1987

EXPERT SYSTEMS: Passing Fad or Lasting Contribution?*

William J. Van Beek and Jerald J. Fletcher**

Department of Agricultural Economics
Purdue University
West Lafayette, IN 47907

Information Services

Prepared for presentation at the 1987 AAEA Annual Meeting
August 2-5, 1987, Michigan State University

*This research was funded, in part, by the Mid-Size Family Farm Project, Indiana Agricultural Experiment Station and the National Fertilizer Development Center, Tennessee Valley Authority. The authors assume full responsibility for any errors or omissions.

**The authors are, respectively, Director of Information Services and Assistant Professor in the Department of Agricultural Economics, Purdue University.

EXPERT SYSTEMS: Passing Fad or Lasting Contribution?

Expert systems are being touted as one of the most promising recent developments in the area of computer-based decision support systems. Interest in expert systems has grown rapidly with the emerging availability of artificial intelligence based development techniques and tools. Considerable efforts are being expended to investigate whether these are viable tools for addressing problems related to agriculture.

This paper examines the characteristics of expert systems and compares and contrasts them with those of conventional programming. A historical view of expert system development is presented, with general examples of expert systems discussed as well as examples related to agricultural decision making. A discussion of the impacts (costs and benefits) that expert systems research and development may have on future agricultural management research is presented.

One such system, which uses human expertise to make nitrogen fertilizer rate recommendations for corn production in Indiana, is presented. The system analyzes a broad mix of information including objective soil and fertilizer data. It incorporates qualitative (subjective) information based on the proposed nitrogen management program and a variety of other management as well as physical parameters to arrive at the recommended rate. Many of the concepts used in this knowledge based decision support system can be used in programs to manage resources such as land, labor, and capital, and in programs to aid of the conservation of non-renewable resources.

What are Expert Systems?

Waterman describes an expert system as a computer program that captures the expertise of an expert in a specific problem domain and utilizes this knowledge to make useful inferences for the user of the system. A expert system is composed of three major subsystems: (1) the language system, (2) the knowledge system, and (3) the problem processing software (Bonczek, Holsapple, and Whinston, 1981). Its purpose is to offer advice or solutions for problems in a particular problem area -- advice comparable to that which would be offered by a human expert in a personal consultation.

The language system or user interface provides the means by which the decision maker interacts with an expert system and consists of two parts. The first part constitutes a retrieval language which, depending on the application and the sophistication of the system, can utilize user inputs that range from an explicit statement of how the data are to be retrieved to a general statement of the data desired. The second part is a computational language. Depending again on application and sophistication, directions may be as specific as to how computations are to be performed to a statement of the problem in terms of the requested data (Bonczek, Holsapple, and Whinston (1981)). The language system is characterized by the syntax it furnishes to the decision maker in the form of statements, commands, or expressions which utilize the retrieval and/or computational languages available. A user interface can be designed so that the user is unaware of whether the expert system is based on information retrieval, a modeling processes, or a combination of both. In the final analysis, the language system must be able to generate all acceptable problem statements.

Expertise consists of knowledge about a particular domain, and skill at utilizing the knowledge to solve problems. Human experts achieve success at solving these difficult problems because they are knowledgeable in many aspects of a problem and not confined by fixed, narrowly defined problem solving methods. Facts about a problem domain compose the knowledge system component of an expert system. Information maintained in a knowledge system must be retained in an organized, systematic manner to allow the problem processing software ready access to the full extent of the available knowledge related to the problem domain. Indeed, a good deal of the power of an expert system is derived from the knowledge base related to a particular problem domain. This knowledge typically includes large volumes of facts that the decision maker has neither the time, the inclination, nor the opportunity to absorb completely. A particular subset of this volume of facts is important and critical to a reasonable or good decision arising from a specific consultation of an expert system.

The goal of any expert system is to take a problem that has been defined by a language system and the knowledge about the problem as defined by the knowledge system and supply information that supports the decision process. To do so requires a computational means to logically combine the knowledge expressions and problem expressions. This portion of the expert system constitutes the problem processing software. If successful consultations of an expert system are to be performed, the problem processing software must understand the decision maker's requests as stated and have the ability to extract germane information from the available knowledge about the problem domain (Nilsson, 1971). The function of the software is to provide an interface between the user and the knowledge base to arrive at an answer to a specific problem.

In general, the strategies driving an expert system to make an assertion or recommendation fall into two disjoint categories, forward chaining and reverse chaining. Forward chaining begins with the basic knowledge about the problem area. This knowledge is examined in a particular, predefined sequence, with the problem processing software keeping track of the implications of examined fragments of knowledge along the way. This process proceeds until sufficient implications are developed to provide a solution to the specific problem being processed. Reverse chaining, on the other hand, begins with the original problem statement. The problem is decomposed into subproblems, which are in turn broken down into further subproblems, and so forth. The idea is that a small subproblem is likely to be easier to solve than a large problem -- it may be solved by simply looking up a relevant fact or assertion in the knowledge system. By solving all (or even some) of a problem's subproblems, the problem itself can more easily be solved.

Most conventional decision support software also utilize captured expertise, (e.g., spreadsheet templates and linear programs), but the fundamental approach is different. Expert systems attempt to model the process by which experts solve problems, exogenous of the programming environment. Conventional decision support software imposes restrictions on the modeling of the problem solving process by forcing the problem to be modeled by a specific algorithm or series of computations determined by the specific software used.

Hayes-Roth, Waterman, and Lenat identify fundamental limitations inherent in the use of lower level languages and conventional programming in the creation and implementation of rule based expert systems. Expert systems are difficult to specify using conventional programming techniques because both the knowledge base and the problem processing software must be interwoven into the coding of the program. The use of an expert system development environment allows the developer to separate the knowledge base from the problem processing software, which in turn allows the knowledge base to be defined in a modular framework, and thus easier to specify and modify. They go on to point out that with conventional decision support system programming there is a lack of experimental development for computer-based competence, and additionally, there can be a lack of expertise in exploiting computer capabilities. The utilization of expert systems for specific problems allows for the incorporation of intelligently coded beliefs, the explanation of both the results and the line of reasoning used to obtain the results in problem solving terms, and, finally, the use of inference chains dynamically assembled by built-in control procedures to perform efficient searches.

By the nature of conventional decision support system software development, users and experts are generally removed from the programming process (McCarl, et. al.; Nilsson (1971); Nilsson (1980)). This usually results in rapid changes in program requirement specifications during development due to inter-twining of the problem processing software with the knowledge system in the development of the program algorithm. In contrast, expert systems can be easily developed incrementally, thus insuring steady performance improvements throughout the development process.

History and Applications

From the mid 1950's until 1970, the perceived scope and purpose of expert system development was to design systems within which to embody general reasoning methods that were applicable to a wide range of problem areas. The most noted example of such an expert system was the General Problem Solver (GPS) which evolved out of the research of Simon and Newell at the Carnegie Institute of Technology (Newell and Simon (1963)). The goal of GPS was to supply a problem solving environment where, regardless of the problem area addressed, the underlying program did not change. This was achieved by storing knowledge that was specific to a problem in a knowledge system rather than storing the specific knowledge in the problem processing software. Thus, by making changes in the knowledge system rather than modifying the problem processing software, the ability of GPS was extended to solve problems in propositional calculus, symbolic integration, resolution theorem proving, and various other problems (Newell and Simon (1972)).

It is generally recognized today that GPS possessed limited generality in terms of the scope of problems that it solved. With the development of GPS, however, Simon and Newell provided insights into the nature and control of reasoning in computer applications (Nilsson (1980)). By the 1970's researchers had begun transforming the generalized approaches for expert systems to very narrow problem areas such as disease diagnosis, determination of chemical structure of specific molecules, and applied mathematical analysis (Kittler, Fu and Pau; Newell and Simon (1972)).

There are two types of problems commonly encountered where the inherent limitations in the problem solving approach of conventional decision support systems can at least partially be addressed by expert systems. There may be no available, developed, well-defined algorithm or approach to handle specific problems. In other instances, the methodology may be well known but the available data insufficient to adequately utilize the proven algorithm or approach (Bonczek, Holsapple, and Whinston (1981)). Stefik, et. al.; Hayes-Roth, Waterman, and Lenat; and Waterman identify these same basic limitations and suggest three general areas of expert system research and development: analysis, prediction, and combination.

Analysis problems can be divided into two major subgroups, interpretation and diagnosis. Both subgroups involve the dissemination of information about a specific problem. Interpretation type expert systems organize data into a consistent, reasonable description of a problem. DENDRAL which was started in the mid 1960's is an example of an expert system developed for interpretation. DENDRAL infers molecular structure from mass spectrographic information by encoding the heuristic knowledge of expert chemists into rules that control the search for all possible molecular structures, making it possible to obtain a satisfactory answer with a fraction of the effort. This expert system is still used today to explain chemical structure analysis for numerous international users (Lindsay, et. al.). Grain Market Advisor (Thieme, et. al.) is a current example of an interpretive expert system that associates marketing alternatives with circumstances defined by market price data and user supplied subjective beliefs. Diagnostic type expert system are an extension of interpretive expert systems in that they require precise and accurate identification of the source of a problem. PLANT/ds (Michalski, et. al.) infers possible soybean diseases based upon user defined problem scenarios.

Prediction problems make up the second major group of problems that can be readily addressed with expert systems. Quantitative methods such as econometrics and math programming are widely accepted for their ability to analyze data in a predictive framework. The ability of an expert system to include qualitative data in a predictive framework facilitates the adoption of expert systems in this area. PROSPECTOR, an expert system developed in 1979 to give expert advice on finding ore deposits, is an example. This expert system uses reasoning networks to express both judgmental knowledge in the form of rules and static knowledge (facts) about domain objects. PROSPECTOR was instrumental in the discovery of a molybdenum deposit whose value will ultimately exceed $100 million (McDermott).

Combination problems are addressed by both planning and design oriented expert systems. Significant research and implementation of combinatory expert systems has taken place. R1 (or XCON) was designed by Digital Equipment Company to configure customer requests for VAX computer systems. R1 allows a technician to methodically specify components that need to be included in the computer system to meet the customer's needs. Almost 15,000 rules are used to describe the relationship among more than 5,000 different, unique computer components (McDermott).

Benefits of Implementation

Implementation of any new software (design or end-product) has costs and benefits attached to the action. To fully evaluate and understand the implications of implementation of expert systems, the costs and benefits must be identified. However, the assessment of cost/benefit relationships of software development is an extremely difficult task, especially in the area of expert system development (Keen).

Many potential benefits can be attributed to expert systems research and development. Expert systems can provide timely advice when a human expert is unavailable or lacking in available time. Benefits from implementation of expert systems are greatest when unassisted decision makers often make wrong choices as a result of limited availability of expert advice. Because an expert systems is dependent on computer rather than human availability, around the clock operation is a possibility. The expert systems does not get sick; it does not go on vacation; it does not retire or resign; and it can be readily replicated. Properly developed, an expert systems can provide consistent, uniform advice through a thorough and methodical approach to problem solving. It will not have relapses. It will not skip steps or overlook important factors once they have been included in the knowledge base. It is not biased, temperamental, or political in nature (unless such characteristics have been specifically included in the model).

An expert systems can be revised, expanded, and updated. It frees experts for more creative activities. The expert system can substitute for, supplement, or confirm the views of a human expert. Another advantage of the expert system is its transparency. That is, if properly developed, the system's behavior is accessible as well as the system's results. Users can ask 'How?' and 'Why?' questions to reveal decisions made. This is in direct contrast to a conventional decision support systems based on programming or simulation models where the only way to verify the system's reasoning is to simulate the system's behavior by hand.

Newell and Simon (1978) suggest that the further study of procedural rationality may be the greatest benefit to occur from expert systems research and development. Development of expert systems require procedural investigation of the processes that experts use when solving problems.

However, experts often experience difficulty in teaching or transferring their skills to others as well as identifying why mistakes are made in certain situations. Effort to model an expert's behavior in the form of an expert system may make it easier to analyze, evaluate, support, and teach problem solving processes.

Another benefit, albeit hard to measure, is the value that an expert derives form the process of development of an expert system. The methodologies utilized in problem solving are exposed in minute detail to the expert in the development stages of the expert systems. The ability of the expert to place confidences on the outcome of inferences generated by the expert systems point out areas of expertise that may, in fact, be less certain than originally supposed. The identification of the relative knowledge for various aspects of a problem can lead to further research and learning in deficient areas for inclusion in future expert systems or other decision support software.

Expert systems also offer a means of interdisciplinary communications to researchers. The problem solving process that agricultural economics researchers, especially in the area of farm management and production, go through includes narrowly defined areas of personal expertise supplemented with interdisciplinary research and research results from other disciplines. The development of expert systems forces researchers to move into a systems approach to problem solving that may offer a mutually agreeable alternative to more narrowly defined, discipline oriented approaches. By fostering such interaction, expert systems development can provide a means of communicating alternative problem solving methodologies among researchers from various disciplines. Thus the recent interest in expert systems may well encourage the communication and interchange needed in our age of specialization.

Costs of Implementation

The cost associated with development time is one portion of the overall cost associated with expert systems implementation. Commitment of time resources by both the expert and the knowledge engineer can become significant in large development projects. Gremillion and Pyburn make the case that the development of an expert systems intended to overcome a shortage of expert knowledge may actually exasperate the shortage in the shortrun. Costs may be increased to such a level that the project may be infeasible.

Complexity of the proposed expert system and the degree of structure that exists in the problem solving process modeled both affect the overall cost of implementation. As the number of possible solutions and the number of categories of facts and beliefs increase, the cost of acquiring knowledge also increases. Newell and Simon (1981) point out that while outwardly difficult, large complex systems can be simplified by factoring them into a hierarchy of simpler problems or by arbitrarily limiting the scope of the system by concentrating on a subset of the problem area.

Other potential costs that must be examined are related to the legal ramifications associated with the implementation of expert systems. Where does the liability reside for an expert system that makes the wrong diagnosis? Can a user sue the developer for "technical malpractice," or is the user of the system responsible? Answers to questions like these could hold significant and surprising results. To date, there has been no directive or test to determine if expert systems are products or services. The question of vendor and user liability hinges on this point. The answer could have a profound impact on future implementations of expert systems by land grant universities as well as other players in the agricultural sector.

A judicial decision that an expert system is a product would have legal implications for users. If an expert system suggests a bad move, the injured party may have grounds to sue and possible win on strict product liability grounds. Both vendor and user could be held equally responsible regardless of who actually did anything wrong. In a strict product liability lawsuit, the plaintiff does not have to show whose negligence caused the problem, or that the vendor or user failed to adequately test the product. All the plaintiff has to do under strict product liability law is demonstrate that the product belonged to party A (the vendor) and that party B (the user) operated it. This is often enough to hold both parties equally liable.

If the courts opt to classify expert systems as services rather than products, however, the plaintiff's job becomes tougher. Legal recourse would be constrained to negligence laws for redress. If the expert system is viewed as a service, the system itself would be merely treated as another source of information, no different than a journal or reference book. The developer could potentially be covered under the free speech safeguards of the First Amendment. If sued by an injured party, both the vendor and the user can argue that because they are not under any direct contractual obligation to the plaintiff, they have no obligation to test the system for hidden defects.

Although most developers would like the courts and legislatures to view expert systems as services, this has not been the case to date. The judicial attitude on expert systems seems to be leaning towards a reinforced view of expert systems as being products. This kind of judicial reasoning does not bode well for the expert systems development community. If an expert system is viewed as just another product, defects in expert systems could give rise to strict product liability. Programmers, vendors, distributors, and users could be held responsible for problems and errors.

An Application

Van Beek, Fletcher, and Mengel illustrated the melding of the three general categories of expert systems in the early development of GUFERT, an expert systems that recommends nitrogen fertilizer rate adjustments for a prespecified yield goal based on physical characteristics and nitrogen management practices. Further refinements and an economic analysis option have been included in the current version version now called N-Man. N-Man is not an optimization or least cost program but serves as a guide to users by disseminating large amounts of knowledge about interactions between nitrogen fertilizers, corn yield, and factors related to specific nitrogen application situations. N-Man provides expert nitrogen fertilizer rate advice on a timely basis. The use of such a program also provides the opportunity to teach the expert's problem solving techniques to the user.

In any consultation with N-Man, the user is asked for information on soil characteristics, yield goal, and information on nitrogen application(s). Based on this information, further questions are asked to build an information array which is used to make fine tuning adjustments to a general nitrogen recommendation. When all the information is supplied, N-Man determines and reports the final nitrogen recommendation.

Many scenarios and alternatives can be posed to N-Man to identify the cost (material and application) implications of differing nitrogen management systems. One scenario that a producer might consider would include: corn following soybeans, fall plowed, no cover crop, and anhydrous ammonia (NH_3) injected previous to planting with no nitrification inhibitor used. A number of reasonable alternatives can be suggested. Alternative #1 involves changing the base case to the use of Urea Ammonium Nitrate solution (UAN). Alternatives #2 incorporates #1 and changes the tillage system to no-till. Alternative #3 incorporates #2 and adds a non-legume cover crop such as winter rye. Alternative #4 reverts to alternative #1 but puts on 50% of the nitrogen in the "weed/feed" program as UAN and injects 50% of the nitrogen as anhydrous ammonia during late post plant.

N-Man proves recommendatons for each alternative. Recommended nitrogen rates vary from 163 to 195 pounds per acre (Table 1) for a 140 bushel per acre yield on Brookston soils in a corn/soybean rotation. No significant differences are expected between rates of nitrogen for UAN broadcast pre-plant under conventional fall tillage or no-till conditions (both 173 pound per acre) reflecting the minimal effect of soybean residue in decreasing the efficiency of surface applied nitrogen. A significant increase in nitrogen rates accompanies the incorporation of non-legume cover crops in no-till conditions (22 pounds per acre). The nitrogen difference between alternative 2 and 3 is required to compensate for the nitrogen tied up in the residue cover. The alternative #4 nitrogen rate of 163 pounds per acre reflects the added efficiency of split application. Nitrogen is applied closer to the time that the plant will actually utilize the nitrogen thus reducing the risk of leaching or denitrification.

Table 1. N-Man Nitrogen Recommendation for 140 Bushel Corn on Brookston Soil.

Alternative Case	Anhydrous Ammonia	UAN	Total Nitrogen Required	Total Nitrogen and Application Cost
BASE	164	---	164	$28.96
#1	---	173	173	36.37
#2	---	173	173	36.37
#3	---	195	195	40.55
#4	77	86	163	36.62

With these model results, which illustrate conclusions in line with the expert's recommendations, partial budgeting can be used to analyze the economic implication of such changes. Using nitrogen costs of $.14/lb. for NH_3 and $.19/lb. for UAN, application charges of $6.00/ac. for NH_3 application and $3.50/ac. for broadcast application of UAN, costs range from $28.96 for the base case to $40.55 for alternative #3. As can be noted in the two cases presented, anhydrous ammonia is by far the most inexpensive form of nitrogen, and because it requires incorporation at application time, is also one of the most efficient forms of nitrogen to use. Split applications in both cases were substantially more expensive than the base case due, primarily to increased application costs. In fact, the cases illustrate that anhydrous ammonia prices would have to more than double before an economic break-even occurs.

The process of developing N-Man pointed out many of the benefits that land grant universities can anticipate from expert systems research and development. These benefits include the utilization of scarce resources to enhance applied interdisciplinary research, the ability to point out strengths and weaknesses to individual researchers in their own research programs, and the ability to communicate results and methodologies among disciplines and to the end users.

Summary

While "expert systems" is a current buzzword, it is more than just a buzzword. Expert systems are making significant impacts in the scientific and business world as computer systems, software development environments, and the experience necessary to develop and implement useful systems continue to evolve. Expert systems are one part of a much larger framework of decision support tools and aids that are becoming more readily available to individual computer users today.

Expert systems offer a means of disseminating the vast amounts of data and expertise that exist in many areas of agricultural research. Tools are now available for establishing priorities for this data and ranking the level at which it should be made available. In areas of agricultural economics that rely on interdisciplinary research or research results from other disciplines, especially farm management and production, expert systems can serve as a means of communicating problem solving methodologies from other disciplines for inclusion in economically relevant decision support systems.

Successful implementation of expert systems and the eventual general acceptance of this form of decision support systems can be enhanced by following six general steps:

Testing -- All expert systems software that is distributed should be tested thoroughly using accepted, comprehensive evaluation procedures.

Honesty -- Developers should guard against making false or misleading statements of the abilities and functionality of their product.

Disclosure -- Developers should inform the potential users, in writing, of the potential limitations of the particular system.

Training -- Users should be educated in procedures governing the use of a specific system.

Insurance -- Consideration should be given to transferring liability to a third party to remove the burden of uncertain product liability status.

Keep Up -- Because of technological advancements occurring on a daily basis, developers should provide users with information on newly discovered potential limitations of a particular expert system until the new information can be incorporated in the program. To the extent feasible, expert systems should be modified to incorporate new information as soon as possible.

Incorporation of the above mentioned steps into the development and implementation of expert systems should allow for emphasis to be placed on the development of concise, beneficial expert systems and encourage user acceptance.

References

Anthony, R.N. Planning and Control Systems: A Framework for Analysis, Harvard University Press, Boston, MA, 1965.

Bonczek, R.H., C.W. Holsapple, and A.B. Whinston. Foundations in Decision Support Systems, Academic Press, New York, NY, 1981.

Bonczek, R.H., C.W. Holsapple, and A.B. Whinston. Micro Database Management - »OH Practical Techniques for Application Development, Academic Press, New York, NY, 1984.

Chang, C. and R.C. Lee. Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York, NY, 1973.

Copi, I.M. Introduction to Logic, MacMillan, New York, NY, 1961.

Ernst, G.W. and A. Newell. GPS: A Case Study in Generality and Problem Solving, Academic Press, New York, NY, 1969.

Gremillion, L.L. and P.J. Pyburn. "Justifying Decision Support and Office Automation," Journal of Management Information Systems, Vol. 2:5-17, 1985.

Hayes-Roth, F., D.A. Waterman, and D.B. Lenat. "An Overview of Expert Systems." Building Expert Systems (F. Hayes-Roth, D.A. Waterman, D.B. Lenat, eds.), Addison-Wesley, Reading, MA, 1983.

Holsapple, C.W. and A.B. Whinston. Manager's Guide to Expert Systems Using Guru, Dow-Irwin, Homewood, IL, 1986.

Keen, P.G. "Value Analysis: Justifying Decision Support Systems," MIS Quarterly, Vol. 5:1-15, 1981.

Kittler, J., K.S. Fu, and L.F. Pau (eds.). Pattern Recognition: Theory and Applications, Proceedings of the NATO Advanced Study Institute, St. Anne's College, Oxford, England, March 29-April 10, 1981. D. Reidel Publishing Company, Boston, MA, 1982.

Kleene, S.C. Mathematical Logic, Wiley, New York, NY, 1967.

Lindsay, R.K., B.G. Buchanan, E.G. Feigenbaum, and J. Lederberg. Applications »OH of Artificial Intelligence for Chemical Inference: The DENDRAL Project, HPP-78-19, computer Science Department, Stanford University. Palo Alto, CA, 1978.

McCarl, B.A., W.V. Candler, D.H. Doster, and P.R. Robbins. "Experiences with Farmer Oriented Linear Programming for Crop Planning," Canadian Journal of Agricultural Economics, Vol. 25 No. 1:17-30, 1977.

McDermott, J. "R1: The Formative Years," AI Magazine, Vol. 2 no. 2, 1981.

Michalski, R.S., J.H. Davis, V.S. Bisht, and J.B. Sinclair. "A Computer-Based Advisory System for Diagnosing Soybean Disease in Illinois." Plant Disease, Vol. 67:459-463, 1983.

Newell, A. and H.A. Simon. Human Problem Solving, Prentice-Hall, Englewood Cliffs, NJ, 1972.

Newell, A., J.C. Shaw, and H.A. Simon. "Emperical Explorations with the Logic Theory Machine: A Case History in Heuristics." Computers and Thought (eds., Feigenbaum and Feldman), McGraw-Hill, New York, NY, 1963.

Newell, A. and H.A. Simon. "GPS: A Program that Simulates Thought." Computers and Thought (eds., Feigenbaum and Feldman), McGraw-Hill, New York, NY, 1963.

Nilsson, N.J. Principles of Artificial Intelligence, Tioga Publishing Company, California, 1980.

Nilsson, N.J. Problem Solving Methods for Artificial Intelligence, McGraw-Hill, New York, NY, 1971.

Shortliffe, E.H. Computer Based Consultation: MYCIN, American Elsevier, New York, NY, 1976.

Stefik, M., J. Aikins, R. Balzer, J. Benoit, L. Birnbaum, F. Hayes-Roth, and E. Sacerdoti. "Basic Concepts in Building Expert Systems," Building Expert Systems (eds., Hayes-Roth, Waterman, Lenat). Addison-Wesley Publishing Company, Reading, MA, 1983.

Thieme, R.H., J.W. Uhrig, A.D. Whittaker, R.M. Peart, and J.R. Barrett. "Grain Marketing with an Expert System." Paper 85-5519, winter meetings of American Society of Agricultural Engineers, December, 1985, Chicago, IL, 1985.

Van Beek, W.J., J.J. Fletcher, and D.B. Mengel. "EXPERT SYSTEMS - Nitrogen Application Decision Model." PACC '86: Conference Proceedings, Purdue University, December:69-88, 1986.

Waterman, D.A. A Guide to Expert Systems, Addison-Wesley Publishing Company, Reading, MA, 1986.

Winograd, T. Understanding Natural Language. Academic Press, New York, NY, 1972.