. However, whether a system is divided into modules or not, it must accomplish the types of tasks outlined above. Thus, with some allowance for differences associated with greater or lesser system integration, the discussion of system characteristics and design goals in this Article is relevant to the operation of all expert legal systems.

words in the manner a human would.⁵³ Some of the most sophisticated expert systems now include language interpreters or "natural language" subsystems that translate a restricted set of human statements into information understandable by other portions of the expert system.⁵⁴ However, because of the inherent ambiguity of language, the importance of context in establishing the meaning of many words and phrases, and the vast range of ideas and concepts that a complete natural language subsystem would need to recognize, no expert system capable of understanding unconstrained human communication now exists. Indeed, it is doubtful that a computer capable of understanding the content of a significant number of randomly selected human language statements will be available soon.

These limitations do not create insurmountable barriers to information gathering by expert systems. The key to making computers understand human statements is to limit the range of statements which the computers must recognize.⁵⁵ This can be done in several ways. One method involves using a structured series of questions (with acceptable responses limited to a few easily recognized choices such as "Yes" or "No") or a numbered list of possible answers. An alternative method involves programming a computer to directly understand a restricted vocabulary. The system user is then required to translate his communications into this truncated language. Both of these communication approaches are already used in expert legal systems.⁵⁶

2. Knowledge Bases

a. Types of Information

Knowledge bases in expert systems serve a function similar to long-term memory in human experts.⁵⁷ They both must retain two

53. The problems surrounding natural language understanding by computer systems are described in 1 THE HANDBOOK OF ARTIFICIAL INTELLIGENCE 223-361 (A. Barr & E. Feigenbaum eds. 1981) [hereinafter HANDBOOK]; B. RAPHAEL, *supra* note 21, at 177-210 (1976); R. SCHANK, *supra* note 18, at 90-109; Winograd, *Computer Software for Working With Language*, SCIENTIFIC AMERICAN, Sept. 1984, at 130.

54. P. HARMON & D. KING, *supra* note 2, at 137-39.

55. *Id.*

56. Compare Hellowell, *supra* note 15, at 1363 (CORPTAX system in which information is gathered through Yes/No answers and number input) with McCarty, *Reflections on TAXMAN*, *supra* note 8, at 837 (TAXMAN system based on input of user information through a restricted vocabulary).

57. Long-term memory refers to that portion of an expert's memory which is used to retain information that has been learned through years of experience or education and that is subject to recall for use in expert analyses. It contrasts with an expert's short-term memory, which only temporarily retains facts and analyses about a particular problem. See P. HARMON & D. KING, *supra* note 2, at 22-25. Legal knowledge held in a lawyer's long-term memory comes in many forms, from basic legal standards like the

different types of information used in expert analyses.⁵⁸ The first is the substantive knowledge commonly shared by human experts in a particular legal field.⁵⁹ For an expert legal system, this type of knowledge would be roughly equivalent to the contents of a leading treatise in a specialized legal field. The second major category of expert information is heuristic knowledge or "heuristics"⁶⁰ — that is know-how or rules of thumb usually gained by an expert through years of practice which expedite the expert's search for solutions to problems.⁶¹

The following example illustrates the distinction between these two types of knowledge. An attorney has a client who purchased goods

provisions of statutes and regulations, to patterns of case results suggesting common law legal rules and practical algorithms for legal analysis and problem solving. Knowledge bases in expert legal systems should incorporate all of these types of information.

58. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 82; P. HARMON & D. KING, *supra* note 2, at 5; D. WATERMAN, *supra* note 1, at 16-17.

59. One group of commentators has called this type of information "public knowledge" since it "includes the published definitions, facts, and theories of which textbooks and references in the domain of study are typically composed." *Overview of Expert Systems*, *supra* note 51, at 4. Another describes the substantive knowledge in expert systems as "the widely shared knowledge, commonly agreed among practitioners, that is written in textbooks and journals of the field, or that forms the basis of a professor's lectures in a classroom." E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 82.

60. "Heuristic" derives from the same Greek root as "eureka" and refers to a branch of philosophy dealing with methods and rules of discovery and invention. HANDBOOK, *supra* note 53, at 29; E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 86. Artificial intelligence specialists have used "heuristic" as a term of art to refer to several different concepts. HANDBOOK, *supra* note 53, at 28-30. Several authors have used "heuristic" to refer to a device or rule that improves the efficiency of an artificial intelligence system's analyses, but which does not guarantee that the analyses will lead to a correct answer. As one group of commentators described this usage:

A heuristic (heuristic rule, heuristic method) is a rule of thumb, strategy, trick, simplification, or any other kind of device which drastically limits search for solutions Heuristics do not guarantee optimal solutions; in fact, they do not guarantee any solution at all; all that can be said for a useful heuristic is that it offers solutions which are good enough most of the time.

E. FEIGENBAUM & J. FELDMAN, *supra* note 20, at 6. Unfortunately, some authors have used the term "heuristic" to refer to processing rules that increase a system's analytic efficiency or that do not carry a guarantee of success, but that do not necessarily have both these characteristics. See Newell, Shaw & Simon, *Empirical Explorations With the Logic Theory Machine: A Case History in Heuristics* in COMPUTERS AND THOUGHT 6 (E. Feigenbaum & J. Feldman eds. 1963); Minsky, *Steps Toward Artificial Intelligence*, in COMPUTERS AND THOUGHT 407 (E. Feigenbaum & J. Feldman eds. 1963).

61. This is equivalent to saying that a rule of thumb in form contract cases directing early attorney attention to possible warranty disclaimers is not guaranteed to produce a definitive answer regarding breach of warranty relief available to a buyer. If time were of no concern, a more thorough assessment of the strength of a client's warranty claims could be reached by systematically reviewing possible sources of warranty liability and corresponding seller defenses. However, in real practice settings, lawyers constantly structure their analyses to address first what experience has taught them is the issue most likely to produce a definitive answer for their client.

under a form contract and later received goods which the client felt were below the general quality of like goods in the industry. In order for the attorney to advise the client about possible contractual remedies, the attorney will probably employ two different types of knowledge. First, the attorney will draw on substantive knowledge about the Uniform Commercial Code and, particularly, the implied warranty of merchantability.⁶² Second, the attorney might draw on his personal experience with form contracts and related disputes and might decide that he should first address issues such as the possibility of a valid warranty disclaimer in the contract.⁶³ While focusing on disclaimer issues may turn out to be a waste of time because there may not be a valid disclaimer in the contract, this ordering of analytic tasks will ensure an efficient use of the time of both the attorney and his clients over the long run. In calling upon his experience with prior cases rather than upon any recorded factual information regarding the likelihood of a disclaimer, the attorney would be using heuristic knowledge to best solve the client's problems.

b. Knowledge Representation in Knowledge Bases

Whether knowledge is drawn from traditional substantive sources or is heuristic, it must be recorded in the knowledge base of an expert system in a usable format. The most useful knowledge representation format in a particular system depends on the type of information involved and the way it is to be used by the system.⁶⁴ Information utilized in expert legal analyses can often be expressed in terms of relationships reflecting taxonomic, causal, definitional, or empirical associations between one set of facts and another.⁶⁵ For example, a typical statute

62. See U.C.C. § 2-314 (1977).

63. See U.C.C. §§ 2-316(2), 2-316(3)(a) (1977).

64.

[A] knowledge representation formalism should (i) represent the concepts and intentions of [an] expert faithfully, (ii) be able to be interpreted by the program correctly and effectively, (iii) support explanations that convey a line of reasoning that the human observer can understand and critique, (iv) facilitate the process of finding gaps and errors in the knowledge base, and (v) allow separation of domain knowledge from the interpretation program so that the knowledge base can be enlarged or corrected without the need for reprogramming the interpreter. These criteria place conflicting demands on the system designer. The first two (fidelity and effectiveness) lead toward complex representations specific to each situation, whereas the other three favor a single, uniform formalism that is simple to interpret.

Duda & Shortliffe, *supra* note 2, at 266; see also E. RICH, *supra* note 23, at 201-02; Buchanan & Headrick, *supra* note 1, at 45-47.

Debate about the best means for representing expert knowledge in knowledge bases assumes that experts' problem solving methods can be articulated and captured in decision rules. See *id.* at 45-46. Some analysts have argued, however, that this is a fundamental error since truly expert behavior depends on intuitive processes that cannot be defined in decision rules. See Dreyfus & Dreyfus, *supra* note 4, at 86-88.

65. Descriptions of such relationships are key parts of knowledge bases in all types of expert systems. See *Overview of Expert Systems*, *supra* note 51, at 12. A variety of sym-

The simplest way to represent relationship knowledge in an expert system is through statements or "production rules" reflecting links between a state of facts and the implications of these facts to an expert.⁶⁶ These production rules can often be expressed in terms of IF-THEN statements, as follows:⁶⁷

IF victim's use of product was foreseeable and reasonable, AND victim was injured by product, AND product was defective, AND defendant manufactured product, THEN consider a strict liability claim.

Using the above "IF-THEN" rule, a system would decide when to conduct a detailed strict liability analysis. Such production rules can easily form the basis for a "recognize-act" expert system.⁶⁸

A variety of more sophisticated knowledge representation schemes can also be incorporated into expert legal systems. Several existing systems are based on predicate calculus,⁶⁹ a formal logical system in which knowledge is represented in terms of "predicates"—i.e., statements about the characteristics of objects.⁷⁰ For example, the federal tax rule that ownership of N shares of Corporation C by person W will be attributed to W's spouse H can be expressed in the following predicate calculus statement:⁷¹

num-con-own(H,N,C):-spouse(H,W), owns(W,N,C).

where:

"num-con-own(H,N,C)" means "N shares of C Corporation stock constructively owned by H"

66. See HANDBOOK, *supra* note 53, at 190-99; P. HARMON & D. KING, *supra* note 2, at 25-26, 42-44.

67. This production rule is similar to those used in an existing computer system that models attorneys' settlement behaviors. See *Models of Legal Decisionmaking*, *supra* note 13, at 14-21.

68. See P. HARMON & D. KING, *supra* note 2, at 25.

69. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 861-62 (1977); Schlobohm, *Tax Advisor: A Prolog Program Analyzing Income Tax Issues*, DR. DOBB'S J., Mar. 1985, at 64.

70. For descriptions of predicate calculus and how it is used in designing knowledge bases, see HANDBOOK, *supra* note 53, at 160-71; P. HARMON & D. KING, *supra* note 2, at 46-48.

71. This representation is part of Tax Advisor, an expert system that interprets constructive stock ownership under § 318 of the Internal Revenue Code. Tax Advisor is written in PROLOG, a computer programming language based on predicate calculus principles. See Schlobohm, *TA—A Prolog Program Which Analyzes Income Tax Issues Under Section 318(a) of the Internal Revenue Code*, in *COMPUTER POWER AND LEGAL REASONING* 765 (C. Walter ed. 1985); Schlobohm, *supra* note 69, at 64.

“:-” means “if”

“spouse(H,W)” means “W is H’s spouse”

“,” means “and,” and

“owns(W,N,C)” means “W owns N shares of C Corporation stock.”

Additional knowledge representation schemes used in expert systems link objects and information about those objects in more complex ways.⁷²

3. Inference Systems

Inference strategies must also be developed to determine how an expert system accesses and applies its available knowledge.⁷³ These strategies or “metaknowledge”⁷⁴ define how an expert system will retrieve

72. See HANDBOOK, *supra* note 53, at 160-222; P. HARMON & D. KING, *supra* note 2, at 34-48; E. RICH, *supra* note 23, at 201-242.

73. Inference strategies control such basic expert system tasks as the selection of a line of reasoning that leads to the solution of a problem or the formulation of a body of consultative advice. Contrary to popular belief, inference strategies may be structured in a relatively simple manner. For example, the syllogism “IF X implies Y and IF Y implies Z, THEN X implies Z” is an example of such a strategy. See E. FEIGENBAUM & P. MCCORDUCK, *supra* note 18, at 125; P. HARMON & D. KING, *supra* note 2, at 49-50; Stefik, Aikins, Balzer, Benoit, Birnbaum, Hayes-Roth & Sacerdoti, *Basic Concepts for Building Expert Systems* in BUILDING EXPERT SYSTEMS 59, 64-66 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1986).

For even the simplest expert systems based solely on “IF-THEN” production rules, an inference control schema must address the following problems:

(1) *Selection*—choosing what inference rules should be applied next in a given analysis. The strategy underlying how these selections are made may be trivial (e.g., all potentially applicable rules can be applied at every step in a system’s analyses) or very sophisticated (e.g., criteria can be developed to eliminate from consideration rules that are unlikely to lead to a problem solution);

(2) *Matching*—comparing selected inference rules against the facts present in the problem at hand;

(3) *Scheduling*—where the antecedents of more than one potentially applicable inference rule are factually supported, deciding which satisfied rule should be utilized for subsequent steps of the analysis;

(4) *Execution*—modifying data to reflect the conclusions of factually relevant inference rules.

See Michaelsen, Michie & Boulanger, *The Technology of Expert Systems*, BYTE, Apr. 1985, at 303, 307.

74. “Metaknowledge” means “knowledge about knowledge.” Lenat, Davis, Doyle, Genesereth, Goldstein & Schrobe, *Reasoning About Reasoning*, in BUILDING EXPERT SYSTEMS 219, 220 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983). In the context of expert systems, metaknowledge includes *strategic information* about how system inference rules should be applied; *descriptive information* about the inference rules themselves (e.g., how fast the rule can be applied, what the rule derives from, etc.); *explanatory information* about which rules were applied to reach a system result (for use in explaining that result or in confirming system accuracy); and *systemic information* concerning the structure and retention of knowledge in the system (for facilitating updates and revisions). See *id.* at 219-38.

information from its knowledge base, how it will combine that retrieved information with data on a particular problem, and how it will draw inferences from this assembled information. Because the information retrieval and assembly process "drives" the expert system in completing its analyses, this portion of an expert system is sometimes referred to as an "inference engine."⁷⁵ In comparison with human processes, the operation of an inference engine corresponds to an analyst's accumulation of information needed for problem-solving in short-term memory and manipulation of this information to reach rational conclusions.

The most efficient inference strategy in an expert system will depend largely on the characteristics of the expert knowledge the system must apply.⁷⁶ Determinative characteristics include not just the magnitude of knowledge involved, but also the level of uncertainty associated with the knowledge and whether the knowledge can be organized in abstract terms.⁷⁷ A brief description of several computer inference strategies and the types of knowledge that can be utilized efficiently with them illustrates the significance of these factors.⁷⁸

a. Inference Rules for Small Knowledge Domains

The simplest expert systems are based on a very narrow range of knowledge involving no uncertainty.⁷⁹ A system can apply such narrow knowledge through a search strategy that is exhaustive in the sense that all of the knowledge relationships contained in the system are considered in every one of its analyses.⁸⁰ The inference relationships found

75. *E.g.* P. HARMON & D. KING, *supra* note 2, at 49; E. RICH, *supra* note 23, at 41; Kinnucan, *supra* note 2, at 32.

76. Some problem solving approaches involve efficient ways of searching among all possible problem solutions. Other problem solving methods focus on finding ways to narrow the search for solutions to smaller, manageable subsets of all possible solutions. Gevarter, *supra* note 2, at 41.

77. *See Basic Concepts*, *supra* note 29, at 66-72.

78. For more comprehensive descriptions of the problem solving strategies used in expert systems, see HANDBOOK, *supra* note 53, at 21-139; B. RAPHAEL, *supra* note 21, at 55-176; E. RICH, *supra* note 23, at 25-112; *Architecture of Expert Systems*, *supra* note 34, at 89-126.

79. Hayes-Roth, *supra* note 2, at 17; *Architecture of Expert Systems*, *supra* note 34, at 90-93.

80. Systems that comprehensively search for problem solutions can apply inference rules in either a forward "data-driven" manner or a backward "goal-driven" sequence. *See, e.g.*, HANDBOOK, *supra* note 53, at 198; P. HARMON & D. KING, *supra* note 2, at 55-57; E. RICH, *supra* note 23, at 56-60.

In a forward reasoning system, premises of inference rules are examined to see whether or not a user's facts match. Where a match is found, the conclusions derived from the rule are added to the facts "known" to the system. The system then reassesses the premises of its inference rules to determine if any additional ones are matched. By repeatedly following this pattern, the system pieces together a line of reasoning from the initial facts presented to a set of expert conclusions. Professor James Sprowl's ABF Pro-

to be factually supported can then be used to define the system's expert conclusions.⁸¹

If a knowledge domain contains uncertain information or if the facts which a user submits to the system are uncertain, a more complex search strategy is necessary. Some expert systems handle this problem by applying numeric certainty factors to facts and rules, with such factors reflecting the possibility that a certain fact or conclusion is correct.⁸² For example, where a clear split of authority is present and there is only a sixty percent chance that a contract with certain characteristics will be enforceable, a rule could be built into an inference process to discount the certainty of the expert system's conclusion to sixty percent. If a further analysis suggested that, if enforceable, there is only a fifty percent chance of recovery from the party asserted to have breached the contract, these two discounting percentages could be combined to yield an overall probability of thirty percent for recovery under the contract.⁸³ In this fashion, expert systems can produce a derived certainty factor for their final conclusions. Another way expert systems deal with unreliable data is to initially treat the data as reliable, reach an expert conclusion,

cessor, which collects facts about a client and then interprets this information to generate appropriately tailored form documents, is an example of a forward reasoning system. See Sprowl, *supra* note 12, at 1; Sprowl & Staudt, *supra* note 12, at 699.

By contrast, a backward reasoning system completes its analyses by examining potential problem solutions and determining which of those solutions are factually appropriate. A medical diagnosis system, for example, might reason backward from all the diseases it knows, considering first the disease within its knowledge that is most likely to occur. It would then review the symptoms of the patient under study to determine if they are consistent with that disease. If an inconsistency were found, the system would consider the next most common disease. Through backward reasoning, the system will either identify a disease that matches the patient's symptoms or determine that none of the diseases about which the system has knowledge are applicable. MYCIN, a system for assisting physicians in diagnosing and treating infections, is a leading example of a backward reasoning expert system. See Davis, Buchanan & Shortliffe, *Production Rules as a Representation for a Knowledge-Based Consultation Program*, in READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE 98 (W. Clancey & E. Shortliffe eds. 1984); P. HARMON & D. KING, *supra* note 2, at 15-21, 61-75 (1985). Professor Robert Michaelsen's Taxadvisor system (derived in part from MYCIN) is a good example of a backward reasoning expert legal system. See Michaelsen, *supra* note 6, at 149, 154.

81. A simple example of an expert legal system based on a small knowledge domain and a comprehensive inference strategy is Professor Hellowell's CORPTAX program in which an unvarying analytic process is used to systematically review and apply the stock ownership attribution rules under Internal Revenue Code § 318. See Hellowell, *supra* note 15, at 1363.

82. P. HARMON & D. KING, *supra* note 2, at 50-52; Hayes-Roth, *supra* note 2, at 14; *Architecture of Expert Systems*, *supra* note 34, at 93-96.

83. A similar numerical "discounting" process is currently used in the Legal Decision-making System for valuing product liability case settlements. See Waterman & Peterson, *An Expert Systems Approach To Evaluating Product Liability Cases*, *supra* note 13, at 632-39.

and then ask the system user for further information which will confirm whether the system's tentative conclusion is correct.⁸⁴

Expert systems require different analytical techniques in order to take changes in circumstances over time into account. This is usually accomplished by representing knowledge in terms of processes which transform a situation with particular characteristics at time 1 into another situation at time 2.⁸⁵ For example, an expert legal system designed to determine if a corporate reorganization meets federal requirements for tax-free treatment might contain a knowledge base composed of pairs of "before" and "after" factual states, with each pair corresponding to a version of a tax-free reorganization. By determining whether the characteristics of a reorganization matched one of these pairs, the system could render an opinion regarding the tax-free status of an already completed reorganization. It could also give recommendations on how to achieve tax-free status when a reorganization is in the planning stages.⁸⁶

b. Inference Rules for Large Knowledge Domains

As the domain of knowledge a system applies grows larger, more sophisticated search strategies are necessary to produce expert systems that are fast enough to be useful.⁸⁷ Various advanced search strategies are necessary to avoid useless analyses.⁸⁸ Since the minimum set of

84. For example, a system could tentatively assume that a liability rule applicable only in some jurisdictions would apply to a case under scrutiny. It could then determine if the requirements for liability under that rule were met. The system would then ask the user if the liability rule would be followed in the case at hand.

85. Professor L. Thorne McCarty's TAXMAN system is an example of an expert system that tests for transitions in factual states having important legal implications, such as changes in stock ownership that will cause a corporate reorganization to qualify for special tax treatment. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 837 (1977); McCarty, *Intelligent Legal Information Systems*, *supra* note 8, at 267-276.

86. Professor McCarty's TAXMAN system incorporates models of before and after factual states corresponding to corporate reorganizations receiving favorable tax treatment. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 864-76. The system is capable of determining whether a particular reorganization qualifies for such treatment. Although a planning mode was not a part of the original implementation of TAXMAN, its developer recognized that this would be a feasible and advantageous extension of the system. See *id.* at 885-86.

87. Slow speed can significantly detract from the usefulness of an otherwise desirable expert system. This problem arises even where the computer being used is large and the programming system employed is particularly tailored for the design of expert systems. For example, the TAXADVISOR expert system for estate and gift tax planning, although developed for large mainframe computers using the EMYCIN expert system shell, operated at a very slow pace until it was reprogrammed to eliminate analyses unlikely to lead to pertinent tax planning advice. See Michaelsen, *supra* note 6, at 149.

88. There are two types of approaches to minimizing the analyses a system must undertake to reach a problem solution. The first involves finding ways to search among possible solutions more efficiently; for example, by considering the most likely solutions first. The second involves finding ways to narrow a system's analyses to small, manage-

analyses necessary to solve a particular type of problem depends greatly on the nature of the problem itself, there is no optimal minimizing strategy that is best in all cases. However, a repertoire of useful techniques has been developed for use in solving particular types of problems.

Some of these techniques include: (1) generating the most likely answers that a system might reach, testing these against the facts at hand, and rejecting answers which do not fit the facts until an acceptable answer is found;⁸⁹ (2) partition problem analyses — the division of the problem into subproblems depending on the characteristics of the problem itself;⁹⁰ (3) guessing between analytic alternatives that seem equally or similarly desirable, coupled with later analysis of whether the guess has led to an acceptable answer;⁹¹ (4) parallel reasoning in which

able sets of possible solutions. See Gevarter, *supra* note 2, at 41.

89. This form of expert system problem solving is sometimes referred to as a "generate and test" strategy. See *Architecture of Expert Systems*, *supra* note 34, at 99-102; E. RICH, *supra* note 23, at 73-75. The operation of DENDRAL is perhaps the most successful expert system to use the generate and test method. DENDRAL's ultimate objective is to identify unknown compounds from physical measurements such as mass spectrographic data. It accomplishes this by applying heuristic "rules of thumb" developed by human analysts to interpret the spectrographic data and project the most likely molecular structures involved. DENDRAL then generates those molecular structures and simulates the spectrographic data each would produce. The set of simulated data most closely matching the actual measurements is identified and the corresponding structure is chosen to identify the unknown compound. See Buchanan & Feigenbaum, *DENDRAL and Meta-DENDRAL: Their Applications Dimension*, 11 *ARTIFICIAL INTELLIGENCE* 5 (1978); P. HARMON & D. KING, *supra* note 2, at 134-137.

90. This approach to problem partitioning is useful where fixed partitioning would be inefficient because variations in problem circumstances would cause numerous subproblems to be addressed unnecessarily. See *Architecture of Expert Systems*, *supra* note 34, at 104-06. The partitioning using this technique occurs through a "top-down" process. The system first tests for the presence of the most abstract characteristics of potential problem solutions and then selects subsequent analytic tasks based on the solution characteristics initially found. For example, a system might require that the attorney specify the types of relief his or her client is seeking. This input can then be used by the system to structure the particular areas of the law the system will analyze. The JUDITH system for assisting lawyers in case analyses follows just such an approach. See Popp & Schlink, *JUDITH, A Computer Program to Advise Lawyers in Reasoning a Case*, 15 *JURIMETRICS J.* 303, 304 (1975).

Partitioning can also be founded on the use of an abstractions test. See E. RICH, *supra* note 23, at 272-275 (1983); *Architecture of Expert Systems*, *supra* note 34, at 106-10. Under this approach, the order of tasks is based on whether sufficient information exists in the system to complete those tasks. Put another way, under the least commitment principle, an analysis is deferred until the system has enough information to complete the analysis properly. In an expert legal system, for example, if the infliction of physical harm is a requirement for liability under an infliction-of-emotional-distress theory, the system could postpone analysis of whether the injuries involved in a particular case met this requirement until a full description of the injuries involved was obtained from the system user. See W. PROSSER & W. KEETON, *PROSSER AND KEETON ON THE LAW OF TORTS* 64 (5th ed. 1984).

91. *Architecture of Expert Systems*, *supra* note 34, at 110-15. It may be more efficient for the system to guess which is the proper alternative rather than attempt to obtain

multiple representations of a particular reasoning path are maintained at different levels of abstraction;⁹² (5) the use of multiple knowledge sources, each accessed by its own inference module;⁹³ and (6) restructuring the knowledge base used by a system to store knowledge in a way that it can be more efficiently accessed.⁹⁴

4. Output Module

Once its inference rules have been applied to the facts of a problem, an expert system must be capable of describing its expert advice or interpretations to the system user. The output of an expert legal system can take three basic forms: summary conclusions; explanations of how the system performs its activities; or advice as to the proper course of conduct to be taken by the user. The first form of output is simply a statement of the system's results. For example, a system determining whether a particular transaction meets statutory criteria for special tax treatment might produce a positive or negative output response.⁹⁵

Because understanding a system's reasoning is often as important as the system's ultimate conclusions,⁹⁶ an expert system often can provide

enough additional information to decide between alternatives. Once it makes its guess, the system can proceed with its analyses until it reaches a conclusion that is inconsistent with known facts about the problem under scrutiny. However, computer "guessing" will only be appropriate when the computer is programmed to recover from wrong guesses. See *id.* at 110-11.

92. *Architecture of Expert Systems*, *supra* note 34, at 115-16. Where little detail is needed to move the analysis forward, a summary definition of the analytic path is utilized to direct the system's reasoning, thereby avoiding consideration of unnecessary information. Yet, where detailed characteristics make a difference, a more complex problem representation is available in the form of a parallel representation.

93. *Architecture of Expert Systems*, *supra* note 34, at 116-19. In such an approach, knowledge bases and analytic processes are accessed as needed by a higher-level coordinating module. This coordinating or "scheduling" module selects the sequence in which other analyses are invoked based on preprogrammed rules dictating the types of analyses most likely to produce a successful result given the partial analytic results already obtained.

94. *Architecture of Expert Systems*, *supra* note 34, at 120-23. One simple data structuring technique is to store the most frequently accessed data in the most readily available locations. A more sophisticated technique is to locate data that would probably be accessed contemporaneously in adjacent storage locations. Such data storage considerations can be implemented directly in the structuring of knowledge bases, or can be handled by special programs which set up or "compile" knowledge bases from raw data submitted by system developers. Eventually, advanced system compilers may be able to analyze the characteristics of knowledge data, determine the most efficient storage format for that data, and then rearrange the data in the format deemed most efficient.

95. See, e.g., McCarty, *Reflections on TAXMAN*, *supra* note 8, at 879-81.

96.

By offering to explain how they reached their conclusions, [knowledge] systems convey to the user an impression of reasonableness. To construct an explanation, they transform the expert heuristic rules and assertions into lines of reasoning. A line of reasoning shows how a starting set of assumptions and a collection of heuristic rules produce a particular conclu-

a second form of output which explains its inference activities.⁹⁷ This output can consist of three different types of explanations. The first type is a description of the inference rules used by a system in reaching a result and the manner in which these inference rules were applied to the problem at hand. This type of explanation is similar to the response a human expert would give if a client asked how the expert reached a particular conclusion.⁹⁸ A second type of explanation given by some expert systems is a description of the significance of a question asked by the system. Such explanations address the manner in which the answer to the system's question will be used in the system's analyses, and thus correspond to the answer an expert might give when asked why he or she needs certain information.⁹⁹ A third form of explanation available in some systems is a definition of terms used in the system's inquiries or responses.¹⁰⁰

The final form of system output, available primarily in systems which are designed for planning purposes, is advice on the best future conduct of the system user in light of the system's expert conclusions. For example, in addition to providing an expert opinion about whether an individual's estate planning desires and property holdings are well matched, a system designed to assist in estate planning might recommend establishing trusts or taking other steps that would minimize the estate taxes which would otherwise be imposed upon the client's demise.¹⁰¹

III. MAN MEETS MACHINE: SATISFYING THE NEEDS OF ATTORNEYS

Useful expert legal systems must meet complex design requirements.¹⁰² An ideal expert legal system should at once be easy to use, inspire user confidence, utilize computer resources efficiently, and, above

sion. End-users generally find these explanations as plausible as the rules themselves.

Hayes-Roth, *supra* note 2, at 14.

97. See E. RICH, *supra* note 23, at 201-242; D. WATERMAN, *supra* note 1, at 90-91; P. HARMON & D. KING, *supra* note 2, at 62-63.

98. For an example of an expert legal system that includes the ability to explain how it reaches its conclusions, see Schlobohm, *supra* note 71, at 782-83; Schlobohm, *supra* note 69, at 70-72.

99. For an example of an expert legal system that is capable of explaining why it needs information, see Schlobohm, *supra* note 69, at 72.

100. Professor Hellawell's expert legal system dealing with stock redemptions which qualify for special tax treatment includes the capability to define and explain to users a number of the terms used in the system's questions. Hellawell, *supra* note 15, at 1364.

101. See Michaelsen, *supra* note 6, at 149.

102. For an early assessment of basic design criteria for computer systems capable of legal problem solving, see Buchanan & Headrick, *supra* note 1, at 45-47, 53-60.

all, perform useful tasks.¹⁰³ In order to define these design objectives more completely, this section will identify features of expert legal systems that attorneys would find desirable.

A. Functional Needs of Attorney Users

Attorney users will tend to utilize expert legal systems to make suggestions on how to solve a particular problem or to help organize information. Using systems in this fashion, attorneys will usually supplement the systems' analyses with their own expertise, conceptual insight, and research.¹⁰⁴ Beyond just possessing accurate and complete knowledge bases, an ideal system will also need to be capable of sophisticated analyses. The ability of attorneys to perform many simple legal analyses in a very short time means that the work of an expert legal system must rise above this "trivial" level to be of interest to attorney-users.

Furthermore, attorneys typically demand that they know why a question has been asked, the reasoning behind a given conclusion, and even what the terms of art in a given question mean as a matter of law. In order to inspire user confidence and assist an attorney's analysis, an ideal system would possess a comprehensive explanatory ability. An expert legal system used by an attorney will not need to explain basic concepts like "negligence" or "mens rea" since attorney users will already be familiar with such nomenclature and concepts. However, where a system is designed to assist an attorney in a specialized field, explanatory and definitional capabilities concerning that field must exist within the system.

The average attorney's expectations of expert computer systems will be shaped by the ways the attorney now interacts with human experts. For example, an attorney might hope to use an expert system as a partial substitute for consulting another attorney. The attorney-user will, therefore, expect the system to perform many of the same functions that its

103. Overall, a desirable expert system is one in which the costs of use, measured broadly, are outweighed by the benefits of that use. The factors mentioned in the text weigh heavily in this cost/benefit balance. See Liebowitz, *Useful Approach for Evaluating Expert Systems*, EXPERT SYSTEMS, Apr. 1986, at 86 (evaluation of an expert system should focus on its "utility" in light of the costs and benefits of its use).

104. The JUDITH system, which helps attorneys build lines of reasoning regarding various types of civil liability, is a good example of an expert legal system designed to assist an attorney by organizing legal knowledge for systematic consideration by the user. See Popp & Schlink, *supra* note 90, at 304-06. A somewhat different interactive approach is utilized in the ABF Processor system in which attorneys provide information requested by the system, the system generates a first draft of a legal document (e.g., a will or a trust agreement), and the document is then subject to further editing by the attorney-user to insure completeness and accuracy. See Sprowl & Staudt, *supra* note 12, at 704-10.

human counterpart performs. In addition, attorney users might expect even more accurate and effective performance from expert systems because of the highly accurate long-term memory and rapid computational methods available in those systems.¹⁰⁵

Recognizing that the activities of human experts provide a design benchmark for expert legal systems brings a few basic design requirements for such systems into focus. Stylistically, expert legal systems should interact with users in ways similar to the ways attorneys might interact with each other.¹⁰⁶ As when consulting another attorney, an attorney seeking computer assistance in completing an analysis will wish to actively participate in the development of that analysis to a high degree rather than passively receive bare analytic results.¹⁰⁷ Attorneys having an inherent distrust of the analytic capabilities of computers or a dislike of how computers operate will be much more comfortable accepting the results of analyses they have played a primary role in shaping rather than results essentially dictated to them by a computer system. Furthermore, the use of a participatory approach in which computer resources are used to complement the analytic strengths of the system user can produce a synergistic effect resulting in legal analyses beyond the capabilities of either attorney or computer working alone. This effect could be achieved, for example, if computer resources were used in areas like information recall or numeric calculation where computer capabilities are particularly strong. An attorney could then control portions of the analysis such as reasoning by analogy or similar analytic processes which are weakly handled by computers.

105. For example, a computer system for comparing the outcome of two calculations having legal significance could perform this task more quickly and accurately than a human counterpart, permitting an attorney relying on the system to examine a wider variety of planning options in a more efficient manner than if he made the calculations by hand. See, e.g., Hellowell, *supra* note 9, at 342.

106. Since they take place under the severe time pressures common to practice environments, attorney interactions provide a good model because they represent relatively efficient modes of communication. At the same time, they are necessarily tailored, given the underlying purpose of consultation, to convey information to solve legal problems.

107. At a minimum, this participation would require that an expert legal system have the ability to explain how it reached partial or final results and why it needs specific types of information. See, e.g., Hellowell, *supra* note 9, at 356; Schlobohm, *supra* note 71, at 781-84; cf. Brachman, Amarel, Engelman, Engelmores, Feigenbaum & Wilkins, *What Are Expert Systems?*, in BUILDING EXPERT SYSTEMS 31, 42 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983) (explanations by expert systems "reassure a human observer of the validity of a chain of inference steps"). For an example of an expert legal system with these capabilities, see Schlobohm, *supra* note 71, at 765; Schlobohm, *supra* note 69, at 64.

Finally, an attorney seeking assistance through an expert legal system must have great confidence in the accuracy of the system.¹⁰⁸ The overall quality of an expert legal system should be demonstrable through favorable comparisons of system outputs with the analyses of human experts where both the system and the experts have been given identical problems to resolve.¹⁰⁹ Furthermore, the accuracy of both the expertise captured in a system's knowledge base and the inferential reasoning processes used in a system should be manifest to attorney-users. Finally, the ability of a system to appreciate the limitations of its expertise and inform a user that a different source of assistance should be consulted will also be an important factor in establishing confidence in the system among attorney-users.

B. Characteristics of an Ideal Expert Legal System

This section will apply the functional requirements of attorney-users in order to formulate specific design goals for expert legal systems.¹¹⁰

108. An attorney using an expert legal system to shape his advice, or directly incorporating the results of a system's analyses into his advice, risks liability for negligence in selecting the system or perhaps strict liability for system errors. See Nycum & Fong, *Artificial Intelligence and Certain Resulting Legal Issues*, COMPUTER LAW., May 1985, at 1, 5; Nycum, Bowen & Van Arsdale, *Selected Legal Issues Applicable to Computerized Data Bases and Expert Systems*, COMPUTER LAW., Aug. 1986, at 8, 8-10; Willick, *Professional Malpractice and the Unauthorized Practice of Professions: Some Legal and Ethical Aspects of the Use of Computers as Decision-Aids*, 12 RUTGERS COMPUTER & TECH. L.J. 1, 14-15 (1986). Furthermore, attorneys will not use a computer system until they are confident of the system's accuracy and are familiar with its process. See Berring, *Full-Text Databases and Legal Research*, 1 HIGH TECH. L.J. 27, 38-39 (1986).

109. In determining the success of artificial intelligence programs, "the best test is usually just whether the program responded in a way in which a person could have." E. RICH, *supra* note 23, at 19. This can be determined from various kinds of experiments and protocol analyses. *Id.* At least one expert legal system has been tested or "validated" under this standard. See Michaelsen, *supra* note 6, at 164-65.

Some computer experts, however, have argued for a broader set of criteria for assessing expert systems. See, e.g., Liebowitz, *supra* note 102, at 86 (proposing that expert systems be evaluated based, *inter alia*, on their ability to be updated, their ease of use, the range of compatible computer hardware, cost-effectiveness, output comprehensiveness, output quality, and design time).

110. While it is impossible to predict precisely what types of expert legal systems may be developed in the future, intelligent activities that might be incorporated into future expert legal systems fall into four categories:

Association—processes that associate a particular user input with a rigid pattern of output without any intervening conceptual analysis;

Simple Formal—processes applying formal analytic concepts that are completely defined and used in a way that is situation independent;

Complex Formal—similar to simple formal processes, except that the way analytic concepts are applied in complex formal processes is somewhat situation dependent;

Nonformal—processes that depend on concepts that are not formally definable,

1. *Input*

An attorney will generally hope to communicate with an expert computer system in the same way he might talk to a human expert. Thus, an expert legal system would ideally satisfy the famous, and as yet unsatisfied, "Turing test," which is whether a user placed in communication with the expert system would be able to tell from the system's questions or answers that a computer was on the other end of the communication link rather than a human expert.¹¹¹

Even relatively minor differences in communication style may not be easily tolerated. For example, many persons feel far less comfortable communicating with a computer by typing on a keyboard than they would in relating the same information verbally to a human listener. More serious limitations, such as the inability of a computer system to understand more than a severely limited vocabulary of words or symbols, will cause an expert legal system to fall even further short of the desires of attorney-users.

At the input stage, an ideal expert legal system should be capable of the sorts of communication that a human expert would undertake in the course of fact gathering. For example, the expert system should provide definitions of legal terms of art either on request or, if a definition is counterintuitive, in the first instance where such a term is used.¹¹²

such as reasoning by analogy or recognition of a type of controversy based on knowledge of a paradigm case.

These categories of intelligent activities are drawn from H. DREYFUS, WHAT COMPUTERS CAN'T DO 203-17 (1972).

111. In the early 1950's, artificial intelligence researcher Alan Turing proposed the following test as a standard for determining whether a computer could think: a person and a computer are placed in two separate rooms. A second person acts as an interrogator and communicates with the two rooms by typing questions on a communication device and by receiving typed answers. The interrogator can ask questions of either the person or the computer, but is only told that the answers came from Room A or Room B. If the computer is able to fool the interrogator into thinking its room is the one occupied by the person, then Turing would conclude that the machine passed an empiric test of thinking behavior. See Turing, *Computing Machinery and Intelligence*, in COMPUTERS AND THOUGHT (E. Feigenbaum & J. Feldman eds. 1963). No computer has yet met this standard; some researchers argue that it is an aspirational standard that will never be satisfied. See E. RICH, *supra* note 23, at 18-19.

The test described in the text is actually a modified form of the Turing test since, in order to be mistaken for a human expert giving advice, an expert system would only need to be capable of dialog in the applicable field of expertise. The elimination of any need to address randomly chosen topics significantly reduces the amount of knowledge a system needs to incorporate and apply in order to simulate intelligent behavior. See *id.*

112. This assistance could be provided in either of two ways. The first time a system asked a question or made a statement containing an important term of art, that term could be defined in an accompanying sentence. Alternatively, the same definition could be given in response to a user request for help from the system. The latter approach is, of course, only useful where a user realizes that he or she needs help. See Schlobohm,

Similarly, the system should be able to explain the importance of a given question.¹¹³ It should also be able to complete partial analyses in response to partial information about a problem, and should be able to indicate how additional facts would affect the outcome of the system's conclusions.¹¹⁴ Finally, factual information, once provided to the system, should be easily alterable both to permit easy correction of erroneous data entries and to allow the "what if" analyses that often form the basis of legal planning.¹¹⁵ In addition to these considerations, the attorney should have complete confidence that all relevant information is being considered by the system in producing a given legal analysis. In part, the scope of information sought by a system depends on the completeness of its knowledge base, an issue addressed below. However, to provide an independent means to gauge the thoroughness of its fact gathering, a system should inform users of the range of input information actually relied upon in a particular system analysis.¹¹⁶

supra note 71, at 784-85.

113. This amounts to explaining "why" a given question has legal significance. *See id.* at 783-84.

114. One way to convey this information is for the system, upon inquiry by a user, to display the reasoning path it is following along with the reasoning steps it has already confirmed. This approach is illustrated by the following portion of a Tax Advisor consultation session (user responses are italicized):

Is the client willing to relinquish possession and enjoyment of assets and power to alter, amend, revoke, or terminate any trust in which the assets are placed?

WHY?

This will aid in determining whether the client should make permanent gifts. It has already been established that client has an estate that is substantial enough, given his present age and health, that he can afford to forego the income from gifted assets, and to pay the expenses associated with making gifts. Therefore, if . . .

A: Client is willing to relinquish possession and enjoyment of assets and power to alter, amend, revoke or terminate any trust in which the assets are placed, or

B: is willing to transfer term life insurance, or whole life that is borrowed up, to an irrevocable trust, . . .

then client should explore permanent inter vivos gifts.

See Michaelsen, *supra* note 6, at 158.

115. For example, a system capable of analyzing whether one business form is preferable to another from a tax standpoint could be programmed to operate in a planning mode. In this mode, the system would permit users to describe a projected business operation, obtain a tax characterization from the system, modify part of the description of that operation, and immediately receive a recharacterization of the modified operation. Through this process, the system can assist the user in formulating business plans by testing various alternatives. *See* Hellowell, *supra* note 9, at 344-45, 346.

116. This type of information can be disclosed when a system reviews how it has reached a particular conclusion or piece of advice. For example, one existing expert system lists the sequence of inferences drawn by the system when a user asks how the system reached a result. *See* Schlobohm, *supra* note 71, at 782-83.

2. Knowledge Bases

Obviously, a knowledge base should be as comprehensive as is necessary to serve the specific goals of a system. A thorough knowledge base serves both to increase an expert system's analytic strengths and to inspire user confidence. The knowledge base should be easily alterable so that new knowledge can be added and old knowledge deleted as legal standards change or develop. In this respect, new knowledge might involve revision of existing inference rules or merely the specification of new exceptions to previously unqualified rules. Corresponding sets of term definitions will also need to be added to the database as inference rules are altered.

An ideal expert legal system would also retain information about problems and related expertise in the same forms and groupings that an attorney-counterpart would. Where such information ordering is not compatible with computer processing, information should be organized in a way that is at least understandable to attorneys seeking to reconstruct the information used by the system.¹¹⁷

Clearly understandable information storage serves several functions. Attorneys will be able to determine what information has been used by a system in reaching a particular conclusion, and will thus be able to easily hypothesize about which circumstances, if changed, might produce different results. Further, attorneys can readily examine the inference rules being used by a system to determine whether important considerations have been left out.

3. Inference Processing

As with the other components of expert systems, reasoning processes in an ideal expert system should approximate those of attorneys. Attorneys then will be able to examine and understand the reasoning underlying a particular conclusion to determine if it warrants his or her confidence. Moreover, system designers will be able to periodically compare the logic of a system with that used by attorneys in addressing new problems in order to determine if new inference programming of the system is necessary.

117. In constructing a knowledge base:

notational convenience is a virtue because most expert systems applications require the encoding of substantial amounts of knowledge, and this task will not be an enviable one if the conventions of the representation language are too complicated. The resulting expressions should be relatively easy to write and to read, and it should be possible to understand their meaning without knowing how the computer will actually interpret them.

P. JACKSON, *supra* note 17, at 30.

The inference processing components of expert legal systems should thus meet several criteria. Of course, the reasoning approach in a particular system (as chosen from among the available computer reasoning schemes discussed in Section II) should be an efficient match for the characteristics of the system's knowledge base and the problems the system must resolve.¹¹⁸ In addition, where the most efficient means for a computerized system to reach a solution to a problem do not parallel the way an attorney would solve the same problem, the system should at least be able to justify the correctness of its answer in a manner that an attorney would find understandable and persuasive. For example, an expert legal system designed to assist in solving a particular type of tax planning problem might have sufficient processing speed to simply generate and test a variety of potential problem solutions rather than to initially attempt to identify the one solution that is acceptable. On the other hand, an attorney might be more likely to attempt to develop a single solution through deductive processes.¹¹⁹ Such a difference in problem solving approaches will not detract from the usefulness of an expert legal system if the system is at least able to "defend" its answer by describing why the answer meets the tax planning objectives at stake.

Finally, in addition to being able to manipulate substantive legal rules, the inference processing of an ideal expert legal system should incorporate many practical considerations or "rules of thumb" that influence the steps taken by the system in solving problems. For example, in the interests of efficiency an expert system must determine which tax planning vehicle or type of factual information to examine first in trying to reach a particular planning goal. A "rule of thumb"

118. Obviously, the accuracy of the inference process in performing a particular type of analysis is one important measure of the suitability of that process. However, other features are also important, such as how easy the process is to understand, modify, use, and transport to other computers. See Liebowitz, *supra* note 102, at 86.

Efficiency, as used here, is a measure of the system's overall performance in all of these areas. An inference process that is not well tailored to solving a particular type of problem may still produce accurate results, but often at an unacceptably slow rate. Such a system is much less efficient than a system incorporating a more carefully designed inference process that minimizes the unnecessary analyses performed by the system and thereby reaches successful results in less time. Cf. Michaelsen, *supra* note 6, at 163-64 (discussing steps taken to reprogram an expert legal system to make it more efficient).

119. For example, a system for planning corporate reorganizations might begin with a description of a corporation's shareholders and the characteristics of the desired post-reorganization pattern. The system could then generate combinations of possible reorganization steps that would lead from the initial circumstances to the desired final result. Each of these combinations of steps could be tested to determine if they qualify for favorable tax treatment. Through this mechanism, such a system could perform an analysis equivalent to that which an attorney might perform by deduction directly from the applicable tax law. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 886 n. 117.

concerning types of information most commonly leading to useful tax planning advice would be needed to tell the system which among several good alternatives to examine first.

4. Output

An ideal expert system would describe its expert conclusions in understandable terms and would be capable of reconstructing and explaining the logic underlying those conclusions.¹²⁰ In addition, an expert system's response should be as practical and comprehensive as possible. For example, rather than just describing what tax planning options would be desirable based on a person's income or property holdings, an expert system might create an initial draft of trust agreements or other documents necessary to implement the desirable options.¹²¹

A sophisticated system might also be programmed to provide a follow-up response as the expertise which formed the basis for the system's past advice evolved or changed.¹²² For example, in the tax planning system described above, this could be achieved by programming the computer to keep records of the advice given in response to an attorney's inquiry and the tax law provisions relied upon. When a tax law provision is changed, the system could review its records, identify each client given advice based on the old provision, and draft a letter suggesting that these clients may need a new tax plan in light of the changes in the law. In this way, a legal expert system could keep track of a client's legal status and could monitor when a client should come in for a legal "checkup."

120. Explanation is an important feature of any expert system because "[t]he consumers of automatic advice need to be convinced that the reasoning behind a conclusion is substantially correct, and that the solution proposed is appropriate to their particular case." P. JACKSON, *supra* note 17, at 207.

121. Cf. Boyd & Saxon, *The A-9: A Program for Drafting Security Agreements Under Article 9 of the Uniform Commercial Code*, 1981 AM. B. FOUND. RES. J. 639, 652 (program creating security agreements complying with Article 9 of the Uniform Commercial Code is capable of simultaneously generating ancillary documents like financing statements and continuation statements).

122. An expert legal system:

could monitor legal data bases for the purpose of identifying clients affected by law changes. For example, clients given legal help with estate planning, wills, or business contracts would want to be notified when laws were enacted that nullified key provisions in those documents. These systems would work by setting up links between particular documents and the pieces or types of legislation that could affect them and then notifying the attorneys when the linked legislation was changed.

D. WATERMAN, *supra* note 1, at 225; see also Gray, *Law & Technology Conference Expert System Workshop Report*, in *COMPUTING POWER AND LEGAL REASONING* 621, 624 (C. Walter ed. 1985) (proposing a similar system).

IV. EVALUATION OF EXISTING EXPERT LEGAL SYSTEMS

This section contains a review of the history, objectives, operation, and merits of some existing expert legal systems.¹²³ This review serves several purposes. It serves to describe the many functions that existing systems can perform, thereby demonstrating the potential breadth of future applications for expert legal systems.¹²⁴ It also illustrates the wide

123. The expert legal systems discussed in this Section are organized by the type of legal analysis they perform.

124. This breadth is illustrated by the following list of potential uses of expert legal systems developed by expert system designers at a recent workshop:

1. Aiding legislative drafting by testing a draft against a legal database and linguistic standards, and having the computer make appropriate suggestions;
2. Researching legal databases such as WESTLAW and LEXIS on the basis of a statement of facts or concepts (conceptual retrieval as opposed to the current key word searches);
3. Generating ideas and advising the attorney of arguments for and against a proposition, and also how to strengthen the arguments in a particular case;
4. Advising an attorney on strategy and tactics in negotiations;
5. Evaluating a case as to settlement or legal worth;
6. Evaluating the consistency with prior decisions of a proposed administrative decision in a discretionary area;
7. Aiding document drafting of contracts, wills, and other documents by testing for consistency with existing law, internal consistency, and consistency with linguistic standards;
8. Assisting decision making in which little or no discretion is involved;
9. Planning transactions, such as business mergers, with tax and other legal consequences by presenting alternative scenarios and identifying their legal consequences;
10. Predicting consequences of proposed legislation or draft contracts;
11. Finding legal authority which is inconsistent with proposed laws;
12. Evaluating the effectiveness of existing laws and identifying laws which may need to be modified;
13. Training and disseminating information on the law;
14. Interviewing clients for information relevant to identification of the nature of their legal problem;
15. Informing people of the legal consequences of particular acts in order to enable non-attorneys to know the legality of proposed acts or past acts, and if communication with an attorney is required to obtain a complete answer;
16. Preserving institutional expertise;
17. Reviewing a legal database against new laws or cases and modifying it to keep it current;
18. Identifying clients whose legal affairs may have been affected by changes in the law so that an attorney can determine whether to contact them regarding the change.

Id. at 622-24; *cf.* *Legal Decision Systems*, *supra* note 13, at 223-24 (primary application areas for expert legal system development include case management, monitoring, legal interpretation, and document generation).

range of programming and design approaches now available for creating expert legal systems. Moreover, by evaluating the advantages and disadvantages of present systems, it provides design guidelines for developers of future legal expert systems.

A. Interpretation

Applying legal rules to produce legal interpretations of factual circumstances lies at the heart of many lawyering tasks, from assessing a client's potential civil liability for past actions to identifying the probable tax consequences of present client conduct. Several computer systems have been developed to make expert legal interpretations in limited factual circumstances¹²⁵ or to assist attorneys in making such interpretations.¹²⁶

1. TAXMAN¹²⁷

Developed during the early 1970's by Professor L. Thorne McCarty, TAXMAN deals with the classification of corporate reorganizations for federal income tax purposes. Specifically, TAXMAN is capable of determining whether a particular corporate transaction is a Type "B," "C," or "D" reorganization within the meaning of Internal Revenue Code section 368(a).¹²⁸ This area of law was chosen as the focal point of the TAXMAN project for several reasons.¹²⁹ First, the rules for distinguishing among these types of transactions are succinctly set out in the

125. For example, the CORPTAX system examines stock redemption patterns in corporate reorganizations to determine if the reorganizations have characteristics specified in the Internal Revenue Code that will cause them to receive special tax treatment. See Hellawell, *supra* note 15, at 1363.

126. For example, the JUDITH system assists an attorney in developing a chain of legal reasoning. It does so by presenting the attorney with a series of reasoning options through a conversational process. See Popp & Schlink, *supra* note 90, at 304-06.

127. McCarty, *Intelligent Legal Information Systems*, *supra* note 8, at 267-76; McCarty, *Reflections on TAXMAN*, *supra* note 8, at 837.

128. Section 368(a) of the Internal Revenue Code describes categories of corporate reorganizations which may be conducted without taxable gain or loss. See I.R.C. §§ 354, 361(a), 368(a) (1982). These categories of reorganizations are known colloquially by the letter of the subsection of § 368(a)(1) in which they are defined. Thus, the three types of reorganizations analyzed by TAXMAN are:

B TYPE—“the acquisition by one corporation, in exchange solely for . . . [the] voting stock . . . of another corporation,” if immediately thereafter the acquiring corporation has control of the other corporation. I.R.C. § 368(a)(1)(B).

C TYPE—the acquisition by one corporation of “substantially all of the properties” of another corporation, again in exchange “solely for . . . voting stock” of the acquiring corporation. I.R.C. § 368(a)(1)(C); see also I.R.C. § 368(a)(2)(B).

D TYPE—the transfer of assets by one corporation to another corporation of which the transferor immediately thereafter is in control. I.R.C. § 368(a)(1)(D).

129. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 842-50.

Internal Revenue Code and accompanying IRS regulations.¹³⁰ Second, where these statutory or regulatory standards are unclear, detailed case law explicitly resolves most of the ambiguities. Consequently, answers to questions in this area of law depend heavily upon analyses which are highly structured in accordance with the underlying statutory scheme, yet fairly complex because of the presence of judicial interpretations of that scheme.

TAXMAN records knowledge in terms of relationships between factual states and concepts. The objective of this design was to identify and record legal relationships in a workable approximation of the way knowledge in this area is represented in the human mind.¹³¹

An example of how TAXMAN represents a particular legal rule illustrates this design technique. The statutory definition of a Type B reorganization is as follows:¹³²

the acquisition by one corporation, in exchange solely for all or a part of its voting stock . . . of stock of another corporation, if, immediately after the acquisition, the acquiring corporation has control of such other corporation

A translation of this definition into logical rules might read as follows:

If (1) there exists an ACQUISITION OF STOCK in exchange for VOTING STOCK, and (2) there exists a CONTROL relationship following the acquisition, then there exists a B TYPE REORGANIZATION.

In this definition each of the capitalized terms (except "B TYPE REORGANIZATION") represent a factual state or relationship which TAXMAN must identify as being present or absent. "B TYPE REORGANIZATION" is a factual state considered present if both parts (1) and (2) of this test are satisfied.

TAXMAN represents the definition of a B TYPE REORGANIZATION as follows:¹³³

```
(ACQUISITION ACQ1
  <acquirer (CORPORATION C1>
  <acquired prop
  (STOCK S1
    <issuer (CORPORATION C2)>>
  <transferred prop
```

130. See I.R.C. §§ 268, 354, 355, 356 (1982).

131. See McCarty, *Reflections on TAXMAN*, *supra* note 8, at 850-62.

132. I.R.C. § 368(a)(1)(B) (1982).

133. McCarty, *Intelligent Legal Information Systems*, *supra* note 8, at 274.

```

(STOCK S2
  <issuer (CORPORATION C1)>
  <voting YES>>
<time1 (STATE T1)>
<time2 (STATE T2)>
  (CONTROL CON1
  <controller (CORPORATION C1)>
  <controlled (CORPORATION C2)>
  <time (STATE T2)>)

```

The ACQUISITION language of the program corresponds to part (1) of the Internal Revenue Service rule, while the CONTROL language corresponds to part (2) of the rule. Both standards must be met for the system to conclude that a B type reorganization is present. These definitions of ACQUISITION and CONTROL are not complete since each depends on definitions of lower-level concepts. For example, CONTROL depends on the definition of the relationship "OWN" and a determination whether CORPORATION C1 "owns" 80% or more of the STOCK of CORPORATION C2 at time T2.¹³⁴ This illustrates how relatively complex legal tests can be implemented in computer systems by building upon earlier definitions of basic concepts.

The TAXMAN system is an early example of an expert legal system based on semantic information processing. Such an implementation involves the specification within a domain of knowledge of significant "things" and "relations," with the latter typically being represented as a network of links between instances of the former. Using such semantic networks, McCarty stored representations of certain legally significant transactions in TAXMAN. The system was then able to characterize user actions by applying pattern recognition techniques to determine if the relationships present in the user's facts corresponded with patterns known to the system.

Although a useful prototype, TAXMAN has several limitations as an expert system. First, information must be presented to the system in a rigid format, making TAXMAN, as its developer noted, "inaccessible to most potential users."¹³⁵ Second, it deals with a very narrow field of law.¹³⁶ Third, TAXMAN was designed to characterize completed corporate transactions, and thus it provides only awkward assistance to parties wishing to plan future corporate activities to achieve a predetermined tax consequence. Finally, TAXMAN's requirement that the full factual circumstances of a transaction be entered into the system before

134. *Id.*

135. McCarty, *Reflections on TAXMAN*, *supra* note 8, at 882.

136. *Id.*

it begins its analysis fails to utilize user time as efficiently as a system that accepts a partial factual description, produces a partial analysis, and then asks the user for only the information needed to complete the analysis.¹³⁷

Despite these limitations, TAXMAN was a provocative early work, illustrating how semantic information processing techniques could be applied to legal knowledge and how legal knowledge could be formalized to accommodate those techniques. TAXMAN remains one of the most fully realized "deep knowledge" systems in the legal field.¹³⁸

2. *Tax Advisor (Schlobohm)*¹³⁹

Tax Advisor is another expert system designed to produce legal interpretations in the tax field. Developed by Dean Schlobohm, Tax Advisor was created to demonstrate to attorneys the usefulness of expert legal systems implemented on microcomputers. Potential system users were presumed to have little or no understanding of computers. Consequently, much of the development effort centered not on the incorporation of broad analytic powers into the system, but on designing an "interface" between the system and attorney-users which would allow attorneys to use Tax Advisor without understanding the computer language in which it is written.¹⁴⁰

Tax Advisor's expertise concerns the constructive ownership of stock for income tax purposes. Its knowledge base is derived from the ownership attribution rules in Internal Revenue Code section 318.¹⁴¹ Under these rules, stock ownership is attributed to persons other than the individuals or entities holding title to the stock. For example, the rules provide that an individual is considered to be the owner of shares directly or indirectly owned by his or her spouse, children, grandchildren and parents.¹⁴² The rules in sections 318 are unambiguous, but tedious to apply. Part of this tedium stems from their recursive nature—that is,

137. See *id.* at 885 n.114.

138. TAXMAN is a "deep knowledge" system because its expert conclusions are derived from an internally recorded model of legal standards, in contrast to a rule-based or "shallow knowledge" system in which narrow judgmental rules are used to produce expert conclusions. See Michaelsen, Michie & Boulanger, *supra* note 73, at 305-10. TAXMAN's author, Professor L. Thorne McCarty, believes that deep conceptual models are generally preferable as bases for expert legal systems, primarily because a conceptual model developed for a legal analysis and planning system could also form the foundation for a legal materials retrieval system in the same substantive area. See McCarty, *Intelligent Legal Information Systems*, *supra* note 8, at 265.

139. Schlobohm, *supra* note 71, at 765; Schlobohm, *supra* note 69, at 64.

140. See Schlobohm, *supra* note 71, at 767.

141. I.R.C. § 318 (1982).

142. I.R.C. § 318(a)(1) (1982).

the need to apply the rules repeatedly to produce a complete analysis. This recursiveness follows from the provision of the attribution rules that where that an individual's stock ownership is attributed to a second party under section 318, that second party's ownership can be attributed to a third party through an additional application of the section.¹⁴³

Tax Advisor is written in PROLOG, a computer programming language popular among creators of expert systems.¹⁴⁴ PROLOG permits easy definition of the knowledge bases that are at the heart of most expert systems.¹⁴⁵ PROLOG also promotes efficient expert system development by permitting a system author to define tests initially and then to refer to them by shorthand labels thereafter.¹⁴⁶ For example, assume "spouse(H,W)" is PROLOG shorthand for a test that gives a positive response if H is married to W, and "owns(W,N,C)" denotes that individual W actually holds legal title to N shares of stock in C corporation. These terms can now be used to define more complex tests. For example, since section 318 provides that a husband constructively owns his spouse's shares,¹⁴⁷ the number of shares constructively owned by H under section 318 due to marriage ("num-con-own(H,N,C)") can be defined in PROLOG by the following statement:¹⁴⁸

num-con-own(H,N,C):-spouse(H,W), owns(W,N,C).

In addition to being an easier way to write programs, these shorthand terms make the programs easier to understand and modify.

Tax Advisor has several other features which make it easy to operate. One such feature is the system's ability to explain many of the terms it uses in its questions. This is accomplished through a "Help" command. For example, one of the questions asked by Tax Advisor is:

Does your client have a spouse?¹⁴⁹

If the word "help" is given in response, Tax Advisor explains:

A spouse does not include one who is legally separated from [the client] under a decree of divorce or separate maintenance. Section 318(a)(1)(A)(i).¹⁵⁰

While this is not a complete definition of the term "spouse," it contains important information on the ways the legal definition of this word

143. For example, if a trust owns 50% of the stock of corporation X, stock of corporation Y owned by corporation X which is attributed to the trust may be further attributed to the beneficiaries of the trust. See Treas. Reg. § 1.318-4(a) (1968).

144. See *supra* note 25 and accompanying text.

145. *Id.*

146. *Id.*

147. I.R.C. § 318(a)(1) (1982).

148. See Schlobohm, *supra* note 71, at 766-67.

149. See *id.* at 784.

150. *Id.*

differs from common usage.¹⁵¹

A second type of explanation given by Tax Advisor is a description of the reason it is asking a user for a piece of information. For example, when Tax Advisor asks,

Is [the client] an individual?¹⁵²

and the user types "why," Tax Advisor supplies the following explanation:

An individual is deemed to own the stock owned by his family.
Section 318(a)(1).¹⁵³

A third type of explanation is available from Tax Advisor if a user has doubts about how the system reached its conclusions or if the user wishes to identify facts that were the basis for a given answer. Tax Advisor can review the analytic rules it utilized, the tentative conclusions it reached, and the reasoning path it followed in formulating a final conclusion. A simple reasoning explanation given by Tax Advisor might contain the following:

The number of shares of Corporation X stock constructively owned by [the client] as a result of section 318(a)(1) is 1,000, determined as follows:¹⁵⁴

An individual is deemed to own the stock owned by his family.
Section 318(a)(1).

FACT: [The client] is an individual.

A spouse is a family member. Section 318(a)(1)(A)(i).

FACT: [W] is the spouse of [the client].

FACT: [W] actually owns 1,000 shares.

FACT: The client has no more family members.¹⁵⁵

Like the other expert legal systems described in this Article, the success of Tax Advisor stems from its reliance upon the analytic strengths of computers. Tax Advisor is successful primarily because the rules in section 318 are well defined and not subject to varying interpretations by numerous cases.¹⁵⁶ The principal difficulties involved in applying these rules concern the extremely repetitive nature of this process. Human

151. Of course, the usefulness of this explanation capability is limited by the need for a Tax Advisor user to ask for help before any explanation is given. In the instance quoted in the text, for example, a user of the system might simply assume that the term "spouse" includes any person the taxpayer is still legally married to, regardless of legal separation. Unless the user knew to ask for help, he or she might simply overlook the correct meaning. *Id.* at 784-85.

152. *See id.* at 783-84.

153. *Id.*

154. *See id.* at 782-83.

155. *See id.* at 782-83.

156. *Id.* at 785.

repetition of such analyses takes significant amounts of time and can lead to error; by contrast, computers can handle this task quickly and accurately.¹⁵⁷

Tax Advisor represents several advances over the other expert legal systems described in this Article. Its explanation capability is far more sophisticated than that of most of the other systems. Further, Tax Advisor demonstrates that an effective expert legal system can be implemented with relative ease through use of PROLOG or similar programming languages specially designed for representing and applying symbolic knowledge.¹⁵⁸ Specialized computer programming languages like PROLOG offer excellent opportunities for attorneys and legal academics lacking extensive computer programming training to author expert systems.¹⁵⁹

Tax Advisor's limitations derive largely from the specific design goals used in creating the system. Tax Advisor's knowledge base was not expanded beyond the narrow subject matter of attribution issues under section 318 because it was intended primarily as an experimental system demonstrating the extensive explanatory capabilities possible in an expert legal system implemented on a microcomputer.¹⁶⁰ Furthermore, the limited processing power of the microcomputer and the version of PROLOG used to create Tax Advisor imposed significant design constraints. Because an overall limit was placed on the processing power available to the analytic and explanatory portions of Tax Advisor, greater explanatory capabilities could often be achieved only by limiting some of the analytic power of the system.¹⁶¹ To accommodate this trade-off, Tax Advisor was implemented in two versions, one in which the full range of section 318 analyses were implemented but with little explanatory capability, and a second version in which only a few of the section 318 attribution rules were carried out but which had the full explanatory capabilities described above.

Fortunately, these limitations can be overcome in subsequent implementations of Tax Advisor and similar systems. The two existing versions of Tax Advisor illustrate, albeit in separate systems, how to

157. See *id.* at 779-80.

158. For overviews of computer programming languages commonly used in expert system development, see P. HARMON & D. KING, *supra* note 2, at 79-89; E. RICH, *supra* note 23, at 389-406.

159. A number of legal experts have authored expert legal systems in their own areas of expertise. See *e.g.*, Hellawell, *supra* note 15, at 1363; Boyd, *supra* note 5, at 699. For an assessment of the problems a legal expert may encounter in this authoring process, see Hellawell, *supra* note 15, at 1364-65.

160. Schlobohm, *supra* note 71, at 767.

161. See *id.*

implement both analytic breadth and explanatory capabilities. All that is needed to combine these capabilities in a single system is either a larger target computer or a more efficient microcomputer implementation of the PROLOG language. The former is available at a price; the latter has recently become available in the form of several new PROLOG language products for microcomputers.¹⁶²

3. CORPTAX¹⁶³

CORPTAX is a computer program developed by Columbia University Professor Robert Hellawell to demonstrate four important functions computers can perform for attorneys: (1) the orderly and comprehensive accumulation and arrangement of facts relevant to a problem, (2) the completion of mathematical calculations and logical deductions necessary to solve a problem, (3) the retention of data on complex legal standards in forms that can guide problem solving, and (4) the description of alternative analyses in controversial legal areas, along with arguments and precedents supporting each. CORPTAX analyzes a specific corporate taxation question: does a stock redemption qualify for favorable tax treatment under the provisions of Internal Revenue Code section 302(b)(2) and the related attribution rules under section 318? These provisions determine whether a purchase of a shareholder's stock by an issuing corporation will be treated as a sale of the shares, with the consequent gains potentially subject to capital gains treatment, or as a dividend with gains taxed at ordinary income rates.

CORPTAX differs from TAXMAN and Tax Advisor in several significant ways. One fundamental difference is the lack of a formal knowledge representation scheme in CORPTAX. Tax rules are implicitly represented in CORPTAX's programming through the sequence of questions the program asks the user and the series of tests that are applied to the user's answers.

A portion of a typical CORPTAX session will illustrate this approach. CORPTAX begins by asking the user a variety of questions,

162. For example, Turbo PROLOG, a new version of PROLOG for use with IBM personal computers and compatibles, can work with considerably more computer memory than the version of PROLOG used to develop the Tax Advisor. This, in turn, should permit much more elaborate expert systems to be implemented on these small computers.

163. Hellawell, *supra* note 15, at 1363. The usefulness of CORPTAX has been eclipsed by the 1986 Tax Reform Act, in which capital gains and ordinary income are taxed at the same rate. See Tax Reform Act of 1986, Pub. L. No. 99-514, § 301(a) (repealing provisions of I.R.C. § 1202 previously providing for special capital gains treatment).

almost all requiring either a "yes" or "no" answer or a numerical response. Here are some samples:¹⁶⁴

Does REDCORP [the corporation redeeming its stock] have outstanding more than one class of voting stock? Answer Yes or No. If you would like information on the meaning of voting stock — type XXX.

Does REDCORP have outstanding more than one class of common stock? Answer Yes or No. If you would like information on the meaning of common stock — type XXX.

How many types or classes of REDCORP voting stock are outstanding? Program will accept up to 5.

CORPTAX goes on to ask additional questions about the circumstances of the redemption transaction of the user. It then reaches the following types of conclusions:¹⁶⁵

Redemption passes the voting power tests of I.R.C. Sec. 302(b)(2)(B) and (C).

Redemption does not qualify under I.R.C. Sec. 302(b)(2). Although it passes the 50% voting stock test and the 80% voting stock test it flunks the value test found in the last paragraph of I.R.C. Sec. 302(b)(2)(C).

If third parties have common shares redeemed worth — \$119,929.00 (which your answers indicate) then to pass the value test of Sec. 302(b)(2)(C) the minimum value of common shares that must be redeemed from the total of shares actually owned by taxpayer plus any shares attributed to taxpayer is — \$57,998.00.

These examples illustrate both the strengths and weaknesses of CORPTAX. CORPTAX's principal strengths derive from the fact that it is very easy to use. The user need only answer a carefully sequenced series of relatively simple questions. Each question asks for a limited factual assessment that can be made by someone having little legal sophistication. For users who are not familiar with the terms of legal art included in some of the questions, an explanation of the terms is available as an optional supplement to each question.

Unfortunately, the careful sequencing of CORPTAX's questions also reflects one of the weaknesses of the system: it is not easily updated or expanded. The programming statements which cause the system to ask direct questions and which control how the responses are processed are carefully crafted and intertwined to serve their analytic purpose in CORPTAX's original design and thus cannot be easily removed or changed without fundamentally altering the program itself.

164. *Id.* at 1370.

165. *Id.* at 1374.

Furthermore, the relationship between the legal rules governing the interpretation of stock redemptions and the actual programming of CORPTAX is far from obvious.¹⁶⁶ That relationship will only be clear to a person with both tax law and programming expertise who takes time to study the program's design. This means that a user without both these types of knowledge must rely wholly upon the expertise of the system developer (or subsequent system modifiers) for the accuracy and completeness of the legal analyses produced by CORPTAX. Because user confidence is a key prerequisite to the utility of an expert system,¹⁶⁷ many persons may be hesitant to use systems like CORPTAX because the analytic methods in these systems are inaccessible to users.

4. JUDITH¹⁶⁸

Developed by German researchers Walter Popp and Bernhardt Schlink of the Universities of Heidelberg and Darmstadt, JUDITH was created to demonstrate the usefulness of an interactive system in which a chain of legal reasoning is built up step-by-step through a dialog between lawyer and computer. Rather than producing expert interpretations itself, JUDITH guides attorneys in analyzing the probable scope of a party's civil liability for past actions. The system's expertise lies in its structured records of criteria for civil liability and the systematic presentation of those criteria for review by lawyers using the system.¹⁶⁹

JUDITH applies a backward-reasoning, goal-driven approach to guide an attorney through the analysis of a case.¹⁷⁰ An attorney using JUDITH is first presented with a range of possible remedies and must select one acceptable to his or her client. Based on that selection, JUDITH then displays all areas of the law where the chosen remedy is available (e.g., contracts, torts, property). Once one of these areas is chosen, the system presents a list of the types of causes of action in that field.¹⁷¹ The user selects one cause of action (e.g., negligence) and the system displays its elements. Each of these elements has corresponding

166. For example, in CORPTAX's programming, the variable recording the number of voting shares being redeemed is labeled as "TPVOTSHSRED." See *id.* at 1394. While this label would be sufficient to remind the programmer of the system what it referred to, it would be unlikely to be intelligible to someone examining the program for the first time.

167. To some extent, this limitation is mitigated by the textual explanations CORPTAX gives of some of its answers. See *id.* at 1376-81. However, many of the analytic steps undertaken by CORPTAX are not explained and could only be understood upon close inspection of the system's programming. See *id.* at 1393.

168. Popp & Schlink, *supra* note 90, at 303.

169. See *id.* at 304-09.

170. See *id.* at 304-05.

171. *Id.*

sub-elements. For example, under the topic of breach of duty, the system addresses statutory duties, special duties defined by common law, and the reasonable man standard.¹⁷² The user can request that these sub-elements be broken down further into lesser elements.

Once a user has progressed downward along a chain of concepts, he or she can ascend or descend another analytic chain. For example, after multiple levels of information on breach of duty are explored, the user can examine other elements of negligence liability (such as probable cause) and the analytic chains associated with those elements.¹⁷³ Thus, JUDITH resembles an automated outline of liability considerations that assists an attorney-user in focusing attention on portions of the outline that are relevant to the attorney's particular controversy.

JUDITH was implemented in FORTRAN, a general purpose programming language most often used for numerical analyses.¹⁷⁴ The program utilizes two types of data. The first, contained in a "construction file," includes the core of JUDITH's knowledge in a logical structure representing the hierarchy of topics the system addresses.¹⁷⁵ The second, contained in a "premises file," includes descriptions of the topics addressed by the system along with numerical identifiers used throughout the system to denote each topic.¹⁷⁶ Each of these files is indexed, so that any record in them can be found easily, and the order of the records does not significantly affect system performance.¹⁷⁷

The splitting of JUDITH data into these two files involves several design advantages.¹⁷⁸ Separating the premises file from the construction file allows a particular topic to be incorporated easily at several places in either hierarchy. In addition, due to changing case law or statutory definitions, JUDITH is able to change the description of a topic without altering its hierarchical relationships to other topics. Alternatively, additions can be made to the analytic hierarchy without any need to change existing topic descriptions.

Finally, the splitting of knowledge hierarchies from topic descriptions permits controversial topics to be the subject of two or more duplicate topic descriptions, allowing the system user to apply each to determine if differences in the descriptions will affect the outcome of his or her case. For example, if it was unclear under existing precedents in a given jurisdiction whether a knowing misrepresentation of fact (rather

172. *Id.*

173. *Id.* at 305-06.

174. *Id.* at 311.

175. *Id.* at 306-07.

176. *Id.*

177. *Id.* at 307.

178. *Id.*

than a negligent one) was necessary to establish fraud,¹⁷⁹ JUDITH could incorporate two parallel descriptions of the elements of fraud, one including a requirement of intentional misrepresentation and the other specifying the lesser negligence standard. A user interested in a complete analysis of a potential fraud action could utilize both these fraud descriptions to determine whether the differences between them affected the analytic outcome in the particular case under scrutiny.

JUDITH has several severe limitations as an expert system. First, it is only useful in fields which can be adequately represented in a formal hierarchy of topics. Second, utilization of JUDITH will be worthwhile only where the hierarchy of information needed to analyze a case is complex, yet familiar enough to the attorney-user so that he or she can make a proper analysis once given prompts by JUDITH. Third, the implementation of JUDITH in the FORTRAN language severely handicaps the system since this language handles text and knowledge processing relatively poorly.¹⁸⁰ Fourth, the representation of knowledge as links between multiple topic identifiers in JUDITH construction files may make these construction files difficult for unsophisticated users to interpret, thus limiting user confidence in the system. Finally, although JUDITH does have some capacity to explain itself, the system cannot provide explanations at intermediate stages of its operation regarding what additional information will be necessary to finish its analyses and why that further information will be significant.

B. Planning

Expert planning systems attempt to determine the best course of future action, usually within specified constraints.¹⁸¹ Planning client actions within legal restrictions or to achieve certain legal benefits is a common practice activity for many attorneys. Several existing expert legal systems attempt to assist lawyers in these planning roles.

179. See W. PROSSER & W. KEETON, PROSSER AND KEETON ON THE LAW OF TORTS, 741-65 (5th ed. 1984).

180. As a scientific programming language designed primarily to manipulate and analyze numeric data, FORTRAN originally included almost no means for manipulating character data. While such capabilities have been implemented in some later versions, text handling remains a poorly realized afterthought in FORTRAN. See, e.g., BARRON, AN INTRODUCTION TO THE STUDY OF PROGRAMMING LANGUAGES 114 (1977).

181. See D. WATERMAN, *supra* note 1, at 35-36; *Overview of Expert Systems*, *supra* note 51, at 14-15.

1. TAXADVISOR (Michaelsen)¹⁸²

The most sophisticated example of a legal planning system to date is TAXADVISOR, an expert system that gives estate planning advice. The similarity of its name to Dean Schlobohm's PROLOG-based Tax Advisor is merely coincidental—the two systems are unrelated and address different tax questions.¹⁸³ Developed by Robert H. Michaelsen of the University of Nebraska, TAXADVISOR was created to test the usefulness of a skeletal expert system base or "shell" in implementing expert legal systems.¹⁸⁴ The shell used by Michaelsen, EMYCIN,¹⁸⁵ had previously produced successful expert systems in a number of fields.¹⁸⁶

EMYCIN¹⁸⁷ contains modular programming that controls aspects of expert system operations, but does not contain production rules related to any particular subject matter.¹⁸⁸ To create an expert system using EMYCIN, a developer need only add "IF-THEN" rules. Consequently, the developer can concentrate on creating accurate, complete inference rules without having to worry about the task-independent aspects of the system.

The "IF-THEN" production rules in TAXADVISOR relate factual circumstances to appropriate estate planning advice. The system provides advice by examining a number of potentially advantageous estate planning actions and determining which of the factual predicates recommending these actions are present. It gathers factual information by

182. Michaelsen, *supra* note 6, at 149.

183. Schlobohm's system addresses stock ownership attribution rules under federal income tax laws. See Schlobohm, *supra* note 71, at 767-71, 776-80. In contrast, Michaelsen's system concerns a range of tax planning issues under federal gift and estate tax laws. See Michaelsen, *supra* note 6, at 152.

184. A skeletal system or expert system "shell" is simply a stripped-down expert system—that is, a system that has had all domain-specific knowledge removed, leaving its inference engine and other support facilities. See P. JACKSON, INTRODUCTION TO EXPERT SYSTEMS 107 (1986); D. WATERMAN, *supra* note 1, at 83.

185. EMYCIN and related skeletal expert systems are described more fully in Barstow, Aiello, Duda, Erman, Forgy, Gorlin, Greiner, Lenat, London, McDermott, Nii, Polidakis, Reboh, Rosenschein, Scott, van Melle & Weiss, *Languages and Tools for Knowledge Engineering*, in BUILDING EXPERT SYSTEMS 283, 286-301 (F. Hayes-Roth, D. Waterman & D. Lenat eds. 1983) [hereinafter *Languages and Tools*]; D. WATERMAN, *supra* note 1 at 112-18.

186. E.g., Aikins, Kunz, Shortliffe & Fallat, *PUFF: An Expert System for Interpretation of Pulmonary Function Data*, in READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE 444 (W. Clancey & E. Shortliffe eds. 1984) (describing an expert medical system designed using EMYCIN).

187. EMYCIN stands for "Essential MYCIN" and is a distillation of MYCIN, an expert system capable of diagnosing infections.

188. See *Languages and Tools*, *supra* note 185, at 287-91; van Melle, Shortliffe & Buchanan, *EMYCIN: A Knowledge Engineer's Tool for Constructing Rule-Based Expert Systems*, in RULE-BASED EXPERT SYSTEMS 302 (B. Buchanan & E. Shortliffe eds. 1984).

asking the system user detailed questions about the user's plans and property. Although they are all specified in the traditional "IF-THEN" format, some of the production rules in TAXADVISOR are quite complex. For example, the rule governing whether the system will recommend that a short-term trust be utilized as an estate planning device is as follows:

- IF:
- 1) The client and/or spouse does wish to shift property income to another (not for support), etc. for at least ten years or until the death of the beneficiary,
 - 2) The client and/or spouse does desire to eventually reclaim control of this property (for retirement, estate liquidity, etc.),
 - 3) The client and/or spouse are in a higher income tax bracket than the beneficiary,
 - 4) The client and/or spouse is willing to relinquish control of the beneficial enjoyment of the property,
 - 5) The client and/or spouse are able to provide for their living needs without this income, even in the event of disability or unemployment,
 - 6) The client and/or spouse does not plan to have the trust income used to pay life insurance premiums on his/her life without the consent of an adverse party,
 - 7) The client and/or spouse does not plan to use the trust for a lease back of assets, *and*
 - 8) A: The client and/or spouse have a person (e.g., a parent) they are supporting without legal obligation with this property income (will lose a dependent if trust is formed),
B: The client and/or spouse have a child, not a minor, that they will be putting through college with this property income (can set up early and accumulate income without tax problems), or
C: The client and/or spouse are using some of their after-tax income for the benefit of some other taxpayer (child's marriage and/or home purchase, etc.),
- THEN: It is definite (1.0) that client should transfer assets to a short-term trust.¹⁸⁹

Information concerning the predicate portions of TAXADVISOR's "IF-THEN" rules is elicited from system users through a series of questions. If a user does not understand the need for or significance of information sought, she can type "WHY" as a response to a question and receive an explanation. That explanation not only describes the "IF-THEN" production rule being applied, but also recaps the information already gathered about the predicates of that rule.

189. Michaelsen, *supra* note 6, at 156.

To validate the system, advice given by TAXADVISOR was compared with the advice of several certified public accountants ("CPAs").¹⁹⁰ TAXADVISOR and the CPAs were given the same client information and the advice given by each was submitted to additional CPAs for judging without identifying which advice came from which source.¹⁹¹ Although the recommendations reached by TAXADVISOR and its human counterparts were not identical, the judges assessing these recommendations found them equally acceptable.¹⁹²

TAXADVISOR serves to illustrate some of the advantages of using an expert system shell as a programming tool. Its developer devoted his primary attention to defining and revising detailed "IF-THEN" rules in eight major estate planning areas, and consequently the result is a system that has substantive breadth comparable to that of practicing CPAs. Once these rules were formulated, other key operating features of the system were controlled by the EMYCIN shell. Thus, no further programming was necessary to implement TAXADVISOR's capacity to ask questions of users in ordinary English (including explanations of terms where necessary), to explain why it is seeking a given piece of information, or to recommend estate planning actions.

The use of the EMYCIN shell also led to some problems, however. Early in the development of TAXADVISOR, users had difficulties completing analyses within a reasonable time. Investigations revealed that TAXADVISOR was undertaking too many analyses of tax planning options that were irrelevant to the particular client whose plan was being developed.¹⁹³ In part, this stemmed from blind reliance on the EMYCIN shell and its comprehensive scrutiny of system rules. Later versions of TAXADVISOR were designed with "IF-THEN" rules that helped speed system operations by determining at an early stage that a given tax planning option (e.g., *inter vivos* gifts) was not of interest and should therefore be dropped from consideration. Even so, complete TAXADVISOR consultations generally took longer than 30 minutes.

190. The CPAs involved in this study each had at least two years full-time experience in estate planning. *See id.* at 164.

191. *Id.*

192. The CPAs serving as judges of the tax planning recommendations made the following assessments:

Source of Recommendations	Recommendations Rated Equivalent To Judge's Own	Recommendations Rated Acceptable	Recommendation Rated Unacceptable
Taxadvisor	6 (55%)	5 (45%)	0 (0%)
Human Analysts	10 (55%)	5 (33%)	0 (0%)

See id. at 165.

193. *Id.* at 163-64.

Another possible disadvantage of TAXADVISOR is that its consultation sessions do not produce final tax planning advice, but instead result in lists of planning devices that may be useful to a client. The specific combination of devices that will maximize the wealth the client transfers at death and the financial feasibility of that combination must be determined through calculations performed outside the system.

2. CHOOSE¹⁹⁴

CHOOSE, another computer program developed by Professor Hellawell, is designed to assist attorneys in answering a narrow legal tax planning question: whether federal tax considerations favor a branch office or a foreign subsidiary as a means for conducting a foreign mining operation.¹⁹⁵ The program was designed for use by lawyers knowledgeable in this area of the law and thus its capability for giving textual explanations of applicable law is very limited. Its aim was instead to save tax specialists' time and to insure accuracy in making a set of highly complex computations.

As a basis for its computations, CHOOSE elicits a variety of information from the user, ranging from data on the income and deductions of the firm involved to corporate tax rates in both the United States and the foreign jurisdiction where the mining will occur. Based on this information, CHOOSE evaluates the merits of branch and subsidiary operations by computing the relative tax advantages of depletion deductions (available to branches of United States concerns) over tax deferrals (available through the use of a subsidiary).¹⁹⁶ The following example drawn from a typical CHOOSE consultation session illustrates the types of factual information the system collects, retains, and manipulates in completing its analyses. For purposes of clarity, user responses are italicized; all other text is generated by CHOOSE.¹⁹⁷

194. Hellawell, *supra* note 9.

195. The choice between a having a branch office and a foreign subsidiary for foreign mining operations involves a trade-off between the percentage depletion deductions available through branch operations and the total exclusion from United States taxation of foreign source earnings by a foreign subsidiary. If a branch office is used, the domestic firm that creates it will include the branch's net income in the firm's taxable income, but will be able to utilize depletion deductions arising out of the branch's activities to offset that income. See I.R.C. § 61(a) (1954 and West Supp. 1985); Treas. Reg. § 1.11-1(a) (1960). If a subsidiary is used, the foreign subsidiary will pay income tax in its home country, but only earnings paid to its domestic parent will be subject to United States taxation. However, the benefit of potential depletion deductions will be lost. Hellawell, *supra* note 192, at 340-42.

196. See *id.* at 342-45.

197. *Id.* at 347-48.

THIS PROGRAM WILL HELP YOU DECIDE THE FORM TO USE FOR A FOREIGN DIRECT INVESTMENT. WHEN YOU PROVIDE IT WITH DATA ABOUT THE INVESTMENT, IT WILL COMPARE THE TAX CONSEQUENCES OF A BRANCH TO THOSE OF A FOREIGN SUBSIDIARY. THE PROGRAM IS PARTICULARLY ADAPTED TO MINING INVESTMENTS.

WOULD YOU LIKE TO SEE THE GENERAL DIRECTIONS? ANSWER YES OR NO. THEN PRESS CARRIAGE RETURN.

? N

THE CALCULATIONS MAY COVER UP TO TWENTY YEARS. FIRST, STATE THE STARTING YEAR (E.G. 1 OR 1980; DO NOT NUMBER A YEAR ZERO). THEN STATE THE ENDING YEAR.

STARTING YEAR—? 1982

ENDING YEAR—? 1986

FIRST I WILL ASK FOR INFORMATION RELATING TO FOREIGN TAXES.

STATE THE ENTERPRISE GROSS INCOME, COMPUTED UNDER THE FOREIGN TAX RULES, FOR THE FOLLOWING YEARS—1982?

1200000

IF ANSWER FOR ANY YEAR AND FOR ALL REMAINING YEARS IS THE SAME AS FOR THE PRECEDING YEAR TYPE 1.

IF ANSWER FOR ANY YEAR IS THE SAME AS FOR THE PRECEDING YEAR TYPE 2.

1983 ? 1700000

1984 ? 2200000

1985 ? 1

FOR A SUBSIDIARY: STATE THE FOREIGN CORPORATE INCOME TAX RATE AS A DECIMAL (E.G. .45 OR .36) FOR THE FOLLOWING YEARS —

1982 ? .32

Like CORPTAX, CHOOSE is somewhat inflexible and hard to understand because its expertise is embodied in computer programming statements instead of in a separate knowledge base. Furthermore, even within the narrow area of expertise it was designed to address, CHOOSE does not cover all factual circumstances that may arise.¹⁹⁸ However, the excluded circumstances are rare and the user is warned in advance that the system will not address them.¹⁹⁹

198. For example, CHOOSE assumes that: (1) dividends will never exceed the earnings of the current year; and (2) no income from the proposed mining project will be a capital gain. *See id.* at 365.

199. *See id.* at 347-48 n.19, 356.

While it may not have all the flexibility or explanatory characteristics of more sophisticated expert systems, CHOOSE does share one important feature of most successful systems—it relies upon and utilizes the computer's special analytic strengths. The tax problem addressed by CHOOSE requires extensive numerical analyses. In planning situations, these numerical analyses must be repeated many times in order to assess the tax implications of different circumstances. In addition, relatively large amounts of numerical data must be retained and made immediately available when needed in the analyses. Computers are highly efficient at performing these computational and data handling tasks.²⁰⁰ Consequently, expert systems like CHOOSE can exceed in both speed and accuracy the capabilities of human analysts acting alone.

It is also important to emphasize the extent to which CHOOSE was designed to interact with users.²⁰¹ For example, it permits a user to select the level of detail it uses in reporting the results of its calculations. More significantly, it permits portions of the data about a mining situation to be changed easily for repeat analyses, thus facilitating the use of the program in planning situations where the impact of modifying an action often needs to be assessed.

3. *A Bankruptcy Planner*²⁰²

An expert system developed by Professor William Boyd of the University of Arizona (and as yet unnamed) assists attorneys in answering two bankruptcy planning questions: (1) whether a client should file bankruptcy; and (2) if so, whether the client should seek a "liquidation" bankruptcy under Chapter 7 of the federal Bankruptcy Act²⁰³ or a "rehabilitation" bankruptcy under Chapter 13.²⁰⁴ Boyd devised the expert system by first examining how he would respond to a complex bankruptcy planning problem. He then captured the legal and practical inference rules he considered and grouped them in sets to be systematically applied by the computer system.²⁰⁵

200. The business question that CHOOSE addresses involves rather complex numerical analyses of the interplay between two types of tax calculations. If different structuring approaches are to be analyzed, these two sets of calculations must be repeated again and again. CHOOSE was designed to undertake the burden of these calculations not only to insure their accuracy, but to allow lawyers to test the effects of a wider variety of facts in assessing alternate forms for foreign mining operations. *See id.* at 342.

201. *See id.* at 345-46.

202. Boyd, *supra* note 5, at 699.

203. Bankruptcy Reform Act of 1978, 11 U.S.C. §§ 701-766 (1982).

204. *Id.* at §§ 1301-1330.

205. Boyd, *supra* note 5, at 702.

Boyd's system begins a consultation session with an attorney-user by gathering information about the attorney's client and the client's spouse, such as their personal debts, continuing expenses, and income. Utilizing this information, it informs the user whether bankruptcy is advisable, and, if so, which type would be preferable. In determining whether to recommend bankruptcy at all, Boyd's system takes into account the client's overall financial circumstances, with particular emphasis on whether creditors are taking or threatening action which will cause the client to lose essential property and also whether the client is in a position to resolve the problem without resorting to the bankruptcy alternative. Since much of the information the system gathers at this stage is the same information it will need for determining which type of bankruptcy proceeding will be best for the client (if one is initiated), the system retains all necessary information in order to immediately provide further advice on the choice between Chapter 7 and Chapter 13 bankruptcies.²⁰⁶

In general, a Chapter 13 bankruptcy is more desirable than its Chapter 7 counterpart, since the debtor can usually retain more of his or her assets.²⁰⁷ Thus, Boyd's expert system is directed primarily towards determining whether a client is eligible for a Chapter 13 proceeding and, if so, whether a Chapter 13 rehabilitation plan is feasible. The system is capable of examining a variety of eligibility criteria, depending on the nature of the client. Its feasibility assessments turn on a careful assessment of the client's monthly net income and expenses, as well as the possible advantages of liquidating certain assets where a plan would otherwise not be feasible. In addition, the system provides summary reports of important computational information it has gathered, such as descriptions of the debtor's assets that are subject to security interests and the surplus income the debtor is likely to have in excess of his continuing expenses.

A portion of a typical session conducted by Boyd's system will illustrate the type of questions asked by the system and the sorts of information gathered. The following session segment concerns one of several debts of a hypothetical couple named "JOHN" and "JANE," with user responses italicized:

206. *Id.* at 705.

207. For example, assets subject to valid security interests, which would be lost in a Chapter 7 liquidation, can be kept in a Chapter 13 proceeding. *See id.*

We need complete information as to each personal debt. Do not include here 'expenses' (i.e., obligations as to which there are no outstanding balances). Expenses are considered below. Do not include alimony and support payments. Do include arrearages as to such payments.

Is the debt the result of a sale(s), loan(l), or other(o)?

(Enter an 's,' an 'l,' or an 'o.')

Enter the creditor's name.

? *SAM KRAVITZ, INC.*

We need the creditor's complete address. Enter the creditor's street address.

? *200 S. PALO VERDE*

Enter the creditor's city and state.

? *TUCSON, AZ*

Enter the creditor's zip code.

? *85722*

What is the balance owing? Enter without a '\$' or commas.

? *2200*

Is the debt payable in installments?

(answer y or n)? *y*

Enter the monthly payment without a '\$' or commas.

? *105*

Are the payments current?

(answer y or n)? *y*

Is the amount in dispute?

(answer y or n)? *n*

Is the debt cosigned?

(answer y or n)? *n*

Is the debt secured?

(answer y or n)? *y*

How many items of collateral are there?

? *2*

Describe the collateral.

? *Furniture*

Enter the fmV of the collateral without a '\$' or commas.

? *1500*

Is the security interest purchase money?

(answer y or n)? *y*

Is the security interest avoidable? See BRA 522(f), 545, and 547.

(answer y or n)? n²⁰⁸

Written in BASIC, Boyd's system resembles Professor Hellawell's CORPTAX and CHOOSE systems in that it uses algorithms embedded in the system's programming to control both the gathering of necessary facts and the analysis of those facts.²⁰⁹ As with CORPTAX and CHOOSE, this embedding of analytic and information-gathering algorithms directly in computer programming statements tends to conceal the system's logic from users, thereby lowering its perceived trustworthiness and frequency of use. Apparently recognizing some of these limitations, Boyd has proposed a revised system that will contain explicit inference rules and will be written in a dialect of LISP,²¹⁰ a computer programming language well suited to expert system construction.²¹¹

C. Prediction

Prediction systems infer the likely consequences arising out of particular factual circumstances. Often predictions are based on models of real-world activity built into a system's knowledge base.²¹² The system can then simulate real processes to generate scenarios that stem from particular circumstances specified by the system user.²¹³ Although no expert legal system is as yet capable of prediction in this sense, the basis for such a system exists in the Legal Decisionmaking System ("LDS").²¹⁴

208. *Id.* at 723-24.

209. *See id.* at 761 n.68.

210. *See id.* at 761 n.61.

211. *See supra* notes 23-24.

212. *See D. WATERMAN, supra* note 1, at 35; *Basic Concepts, supra* note 29, at 84.

213. For example, the LDS system incorporates a model of the process through which personal injury attorneys and insurance claims adjusters value personal injury claims. This model describes the settlement value of a particular claim in terms of the following formula:

$$\begin{aligned} \text{VALUE} = & \text{LOSS} * \text{LIABILITY} * \text{RESPONSIBILITY} \\ & * \text{CHARACTERISTICS} * \text{CONTEXT} \end{aligned}$$

where:

LOSS = Dollar amount of plaintiff's loss;

LIABILITY = Probability of establishing liability;

RESPONSIBILITY = Percentage responsibility of defendant in comparative negligence jurisdictions;

CHARACTERISTICS = Adjustment for superficial aspects of the claim, such as characteristics of litigants, lawyers, and judges;

CONTEXT = Adjustments for matters of strategy, timing, or type of claim (e.g., auto accident or product defect).

Using this model and a variety of mechanisms for assigning figures to these variables, LDS analyzes a claim and predicts its settlement value. *See Evaluating Product Liability Cases, supra* note 13, at 627-28.

214. *See Evaluating Civil Claims, supra* note 13, at 65-76; *Evaluating Product Liability Cases, supra* note 13, at 627; *Legal Decision Systems, supra* note 13, at 217-18.

LDS is an expert legal system that emulates settlement analyses undertaken by litigators and insurance company claim adjusters in products liability cases. Created by D.A. Waterman and Mark A. Peterson of the Rand Corporation, LDS was developed to explore the feasibility and usefulness of computer prediction models in making legal decisions. In considering whether a meaningful computer model of settlement decisions could be formulated, Waterman and Peterson addressed such novel issues as how the system should interpret imprecise legal standards, what techniques would probably include uncertainty in analyses, and how a system based on a prediction model should explain its analyses.²¹⁵

LDS is a rule-based system, meaning that it is implemented solely in terms of "IF-THEN" rules connecting premises to expert conclusions. It is written in ROSIE, a knowledge representation language designed to facilitate the development of large expert systems using English-like syntax.²¹⁶ The "IF-THEN" rules underlying LDS form chains that represent the inference processes involved in settlement analyses. The conclusion of one rule may be an intermediate result which is one of the premises of another rule.²¹⁷ Similarly, the conclusion of that second rule may correspond to the premise of yet a third.²¹⁸ Using ROSIE, the authors of LDS have estimated that several thousand rules would be necessary to implement a system to *fully* analyze product liability settlements, even where the settlements are restricted to a narrow factual setting such as traffic accidents.²¹⁹

The facts of each problem presented to LDS determine which "IF-THEN" rules will be invoked and, consequently, the inference path that LDS will follow, with each of the "IF-THEN" rules in LDS containing a complicated set of premises. For example, the "IF-THEN" rule in LDS testing for the presence of strict liability includes the following:

215. See *Models of Legal Decisionmaking*, *supra* note 13, at 13-27.

216. ROSIE is an example of a specialized language designed primarily to facilitate the recording of information in knowledge bases. Because ROSIE incorporates fragments of English in its syntax, it makes it easier for factual assertions to be created, manipulated, and accessed. For a description of ROSIE and other related knowledge representation languages used in expert system development, see Barstow, *supra* note 183, at 301-26; D. WATERMAN, *supra* note 1, at 118-23. For an illustration of the ways ROSIE can be used to construct an expert legal system, see *Legal Decision Systems*, *supra* note 13, at 216-17.

217. *Evaluating Product Liability Cases*, *supra* note 13, at 630-32.

218. *Id.*

219. *Models of Legal Decisionmaking*, *supra* note 13, at 38.

IF (the plaintiff is injured by the product)
 or (the plaintiff does represent the decedent and the decedent is killed by the product)
 or (the plaintiff's property is damaged by the product)
 and the incidental-sale defense is not applicable
 and (the product is manufactured by the defendant
 or the product is sold by the defendant
 or the product is leased by the defendant)
 and the defendant is responsible for the use of the product
 and (California is the jurisdiction of the case
 or the user of the product is the victim
 or the purchaser of the product is the victim)
 and the product is defective at the time of the sale
 and (the product is unchanged from the manufacture to the sale
 or (the defendant's expectation is 'the product is unchanged from the manufacture to the sale'
 and the defendant's expectation is reasonable and proper))

THEN assert the theory of strict-liability does apply to the plaintiff's loss.²²⁰

Waterman and Peterson utilized LDS primarily as a research tool in studying how settlement decisions were made, finding the format of rule-based computer models particularly advantageous for this purpose.²²¹ To develop a rule-based model for their system, Waterman and Peterson were required to catalog actual bases for settlement decisions, organize the full range of factors that might influence litigators' decisions in a coherent manner, and record a baseline of decision standards and strategies for use in studying the impact of changes in the law or negotiating techniques on settlement decisions. Each of these byproducts of the development of LDS led to significant research about how settlement decisions are made.

Although it was primarily designed as a research tool for the authors and not an expert system for attorneys,²²² LDS closely resembles

220. *Id.* at 38.

221. *See id.* at 35-36.

222. In part because of the lack of understanding of the criteria attorneys consider in settling cases, LDS was created as a means to systematically identify those criteria as a precursor to the creation of expert systems that would apply the identified criteria to assist litigators in predicting settlement values. *See Evaluating Product Liability Cases, supra*

an expert system capable of predicting successful settlement offers. The rule structure of LDS, if fully refined, would be able to simulate expert interpretations of the strength and value of product liability claims, and hence could hypothesize about the size of an offer which would be necessary to settle a given case. Such predictions would be useful to settlement negotiators in structuring the starting point and sequence of their offers.²²³ Furthermore, LDS is capable of extensive explanations of its actions, including the ability to describe which of its "IF-THEN" rules led to a given deduction and why those rules were applicable.

Even with these sophisticated features, LDS has limitations as an expert system. Although the developers recognized that a litigator's analysis of whether to settle a particular case would turn on broad considerations, LDS only attempts to include legal rules governing strict products liability and rules concerning the calculation of damages.²²⁴ Other factors which would be considered in an attorney's settlement analysis, but which are not incorporated in LDS, include rules concerning the interpretation of indefinite legal standards (e.g., general principles for determining whether or not an act was reasonable or undertaken with due care)²²⁵ and rules that reflect attorneys' expectations about likely jury or judicial responses to the facts or parties involved in a particular case (e.g., predictions of jury reactions to sensational injuries or sympathetic qualities in the parties).²²⁶ Thus, the range of analysis of LDS is significantly limited.

D. Design

A number of expert systems carry out design processes in which specifications are developed for an object or item that must satisfy certain functional requirements.²²⁷ Existing expert legal systems assist attorneys in design tasks ranging from the drafting of complex legal documents to the specification of criminal sentences.

note 13, at 627-28.

223. See *Models of Legal Decisionmaking*, *supra* note 13, at 24.

224. *Id.* at 14-16.

225. See *id.* at 4, 15. Waterman and Peterson have implemented explanatory features in a more recent expert system concerning asbestos exposure claims that would help attorney-users interpret unclear legal standards. See *Legal Decision Systems*, *supra* note 13, at 218-23.

226. *Models of Legal Decisionmaking*, *supra* note 13, at 4, 15. For a discussion of recent developments regarding the incorporation of such litigation characteristics into LDS, see *Legal Decision Systems*, *supra* note 13, at 217-18.

227. See D. WATERMAN, *supra* note 1, at 35; *Basic Concepts*, *supra* note 29, at 66-72.

1. *ABF Processor*²²⁸

The ABF Processor is an expert legal system with a narrow, but important, design function. Developed by James A. Sprowl of the American Bar Foundation, the ABF Processor creates what might be termed "smart documents"—that is, special versions of form documents like wills, trust agreements, and probate filings which are assembled through computer analyses of factual information about particular clients.²²⁹

The ABF Processor operates in two phases. First, a form document is typed into the system, with special variables or "space holders" placed in locations where variable information, such as a client name or alternative clauses, may be added. At this stage, the developer of one of the system's smart documents can specify the conditions under which particular text will be included in the document.

For example, if certain clauses are to be included in a will only if the decedent's estate may be probated without a court appearance, the creator of an ABF Processor document would type in the appropriate optional text preceded by the following command:

[IF the death taxes ARE to be paid out of the res of the estate INSERT].²³⁰

Another portion of the ABF Processor transforms this "IF" statement into the following question which will be presented to the system user each time the automated will is prepared: "[a]re the death taxes to be paid out of the res of the estate?" If the document developer wished to specify criteria for determining when death taxes will be paid out of the res of the estate, he could do so simply by inputting the criteria in "IF-THEN" form, and this new information would in turn be transformed into a corresponding set of questions to be asked of users of the system.²³¹

In the second phase of its operation, the ABF Processor assembles a document using forms created in the first phase. Often, the user of the system may not be the party who created the automated document, but rather another attorney or office assistant who merely answers the factual questions posed by the system.²³² Based on information about a particular client, the system assembles a version of the form document

228. Sprowl & Staudt, *supra* note 12, at 699, 701; Sprowl, *supra* note 12, at 1. A partial implementation of the ABF Processor for use in familiarizing law students with the capabilities of such systems is described in P. MAGGS & J. SPROWL, *COMPUTER APPLICATIONS IN THE LAW* 68-103 (1987).

229. For a brief description of the origins and history of the ABF Processor, see *id.* at 10-17.

230. *Id.* at 23-24.

231. *Id.* at 24.

232. *Id.*

tailored to that client's needs. In this way, the system applies the expertise of the document's originator regarding the suitability of special document terms to frequently encountered circumstances. A document generated in this way will still need to be scrutinized by an attorney to accommodate unusual client problems or to address subject matter outside of the system's expertise.

Although it is limited to document preparation and does not provide expert interpretations in a traditional sense, the ABF Processor does seem to perform document preparation tasks very thoroughly. It has been tested by law students and attorneys in a variety of legal contexts. In one test, a group of law students was able to create a system in twelve hours that automated the completion of income tax form 1040A.²³³ Larger development efforts have produced much more elaborate preparation systems for divorce, will, and trust documents.²³⁴

The ABF Processor does have several important limitations, however. First, completion of the initial stage of document preparation—in which alternative clauses and the conditions for their inclusion are specified—requires the document developer to understand a complex, specialized command language used only by the ABF Processor.²³⁵ Second, when it assembles documents, the ABF Processor asks questions in a manner determined by the order in which optional clauses arise in the document being prepared.²³⁶ This order may often not be the most logical or efficient sequence for gathering information.²³⁷

2. A-9

A-9 is another program that assists attorneys in preparing specialized legal documents. This system produces security agreements complying with Article 9 of the Uniform Commercial Code.²³⁸ A-9 is capable of creating security agreements dealing with many different types of collateral, including any combination of consumer goods, equipment, and fixtures. It also accommodates transactions in which shares of stock are used as additional collateral.²³⁹ Professors William E. Boyd of the

233. Sprowl, *supra* note 12, at 60-61.

234. For descriptions of field tests of the ABF Processor in a variety of practice settings, see Sprowl & Staudt, *supra* note 12, at 723-27.

235. See generally Sprowl, *supra* note 12, at 17-50.

236. See Sprowl, *supra* note 12, at 23-25.

237. See Boyd & Saxon, *supra* note 120, at 644, 645 (document preparation system in which all factually related questions are asked in one session or interval and responses are used to determine proper terms in several documents).

238. See U.C.C. § 9-203 (1978). A-9 can also complete financing statements corresponding to each security agreement that it generates. See U.C.C. §§ 9-401, 9-402 (1978).

239. Boyd & Saxon, *supra* note 120, at 652.

University of Arizona and Charles S. Saxon of Eastern Michigan University created A-9 to assess the feasibility of implementing a useful legal document drafting system on a relatively inexpensive microcomputer.²⁴⁰ They hoped such a system would be affordable by a wide range of attorneys, including sole practitioners.

A-9 operates much like the ABF Processor. "Smart" security agreements are created by A-9 through the use of a special document preparation language developed by Legal Management Systems, Inc. ("LMS"), a private vendor of law office computer systems.²⁴¹ Where the text in these documents will vary depending on the needs of a client, special symbols are entered to indicate where substitutions or changes should occur.²⁴² The material to be added when the document is ultimately assembled can be entered by the user of the A-9 program either as a response to a question or can be derived from other responses.²⁴³ In addition, optional clauses expanding the basic security document can be included or excluded depending on the answers given to questions posed to a system user.²⁴⁴

A-9 offers some advantages over the ABF Processor. Unlike the ABF Processor — which only generates a completed document at the termination of its processing — A-9 allows a user to see the security agreement as it is being assembled. In addition, the LMS software which forms the basis for A-9 operates on smaller, less expensive computer systems than the original ABF Processor.²⁴⁵ Finally, A-9 requires more active user involvement in the assembly of documents than the ABF Processor or other systems which only allow the user to fill in blanks or choose stock terms. The authors of A-9 claim this involvement is necessary to properly balance drafting art and text automation in the preparation of sophisticated security agreements.

Yet, A-9 also has two significant drawbacks. As with the ABF Processor, persons preparing new versions of security agreements incorporated within A-9 will need to understand the special programming language used in that system. Furthermore, at least in its initial version,

240. The system on which A-9 was implemented was available in 1981 for under \$20,000. See Boyd & Saxon, *supra* note 120, at 640 n.9. A comparable system would be considerably less expensive today.

241. See *id.* at 645.

242. *Id.* at 646-47.

243. *Id.*

244. *Id.*

245. Later versions of the ABF Processor have also been implemented on relatively small computers. See Sprowl & Staudt, *supra* note 12, at 730 n.66. A partial implementation of the ABF Processor has recently been distributed for use with IBM-PC microcomputers. See P. MAGGS & J. SPROWL, *COMPUTER APPLICATIONS IN THE LAW* 68-103 (1987).

A-9 is aimed at a much more specialized type of document drafting, i.e., security agreements, than the general purpose ABF Processor system. The degree of additional programming necessary to transform A-9 into a general purpose document processor is unclear, but it may make a microcomputer implementation of the generalized system unwieldy.

3. SENPRO²⁴⁶

SENPRO is another expert system that assists in a specialized legal design task—the formulation of criminal sentences. It was designed as a prototype for a computer system capable of assisting experts with decision making in a variety of legal, social, and economic areas, with criminal sentencing merely an initial test application. Created by Richard V. de Mulder and Helen M. Gubby of Erasmus University in Rotterdam, Germany, SENPRO incorporates a model of ideal sentencing considerations developed at Erasmus University. This model consists of numerous rules concerning the goals underlying criminal sanctions and the means necessary to achieve those goals. Consideration is also given to the need to look outside the criminal justice system in order to resolve certain types of cases. Overall, the model defines a rationalized sentencing pattern, referred to by the system developers as a “tariff.”²⁴⁷

Information about the sentencing decision in a particular case is solicited from system users through a series of multiple choice questions. The following portion of a SENPRO consultation session reflects typical questions asked of system users (user responses are italicized):

We will now start the actual gathering of information in order to fix a tariff of the (category of) case(s) you have in mind.

PRESS RETURN TO CONTINUE?

CASE: Paul de Vries

Case: drunken driving

Informant: Richard de Mulder

LEVEL OF INDIVIDUALIZATION

The fact that sentencing decisions are made within an organization whose capacity is limited has certain consequences. Certain groups of decisions have to be made on a strongly categorized basis. The more that a punishment is individualized the heavier that the burden is of collecting and processing information. Therefore, one of the principal sentencing decisions that has to be made is what level of individualization is applicable to the sentencing process.

246. DeMulder & Gubby, *supra* note 14, at 243.

247. *See id.* at 254.

A number of questions follow, the aim of which is to fix the level of individualization in the case presently under consideration.

FIRST CRITERION FOR THE LEVEL OF INDIVIDUALIZATION

The first criterion to be dealt with concerns the processing possibilities of the system.

Will you now indicate whether or not the case with which you are concerned is the type of case which occurs frequently (e.g., minor traffic offenses)?

1. The case occurs frequently.
2. The case occurs infrequently.
3. The case occurs neither frequently nor infrequently.

THE MOST SUITABLE ALTERNATIVE IS: ? 3

...

SECOND CRITERION FOR THE LEVEL OF INDIVIDUALIZATION

The second criterion concerns the question of to what extent an intrusion of privacy would be justified in order to obtain the information necessary for a highly individualized decision.

How far is such an intrusion justified?

1. In this case it is certainly justified.
2. In this case it is only justified to a certain extent.
3. In this case it is only justifiable at a low level.
4. In this case it is not justified.

THE MOST SUITABLE ALTERNATIVE IS: ? 3

...

The extent to which an offense is regarded as "serious" should determine an upper limit to the intrusion of privacy and any other consequences of an inquiry into the case.

In this case a fairly low level of individualization is indicated.

The final sentencing decision will be of a generalized character; the following processing of information and the decision making will concern the "type" of case that is now submitted for a decision.²⁴⁸

Once basic information is obtained in this manner, sentencing goals and means to attain them are identified and the system suggests a tentative sentencing conclusion. The system then enters a second round of assessment in which the user gathers further information in order to refine the factual picture already obtained. If further factual information

248. *Id.* at 264-66.

is available, the system will repeat, correct, or supplement its earlier analyses as necessary.²⁴⁹

As the SENPRO analysis progresses, the user also has an opportunity to make several collateral decisions regarding sentencing options. For example, following the second assessment round, the user is presented with the system's sentencing recommendations in several categories, including fines, imprisonment, mental institutionalization, and forfeiture.²⁵⁰ The user must then select which type of sanction is appropriate. Similarly, at a later stage in the system's operation, a cost-benefit analysis is initiated to determine if the means selected are suitable for realizing the aims of punishment in the case under scrutiny.²⁵¹ The system also assesses whether the means chosen for achieving one of the selected sentencing goals conflict with the means chosen to achieve another goal. At the conclusion of its analysis, SENPRO describes its conclusions and their bases. This description includes both a summary of the factual information considered by the system and the sentencing goals applied.

SENPRO is written in two parts: a program component which insures that sentencing considerations are addressed in the proper order, and a decision file which includes a list of information about sentencing considerations and their implications. With this split function, the decision file can be altered or updated without need for changes in the processing provisions of the program component.²⁵²

As with CORPTAX, CHOOSE and William Boyd's bankruptcy planning system, SENPRO is somewhat inflexible because it is written in the general purpose BASIC programming language.²⁵³ Its expertise is necessarily masked by the complexity of BASIC to non-computer specialists.²⁵⁴ Furthermore, although the separation of decision information from the other aspects of the program may facilitate easy change in decision information within the system, the unusual knowledge representation scheme used in SENPRO—stated in a peculiar combination of BASIC programming and a special representation format developed just for SENPRO—makes the system's knowledge base uninterpretable by most lay users and, hence, unmodifiable by them.

249. For a description of the sentencing model, see *id.* at 254-57.

250. See *id.* at 259.

251. See *id.* at 259-60.

252. See *id.* at 246-47.

253. See *id.* at 245, 298.

254. See *id.* at 247-54.

E. Control Systems: The Legal Research System²⁵⁵

Expert systems that carry out a control function adaptively govern the completion of a complex task.²⁵⁶ For example, a computer system designed to monitor the progress of product assembly in an automated manufacturing plant and to adjust assembly activities accordingly would be a "control" system. In the legal field, a system which monitored a series of court filings in a particular case and organizing them in terms of an overall litigation plan would be an example of an expert legal system used for control purposes.²⁵⁷

The Legal Research System ("LRS") is an existing expert legal system that undertakes an important control activity—namely, the organization and maintenance of information on cases and statutes in a way that facilitates retrieval of such information by attorneys and other legal researchers.²⁵⁸ Developed by Carole D. Hafner as a doctoral research project at the University of Michigan, LRS was created to study the types of information a useful legal research system would need to include and the ways that information could be structured to best facilitate information retrieval.²⁵⁹ Expertise organized through LRS includes information on statutes and cases concerning negotiable instruments. Specifically, the system captures data on all of the statutes contained in articles 3 and 4 of the Uniform Commercial Code ("U.C.C."),²⁶⁰ and on all of the cases cited in the section on negotiable instruments in White and Summers' influential hornbook on the U.C.C.²⁶¹ Overall, the LRS database contains knowledge about approximately 200 statutory provisions and 200 related cases.²⁶²

LRS specifies information on particular statutory provisions and cases in terms of the following eight criteria ("D" in each instance refers to a descriptive phrase):

(PLAINTIFF D)—The plaintiff of a case was a "D." D must describe a party—for example, the payee of a check. (DEFENDANT D)—The defendant of a case was a "D."

(CAUSE-OF-ACTION D)—The legal basis of a case was "D"—for example, negligence.

255. Hafner, *Representation of Knowledge in a Legal Information Retrieval System*, in INFORMATION RETRIEVAL RESEARCH 139 (R. Oddy, S. Robertson, C. van Rijsbergen & P. Williams eds. 1981).

256. See D. WATERMAN, *supra* note 1, at 38.

257. See *id.* at 226.

258. See Hafner, *supra* note 255, at 139.

259. *Id.*

260. U.C.C. art. 3 & art. 4 (1978).

261. J. WHITE & R. SUMMERS, UNIFORM COMMERCIAL CODE 486-549 (1972).

262. Hafner, *supra* note 255, at 139.

(EXAMPLE D)—The fact situation of a case was an example of "D" — for example, a forged promissory note.

(HYPOTHETICAL EXAMPLE D)—A case or statute describes a hypothetical situation that is an example of "D."

(CRITERIA D)—A case or statute defines criteria for a situation to be an example of "D."

(LEGAL-EFFECT D)—A case or statute describes the legal consequences of "D."

(RULE D)—A case ruled that the situation before the court was an example of "D."²⁶³

These criteria were chosen to reflect typical objects of research interest on the part of attorneys and other legal researchers. In terms of the above criteria, a typical case would have the following description in LRS:

(PLAINTIFF (DRAWER CHECK))

(DEFENDANT (IS A (DRAWER CHECK) BANK))

(CAUSE-OF-ACTION C-IMPROPER-DEDUCTION)

(EXAMPLE IMPROPER DEDUCTION)

(EXAMPLE (DRAWER'S SIGNATURE FORGED))

(EXAMPLE (NOT EXAMINED-PROMPTLY))

(LEGAL-EFFECT (DRAWER'S SIGNATURE FORGED))

(LEGAL-EFFECT IMPROPER DEDUCTION)

(LEGAL-EFFECT (DRAWER (NEGLIGENT (CONTRIBUTING TO UNAUTHORIZED)))

(LEGAL-EFFECT (NOT REASONABLE-CARE-TAKEN))

(RULE C-IMPROPER-DEDUCTION)

(RULE ((NOT REASONABLE-CARE-TAKEN) (IS A (DRAWEE CHECK) BANK)))

(RULE (DRAWER ((NOT NEGLIGENT) (CONTRIBUTING TO (DRAWER'S SIGNATURE FORGED))))²⁶⁴

In English, this description corresponds to a suit by the drawer of a check against the drawee bank for improperly paying a check on which the drawer's signature was forged. Were the plaintiff to prevail, the bank would be required to re-credit the drawer's account for the amount of the check. The bank presented two arguments in its defense: first, that the plaintiff had been negligent in a way that contributed to the forgery and, second, that the plaintiff had failed to examine his monthly

263. *Id.* at 141.

264. Hafner, *supra* note 255, at 142. The case described is Jackson v. First National Bank of Memphis, 403 S.W.2d 109 (1966).

statement promptly and had consequently delayed in notifying the bank of the forgery beyond the time in which the bank could recover the funds paid. The court held that the plaintiff could recover the improper deduction, that the bank did not exercise reasonable care in handling the check and that the plaintiff was not negligent.

In order to retrieve information from LRS, a user would query the system with questions phrased in terms of the system's special terminology. For example, a query aimed at identifying cases involving a dishonored draft where the plaintiff was not a holder of the draft could be phrased as follows:

FIND CASES (EXAMPLE DISHONORED) AND NOT (PLAINTIFF HOLDER)²⁶⁵

In response to requests such as this, LRS has some ability to deduce that a request for information about subject D implies a request for information about related subject D'. In addition to information about statutes and cases which are contained in the manner described above, LRS contains semantic knowledge about the relationships between many of the legal concepts referred to in the system.²⁶⁶ More than 200 concepts were defined in terms of the following six basic concepts: PARTY, LEGAL-INSTRUMENT, LIABILITY, LEGAL-ACTION, ACCOUNT and AMOUNT-OF-MONEY.²⁶⁷

Clearly a research vehicle, LRS is nonetheless important as the first model of its kind for automating legal research. It demonstrates both a viable knowledge representation scheme for capturing and structuring data on legal source materials in a narrow legal field, and a rudimentary retrieval system capable of responding to carefully tailored inquiries about that data.

There are, however, obvious limitations to the employment of LRS as a useful retrieval system. First of all, it is not clear that the set of fundamental concept labels (e.g., "ISA (DRAWEE CHECK)") chosen by the creator of LRS's database are both unambiguous and comprehensively descriptive. Second, only a person well versed in the special definitions given these terms within LRS will be able to expand the system's database or formulate meaningful queries for information from the system. Finally, although it is capable of identifying cases or statutory sections relevant to particular negotiable instrument issues, LRS has no link to a computerized legal research system like WESTLAW or LEXIS²⁶⁸ that

265. *See id.* at 146-47.

266. *Id.* at 144-46.

267. *Id.* at 145.

268. For a survey of four CALRs currently in existence, see Soma & Stern, *A Survey of Computerized Information for Lawyers: LEXIS, JURIS, WESTLAW, and FLITE*, 9 RUTGERS COMPUTER & TECH. L. J. 295 (1983).

would permit automated retrieval of the actual text of those cases and statutory sections.²⁶⁹

CONCLUSION

Expert systems represent a fundamentally new way of transferring know-how from experienced experts to others. Such knowledge transfers have long been at the heart of legal practice. Consequently, as computer technology improves and its costs decline, new applications of expert legal systems are ripe for development.

Increased reliance upon artificial intelligence contained in programs like expert legal systems will not be without significant social implications.²⁷⁰ Some of the potential negative implications of increased reliance on such systems include the possibility of increased unemployment if human workers are displaced by corresponding expert systems,²⁷¹ the risk that users will tend to overestimate the capabilities of expert legal systems and therefore rely on the systems in situations beyond their range of expertise,²⁷² the possibility that persons relying heavily on analyses performed by computers will neglect the development of their own expertise,²⁷³ and, finally, the possibility that increased use of computer analyses will decrease humanism in the work of experts like lawyers because of the new ease with which they can perform analyses lacking full individual attention.²⁷⁴ While each of these problems is real, none is a

269. See Hafner, *supra* note 255, at 151.

270. For an overview of commentary to date on the social implications of increased use of artificial intelligence devices, see Partridge, *Social Implications of Artificial Intelligence* in *ARTIFICIAL INTELLIGENCE: PRINCIPLES AND APPLICATIONS* 315 (M. Yazdani ed. 1986).

271. See *id.* at 316-18; Nilsson, *Artificial Intelligence, Employment, and Income*, *AI MAGAZINE*, Summer 1984, at 5.

272. See Partridge, *supra* note 268, at 325-27.

273. See Dreyfus & Dreyfus, *supra* note 4, at 86, 90 (increased reliance on expert systems could "produce a generation of students and managers who have no faith in their own intuition and expertise").

274. See Partridge, *supra* note 268, at 319-22 (assessing the possibility that limited understanding of artificial intelligence programming by system users will force the consequences of incomplete computer analyses to be increasingly accepted by individuals with the explanation that "I'm sorry, but the computer doesn't allow us to do that").

For discussions of the contrary view that artificial intelligence devices do not threaten the dehumanization of society and may even promote humanism, see *id.* at 329-30 (describing applications of artificial intelligence computers that permit human processes conducted on a mass scale, such as educational procedures or manufacturing activities, to be customized to the needs of individual consumers); Boden, *Artificial Intelligence as a Humanizing Force* in 1983 PROC. EIGHTH INT'L JOINT CONF. ON ARTIFICIAL INTELLIGENCE 1197, 1197-98 (analysis of the humanizing impact of artificial intelligence devices on human occupations); Chance, *Intelligence, Artificial and Otherwise*, *AI MAGAZINE*, Winter 1985, at 22 (arguing that artificial intelligence devices need not be feared as a dehumanizing influence, but should rather be treated like any other human tool such as a hammer or a knife).

complete bar to the increased social benefits which can be gained from carefully constructed expert legal systems, used under constrained circumstances and in conjunction with related social programs (like retraining for displaced workers) that ameliorate negative externalities of expert legal system use.²⁷⁵

The reasons to pursue the development of these systems are compelling. Expert legal systems can preserve expert know-how, effectively distribute otherwise scarce expertise, reduce costs of mediocre or poor human analyses, and facilitate user access to information and computer capabilities.²⁷⁶ Additionally, efforts to develop expert legal systems promise improvements in the systematization and codification of existing legal knowledge.²⁷⁷ Finally, as regulatory schemes and the human endeavors they control become more and more complicated, reliance on the power of expert legal systems to retrieve and interpret complex legal standards may become a necessary means for attorneys to master and apply complex laws governing the equally complex technologies found in modern society.²⁷⁸ With this breadth of potential benefits, few areas of legal research are as exciting or promising as the creation of new expert legal systems.

275. See Partridge, *supra* note 268, at 319-22 (concluding that, although significant new social risks may be involved, "the potential benefits of artificial intelligence are . . . both dramatic and seemingly unlimited").

276. See D. WATERMAN, *supra* note 1, at 12-13; Hayes-Roth, *supra* note 2, at 28.

277. See *Overview of Expert Systems*, *supra* note 51, at 27-28; McCarty, *Reflections On TAXMAN*, *supra* note 8, at 839-41.

278. Cf. P. HARMON & D. KING, *supra* note 2, at 79-89 (describing the potential usefulness of expert systems in understanding and controlling various technological processes); D. MICHIE & R. JOHNSTON, *supra* note 4, at 9-11 (arguing that the development of computer systems based on artificial intelligence may be necessary for humans to maintain control over their increasingly complex technical environment); R. SCHANK, *supra* note 18, at 245-47 (projecting that artificial intelligence computers acting as servants, partners, and teachers will both help individuals deal with today's information overload and assist them in acquiring and interpreting new types of information).

Expert systems are already being used to insure that highly technical government activities conform to complex legal limitations. For example, the federal Environmental Protection Agency now uses an expert system to insure that its practices conform to both the information disclosure requirements of the Freedom of Information Act and the information confidentiality requirements of the Toxic Substances Control Act. See Feinstein, *A Knowledge-Based Expert System Used to Prevent the Disclosure of Sensitive Information at the United States Environmental Protection Agency*, in *COMPUTING POWER AND LEGAL REASONING* 661 (C. Walter ed. 1985); Feinstein & Siems, *EDAAS: An Expert System at the U.S. Environmental Protection Agency for Avoiding Disclosure of Confidential Business Information*, 2 *EXPERT SYSTEMS* 72 (1985).