KNOWLEDGE ACQUISITION PARADIGMS

By: Joseph D. Mainardi

For: CMSC606 Final Paper

ABSTRACT

The emergence of expert systems technology has been received with mixed results. One of the most common problems that knowledge engineers encounter is in the area of **knowledge acquisition.** The problems associated with assimilating domain expertise are important, as an expert system is likely to fail if the incoming expert information is invalid or inappropriate. The problem of the "knowledge engineering bottleneck" is one that is receiving attention from both research and industry; many new approaches are being examined and tested. The current trend has focused on three knowledge acquisition paradigms: interviewing, machine learning, and neural networks. It is worth noting that there are commercial knowledge acquisition tools available, and they are being applied to expert systems projects.

INTRODUCTION

Knowledge acquisition can be defined as the process of extracting domain knowledge from domain experts. Other definitions include: the accumulation, assimilation and accomodation of new facts with respect to an existing program (which can be extended to include the general task of knowledge engineering, as well as enhancement and general maintenance of existing computer programs), the process of incorporating domain knowledge into an expert system by extracing it from domain experts and encoding the information into internal representations such as rules, an automated process by which a program accepts knowledge from domain experts and incorporates it into an existing expert´s system. Knowledge acquisition is considered the biggest bottleneck in any type of expert system, mainly due to the fact that the process is demanding and time consuming.

The concept of domain expertise can sometimes be confusing in and of itself. The concept of domain in Artificial Intelligence means a working group that shares common knowledge about a particular set of practical problems, and that has developed a specialized "language" for describing and handling those problems. Learning to think like an expert is a complex process involving not only acquisition of principles, but also the discovery of structure. Specialists are distinguished from laymen and general practitioners in a domain by their vast task-specific knowledge, acquired from their training, subsequent readings, and experience of hundreds of cases in the course of their practice.

Knowledge acquisition consists of three primary tasks: entering knowledge into the system, avoiding erroneous knowledge, and augmenting the knowledge to make the system perform as designed. Some of the ways of acquiring knowledge are: observation (work sampling), documentation, and learning by doing. By way of contrast, ethnoscience procedures break down into the following tasks: elicitation, sorting, survey interviewing, ranking, and scenarios. Some of the potential sources of knowledge include: human experts, textbooks and manuals, databases, and personal experiences.

There are many stages in knowledge acquisition. The major knowledge acquisition stages are: problem identification, conceptualization, formalization, implementation, and testing. The identification stage requires the definition of the domain experts and the roles of all persons involved, as well as the clear identification of the problem to be solved. The conceptualization stage takes the key relations and concepts from the identification stage and makes them explicit; this is the point where candidate representations and development tools are discussed, but not selected. The formalization stage involves mapping the key concepts, sub-problems and information flow characteristics into more formal representations. The implementation stage involves mapping the formalized knowledge previously gathered, into the framework of the representation associated with the development tool chosen. This is where the initial executable program comes from, and is almost always what is referred to as the prototype. The prototype is extremely important, in that it tests the adequacy of the formalization and of the basic underlying ideas. The testing stage simply involves the evaluation of the prototype system and the associated representational forms used to implement it.

The need for knowledge acquistion systems are critical. A knowledge acquisition system can help a production-system programmer find errors of both commision and omission in a rule, and attempt to correct those errors, as well as helping to control the mechanics of making the changes. As an example, a rule fired erroneously could be changed, and the system could be reset to an earlier stage to check the new change. A sophisticated knowledge acquisition system can help a programmer explore and manage interactions among rules and the knowledge that those rules represent. It should be noted that knowledge acquisition systems are distinguished from other tools by the fact that they can interact directly with the domain expert, and they automatically acquire and generate knowledge bases. Knowledge acquisition systems come with a caveat: for domain experts willing and able to learn the basics of the inference and representation techniques, a well-structured, easy-to-access set of run-time support tools for knowledge acquisition can be as helpful as it is for the knowledge engineer; otherwise it is not advisable to permit domain experts to use a knowledge acquisition system to modify an expert system.

Current discussions of knowledge acquisition are now being focused on automating the tasks associated with the process; in some circles, this is referred to as autonomous learning. Autonomous learning implies much more than self-organization; certain levels of machine independence are required.

As can be observed, the attempt to reduce knowledge acquisition problems are widely varied, with each paradigm having its own merits. A discussion of the three most prevalent paradigms will be discussed in detail: interviewing, machine learning, and artificial neural networks.

INTERVIEWING TECHNIQUES

Interviewing tools have their foundations in the personal construct psychology of G. A. Kelly, dating back to 1955. In drawing a parallel, anthropological knowledge acquisition has been developed over the past 25 years using ethnoscience; this technique is formal, well-documented and well-tested. The knowledge used in many of the early Artificial Intelligence programs of the 1950's was hand-crafted; a programmer would transform a domain expert's knowledge into code without separating the knowledge from the reasoning mechanism. More recently, the concept of knowledge engineering has become a method of knowledge acquisition, where the main difference is that the domain knowledge is separated from the reasoning mechanism.

The interviewing process is actually quite complex, and involves many different areas of study. There are two interview policies: the scheduled versus un-scheduled, and standardized versus non-standardized, and they are used in all possible combinations. Each of these policies are straightforward, and require no further explanation; a standard dictionary definition would suffice. One key in using the interview process is to begin with a survey interview, which is often used to determine the applicablity of a problem to expert systems technology. A survey interview takes time to prepare and is tedious, but is very helpful when people poorly understand the situation in which they are working. A survey interview is also a check on how well the developer understands the problem.

There are five major points in interviewing: reliability and validity of the data, communication between the knowledge engineer and the domain expert(s) , an interviewing strategy, and interviewing techniques and tactics. In addition, there are some key focal skills required in interviewing. The first is to accurately receive the information. This implies: hearing the information, observing the expert, and remembering the information. Second, one must be able to critically evaluate the information, which requires an ability to recongnize relevant information, and to evaluate potential inhibitors. Finally, the interviewer must be able to regulate their own behavior by self-observation, before, during and after an interview.

Interview-based systems can be classified as knowledge-driven or technique-driven, based on the type of heuristic the system employs to drive the interviewing process. The knowledge-driven systems contain task models, which assist in the identification and classification of new knowledge; the technique-driven systems rely on formal methodology to elicit domain knowledge from the expert. In the case of knowledge-driven systems, all people have theories about domains of knowledge and the way they structure that knowledge. These theories tend to be meta-explanations that follow more closely to the equivalent of learning in a classroom setting. Informal acquisition is the the way experts learned the knowledge that sets them apart; from interacting with knowledgeable people in information-rich surroundings. In the case of technique-driven systems, elicited terms can be compared and sorted to reveal the domain's structure. Elicited terms can also yield both named and unnamed attributes which become the basis for further questions; this allows to investigator to organize deeper inquiry into the knowledge's cognitive arrangement.

Expert system developers agree that the hardest part of writing an expert system is getting the information from the expert. The knowledge engineer's job is to act as a go-between to help build an expert system. Since the knowledge engineer has far less knowledge of the domain than that of the domain expert, communication problems impede the process of transferring expertise into a computer program. There are two major inhibitors of communication. The first is seen as the domain expert´s willingness to respond: competing demands for time, ego threat, trauma, and etiquette barriers. The second is seen as the domain expert´s ability to respond: forgetfullness, chronological confusion, inferential confusion, and unconscious behavior.

Other types of problems also exist. Experts often believe that judgements are based on only a few rules, when actually hundreds of rule's may be required to model a minor domain of behavior. In addition, experts tend to underestimate how much they know, as well as how long and complicated the task of transferring their knowledge will take. Also, experts seldom realize how they structure their knowledge in a cognitive fashion. An equivalent problem plagues anthropologists: how do you acquire "out-of-awareness" knowledge? For both, there are no simple answers.

Poor knowledge acquisition may lead to loss of critical information, as implicit knowledge is cast into explicit rules. Also, experts must be able to convey articulately to knowledge engineers the key concepts and heuristics involved in the domain; experts who can not transfer their expertise will undermine the development effort. One of the most troublesome problems is representation mismatch, or the difference between the way a human expert normally states knowledge and the way it must be represented in the computer program.

Consultations between humans typically occur in a linear question and answer fashion because of the limitations of interpersonal communications, not because it is a superior approach. While people tolerate the human-to-human question and answer approach, they are even less receptive to conversing through a screen and keyboard approach. Anthropologists use a technique called ethnoscience. Ethnoscience techniques are designed to access the experts cognition without relying on shared assumptions and presuppositions. Ethnoscience techniques offer an excellent method of tapping an expert´s knowledge, while preserving the knowledge´s structure. Many of these concepts can be applied to expert system interviewing techniques.

A vast majority of the responsibility for a good interview falls on the knowledge engineer. They must first develop an interviewing strategy before their first information seeking meeting. They must then consider an interviewing strategy. The interviewing strategy factors are in: selecting experts, selecting interviewers, selecting a suitable location, choosing the proper time and place for each interview session, choosing the proper method of recording each interview, and selecting the correct mode of contact. Using proper verbal and non-verbal interviewing techniques increase the chances for the success of an expert system application. Some of these techniques are: supplying proper question context, using appropriate question wording, regulating the scope of questions, listening (as opposed to querying and documenting), pace and inflection of questions, proper question formulation, capturing answers correctly, and having the proper attitude. Interviewing tactics help provide context to the information. This can be controlled by: regulating the sequence of topics, providing smooth transitions, varying the sequence of questions, varying topic control, preventing falsification of information, and conducting an informal post-interview session with the expert(s). Knowledge engineers can improve their interviewing skills in many ways. One way is to plan their interview by: clarifying the purpose of the interview, developing interview objectives, and hypothesizing about possible interview scenarios. Another way is when they are conducting the interview by: pre-testing, guiding oneself through the interview, and using probe notes during the

interview. Finally, they can analyze their interview by: evaluating their interviewing notes, analyzing their strategy/techniques/tactics, and by using introspection and practice.
Under the category of miscellaneous items, the following should be considered when attempting to conduct better interviews. First, remember that concepts often lack a conventional label, and the developer and the expert need to create a convenient vocabulary for referring to them. Initial interviews should be more free form, while later interviews should require more specific questions. Also, initial interviews should ask different questions of each domain expert, and later interviews should ask the same questions of each domain expert. In addition,research recently has begun to focus on ways to decrease the representation mismatch problem. One way is in research on learning by being told, or advice-taking; another method is to allow the expert to converse with the expert system in a natural language.

MACHINE LEARNING

Learning can be defined as any change to a behaving system that alters its long-term performance. Learning may be either a single process or a number of overlapping processes. Experimental and educational psychologists have studied learning by examining how it occurs in humans and other animals. Learning is considered one of the most salient and significant aspects of human intelligence; this cognitive behavior is so poorly understood that very little progress has been made in achieving it in Artificial Intelligence systems. Empirical and explanation-based approaches are the most active sub-fields of machine learning; connectionist learning and genetic algorithms are regaining interest, and incremental learning is being more widely used.

An objective of machine learning is to automate the acquisition of knowledge; by getting a machine to direct the knowledge transfer process and to construct its internal representations itself, there is a hope that there will be a reduction in the cost of expert systems and that they will be provided an efficient means for the systems to continually improve themselves. There are four perspectives on learning: any process by which a system improves on its performance, the acquisition of explicit knowledge, skill acquisition, and a combination of theory formation, hypothesis formation and inductive inference. In recent decades the study of machine learning has become an important research tool. A learning machine is considered any device whose actions are influenced by past experiences. Early machine learning systems learned through self-modification of stored parameters, such as in the Boltzman Machine. Recent learning systems have adapted the production-system model, in which incremental changes in performance are effected by adding new production rules to the existing rule base. Machine learning has a high potential for automating discovery.

Artificial Intelligence research on learning has evolved through three stages. The first stage of work centered on self-organizing systems that modified themselves to adapt to their environment. Learning adds knowledge to a system by generating new rules or modifying old ones automatically. The second stage of learning research is based on the view that learning is a complex and difficult process, which implies that the system cannot be expected to learn high-level concepts by starting without any previous knowledge. The third stage is motivated by the need to acquire knowledge for expert systems, and is examining all forms of learning, not just rote learning and learning from examples.

It is convenient to classify learning into six conceptually distinct categories. Category One is called learning by rote or direct implanting. It is the shallowest form of learning, and requires no inference by the learner. Category Two is called learning from instruction, where knowledge may be handed down from mentor to learner in spoken or written form; this is the same as knowledge acquisition. Category Three is known as learning by analogy, in which the scope of existing knowledge is extended by applying it to new domains. This requires the learner to recognize the similarity tween old and new domains, and to find a transformation that will yield new rules that will work correctly in the new domain when applied to the old rules. Category Four is referred to as learning from examples. Here, the learner abstracts, from a set of specific examples and counter-examples, a general scheme for classifying future instances. Category Five is called learning from observation and discovery, where the unsupervised learner focuses on salient features of an environment in order to form rules about observations. Category Six is known as skill refinement, in which the learner creates rules which permit more efficient performance.

A system can generate new rules through the following syntactic learning methods, which have counterparts in the study of human learning. The first method is generalization, and is used in two senses: the finding of a general principle to describe specific examples, and a mechanism of deriving a general principle by abstracting it from a set of more specific principles. The second method is called specialization or discrimination, and it contrasts the second sense of generalization. It is a mechanism of refining too-general principles by specializing them to apply to fewer cases. The third method is referred to as designation, which creates a new rule directly from instruction or observation. For example, an event is observed and the system searches for the appropriate conditions that made the event occur. The fourth method is called composition, which creates a new rule that summarizes the behavior of two or more rules that fired in sequence. This method can produce rules that proceed more directly to the answer.

There are some general requirements for a system to be able to automate knowledge acquisition for knowledge based systems. First, the knowledge acquisition system tool should be domain independent, and should be directly applicable to the domain expert. The knowledge acquisition tool should be able to access a diversity of knowledge sources, as well as having the ability to encompass a diversity of perspectives, and forms of knowledge and their associated relationships. It should also be able to present knowledge from a diversity of sources. In addition, users of the system should be able to apply the knowledge in a variety of familiar domains. The system should provide validation study capabilities, and it should converge to an integrated system.

Several issues stand in the way of using machine learning as a cost-effective method of knowledge acquisition. The first of these is that meaningless and/or trivial generalizations can result from the weak inferencing methods used to select the appropriate generalizations of the data. Another important issue is that systems that are specially tuned to a problem domain by trail and error discovery may require more effort to develop a learning program than would be required in a manual knowledge acquisition process. There are three ways by which learning techniques may become cost-effective. One way occurs when the large costs of automation can be offset by outperforming manual knowledge acquisition methods when reaching an efficient level of sophistication. Also, general learning systems can be developed. Finally, automated knowledge acquisition can become cost-effective whenever the domain has no human experts because of its novelty and/or complexity.

It is important to note that explaining something doesn't mean that someone is learning it, and that the explanation may require just as much explanation as what is being explained. One author notes that his favorite theory of pedagogy is that people learn by analogy, wherein students tend to construct an analogouos meta-model to any model that they are presented with. Obviously, this is not really learning, but a substitute for learning.

There is a real need to focus on the acquisition of the kinds of knowledge that are difficult to acquire manually but for which automated methods are feasible, in order to maximize the benefits of automated knowledge acquisition. A system must learn to perceive distinctions and learn where to direct attention. These skills are tacit not explicit, hence they are learned slowly, Explanation-based reasoning is a technique for obtaining generalized concept definitions based on an analysis of an example using a domain theory.

A major concern among machine learning researchers is that the cost of computer resources is falling rapidly, while the cost of human labor is increasing. This implies that the benefit of reducing human resource costs far outweigh the costs in computer resources for automated knowledge acquisition and machine learning. The possible benefits of automating knowledge acquisition are that automated methods may prove more competent than humans for fine-tuning or acquiring certain kinds of knowledge, and they could significantly reduce the high cost of human resources involved in constructing expert systems.

Autonomous systems require many qualities. The first of these is that they must organize knowledge into associated categories with no human assistance and reliably retrieve related information from garbled input data. Another quality is to provide rapid response when asked to recall a learned input pattern. Autonomous systems should also learn arbitrarily complex input patterns without placing restrictions on those patterns. In addition, it should learn constant and significant information, and know what significant information is. Also, an autonomous system doesn´t destroy already learned information. It should also have the ability to decide when information goes from significant to insignificant and vice-versa. Autonomous systems must be able to reorganize associative categories when appropriate, and it must generalize from specific examples. Finally, autonomous systems must have unlimited storage capacity for its expected lifetime.

NEURAL NETWORKS

Neural networks were originally built by researchers investigating the general question of how a mass of brain cells can host what people perceive as intelligence. When referring to neural networks, a more appropriate reference would be artificial neural networks; that is the context implied here. A neural network can be viewed as a logical machine which structurally and functionally resembles a biological brain. Neural networks appear to be promising alternatives to traditional computer solutions in pattern recognition and search space optimization, for example. Neural networks are often given serious consideration as a way around the "knowledge acquisition bottleneck" problem, by accessing the domain data directly and omitting the human interpreter/expert altogether. Systems based on neural network methodologies offer the promise of exceedingly fast and robust implementations that can conveniently and flexibly be trained to respond to a set of given situations in an appropriate manner.

In more of a technical perspective, neural networks are composed of massively parallel, interconnected processing elements, each of which is connected to others by means of directed links with associated weights. Although neural networks work on the basic premise of parallel processing, they are different. Parallel processing has to do with arranging and connecting digital computers to facilitate concurrent operations for a single problem. Processing elements in a neural network can fire at the same instant, and in reality they are crudely reflecting the parallel structure found in brain tissue.

How the brain works is not completely understood, so a neural network should be considered a simplified model of nature. There are many differences between biological neuron and a simulated neurode. Some of these are now discussed. The first difference is that brain neuron connectivity ranges between 10,000-to-1 and 100,000-to-1, while an artificial neural network becomes difficult to manage a 100-to-1 level of connectivity. Another difference is that biological neurons are electrochemical (the synaptic weights are moderated by chemistry and electrical signals), while artificial neurodes are only electrical signals. In addition, biological neurons are fully analogous, while artificial neurodes are not. It should be noted that the behavior of the neural network depends on the weights assigned to the connections between the layers. Additionaly, the assignment of the weights are determined when a neural network is trained to associate input patterns representing sensor data to output patterns. These output patterns represent particular conditions.

Neural networks can be categorized according to various attributes. The first attribute is feedforward versus feedback, with respect to the output from one processing element being allowed to be an input to a processing element from a previous layer. Another attribute is continuous versus discrete operation, where continuous operation implies that processing elements execute whenever input values are available, and discrete operation takes action only when specific commands have been executed. There is also the issue of biological versus non-biological models, but distinctions are diffucult to delineate at this time. Another attribute is that of structured versus random connectivity, which is project dependent. One final attribute to consider is that of dynamic conditioning versus learning. Dynamic conditioning is considered unsupervised training and learning is considered supervised training.

Neural networks are attractive options to conventional methods for various reasons. First, they don't need to be programmed, although the learning process, or set-up, still

is a manual process. In addition, they are fault-tolerant with respect to processing element failures, due to the many interconnections between processing elements. Also, they can provide approximate answers, and no complete representation is mandatory. Finally, they "learn" from examples, although "conditioning" from examples might be a better description.
Understanding the limitations of nueral networks is very important, as it is not a solution for every type of problem. First, the strength of the neuron's response is measured by the frequency with which the neuron fires, not with the size of its output. Also, negative frequencies have no meaning, and each neuron has a maximum firing frequency beyond which it cannot go. The use of the back propigation algorithm to develop a knowledge base illustrates the ease and appropriateness of implementing neural networks that deal with implicit knowledge; when back propigation is used, the limitation is that there can be no explanations for a conclusion.
Understanding how stimuli react within a neural network is also a critical consideration. If the incoming stimulus is increasing in strength and the output of the receiving neurode also goes up, then the changes in both go up. The stimulating neurode will therefore find it easier to excite the receiving neurode in the future. When the activity of the receiving neurode goes down when the recieving stimulus increases, it will be harder to stimulate the receiving neurode in the future, but easier to inhibit. In essence, if the stimulus increases or decreases when the output changes in the same way, the overall weight will be positive, and the signal has an overall excitatory effect. If the stimulus increases or decreases when the output changes in opposite way as the receiving neurode, the overall weight change will be negative, and the signal has an overall inhibatory effect. In conclusion, if a neurode's activity increases, all weights

involved increase; if a neurode's activity decreases, all weights involved decrease.

There are some solutions to typical neural network design problems. One simple solution, for non-linearly separable problems, is to use a heirarchical or layered neural network. Simple single-layered networks cannot solve this type of problem. The selection of the proper method of organizing neural networks can make a significant difference in performance. There are three typical methods for organizing a neural network. The first method is supervised learning, in which an omniscient teacher provides the network with the proper category for each input pattern. Another method is unsupervised learning, where there is no explicit external teacher, and the network organizes its own recognition categories. The last method is learning with a critic, which involves an external teacher merely providing feedback relating to the correctness of a particular category. More generically speaking, problems can be controlled by properly implementing the stability-plasticity principle. This is a design restriction that requires the network to be relatively changeable (plastic) when significant new stimuli are presented, but unchanging (stable) when noisy or irrelevant stimuli are presented.

NEXTRA: A KNOWLEDGE ACQUISITION TOOL

NEXTRA is a high-end knowledge acquisition and transfer tool for expert systems development. It elicits knowledge from the expert through interactive interviewing, which is then structured into logical relationships. The information then displays the analysis of the knowledge in a graphical representation. It provides system developers with structure and focus for expertise that is required in the system being developed, and it automatically generates rules and objects for an associated expert system shell (NEXPERT *OBJECT*).

NEXTRA combines several knowledge acquisition techniques, including automated repertory grids and heirarchical clustering (with spatial representations) to elicit and process data. It uses inductive reasoning to construct objects and rules from the information supplied. In addition, it automatically generates rules, classes and objects and imports them to NEXPERT *OBJECT*, as well as displaying the knowledge base structure as graphical rule and object networks.

NEXTRA provides the tools to acquire, visualize, validate and maintain knowledge bases. It is designed to involve domain experts in the exploration of their conceptual structures and thinking processes. It can also provide a long-term knowledge management tool for the maintenance, enhancement and general management of knowledge bases.

NEXTRA offers [illegible] following facilities: flexible control, interactive i[illegible]view, interactive analysis, knowledge transfer, and [illegible]ibility with other programs. The link to NEXPERT *OBJECT* [illegible]seful, but can only be used as a stand-alone; it is [illegible]ly difficult to insert NEXTRA knowledge into an existi[illegible]PERT *OBJECT* application.

COGENSYS JUDGEMENT SOFTWARE

COGENSYS Judgement Software uses inductive learning to elicit information from human experts. Its purpose is to capture and emulate the decision-making process of an effective human expert within a specific domain. The processes and methods have allowed the inventor to receive one of the few U.S. patents issued for a software product. The commercial product line associated with this tool is financial system based.

Some key features of Judgement Software are: reductions in the knowledge engineering phase of expert systems projects, allowing gradual system integration in phases, its easy deployment, its adaptability to changes in domain climate, its user-oriented design, and the fact that it makes use of learning by observation and interaction methodologies. The features are tied together by three interrelated and patented technologies: the Judgement Processor, the Information Manager, and the Applications Manager.

The Judgement Processor is designed to learn the logic of a mentor by observing real examples of the mentor's decision-making process, while the mentor continues to function in day-to-day tasks. There is an initial period where the expert mentor defines significant decision-making factors to the system, as well as categorizing the expected range of answers. The learning process then begins, by having the expert enter information as part of the daily work load. The associated Judgement Base will then be able to make decisions based on the expert's day-to-day input into the system. The Information Manager is a user-friendly front-end that has the ability to create and manipulate electronic versions of forms. The Applications Manager is a flexible tracking and scheduling system that manages the activities of both the Judgement Processor and the Information Manager.

CONCLUSIONS

From the information presented, knowledge acquisition appears to be the section of experts systems development that must be improved, if production expert systems are to ever gain wide recognition in the commercial workplace. Some of these changes will occur faster than others, but they must occur if expert systems technology is to flourish.

In the area of interviewing, the technically competent knowledge engineer will need to develop strong interpersonal communication skills. The future may also lead to the creation of a new vocation called knowledge acquisition engineers; these people will be trained in basic technology and will come from backgrounds where the emphasis is on communications and interpersonal skills. Also, a new knowledge acquisition method could involve the absorption of knowledge directly from textbooks, via a text understanding program. It appears to be a bit premature to believe that machine learning or neural networks will be the savior of expert systems; they appear to headed towards their own mutually exclusive domains of technology.

There may still be a time in the future where these three conflicting and contrasting knowledge acquisition paradigms come together to form a powerful technology. One that will be powerful enough to survive the onslaught of nay-sayers and conventional progamming defeatists, and in turn provide strong solutions for the problems of tomorrow.

REFERENCES

1. PROGRAMMING EXPERT SYSTEMS IN OPS5; L. Brownston, R. Farrell, E. Kant; Addison-Wesley Publishing Co., Inc., 1985

2. A conference report: THE FIFTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING; U. Fayyad, J. Laird, K. Irani; AI MAGAZINE, Summer 1989

3. ARE HEIRARCHICAL NETWORKS BETTER?; M. Caudill; AI Expert, June 1988

4. SEQUENCE INTOLERANCE IN EXPERT APPLICATIONS; C. Rovira; AI Expert, April 1989

5. NEURAL NETWORKS PRIMER - Part VII; M. Caudill; AI Expert, May 1989

6. NEURAL NETWORKS PRIMER - Part VIII; M. Caudill; AI Expert, August 1989

7. NATURALLY INTELLIGENT SYSTEMS; M. Caudill, C. Butler; MIT Press, 1989

8. Picking The Right Expert System Application; J. Casey; AI Expert, September 1989

9. Knowledge Acquisition In The Peruvian Andes; R. Benfer, L. Furbee; AI Expert, November 1989

10. HIERARCHICAL NEURAL APPROACH; R. Johnson; Electronic Engineering Times, 8 January 1990

11. AI MUST CATER TO NONEXPERTS; R. Moore; Computer Design, 15 February 1986

12. THE HANDBOOK OF ARTIFICIAL INTELLIGENCE - VOLUME 1; A. Barr, E. Feigenbaum; HeurisTec, Inc., 1981

13. THE HANDBOOK OF ARTIFICIAL INTELLIGENCE - VOLUME 2; A. Barr, E. Feigenbaum; Addison Wesley Publishing Co., Inc., 1982

14. THE HANDBOOK OF ARTIFICIAL INTELLIGENCE - VOLUME 3; P. Cohen, E. Feigenbaum; Addison Wesley Publishing Co., Inc., 1982

15. THE ELEMENTS OF ARTIFICIAL INTELLIGENCE; S. Tanamoto; Computer Science Press, Inc., 1987

16. BUILDING EXPERT SYSTEMS - Constructing an Expert System (Chapter 5); D. Waterman, B. Buchanan, et al; Addison Wesley Publishing Co., Inc., 1983

17. KNOWLEDGE ENGINEERING IS THE KEY TO SUCCESSFUL TECHNOLOGY TRANSFER; L. Geisel; Expert Systems: Planning/Implementation/ Integration, Auerbach Publishers, Volume 1, Number 2, Summer 1989

18. JET AND ROCKET ENGINE FAULT DIAGNOSIS IN REAL-TIME; W. Dietz, E. Kiech, M. Ali; Journal of Neural Network Computing, Auerbach Publishers, Volume 1, Number 1, Summer 1989

19. THE SIGNIFICANCE OF SLEEP IN MEMORY RETENTION AND INTERNAL ADAPTATION; C. Tapang; Journal of Neural Network Computing, Auerbach Publishers, Volume 1, Number 1, Summer 1989

20. SELF-ORGANIZATION, PATTERN RECOGNITION, AND ADAPTIVE RESONANCE NETWORKS; D. Stork; Journal of Neural Network Computing, Auerbach Publishers, Volume 1, Number 1, Summer 1989

21. A DESKTOP NEURAL NETWORK FOR DERMATOLOGY DIAGNOSIS; Y. Yoon, R. Brobst; Journal of Neural Network Computing, Auerbach Publishers, Volume 1, Number 1, Summer 1989

22. NEURAL NETWORK TECHNOLOGY OVERVIEW; author unknown; McDonnell Douglas Corporation (unpublished), 26 Sep 1988

23. FUNDAMENTALS OF HUMAN-COMPUTER INTERACTION; A. Monk, A. Kidd; Academic Press, 1984

24. EXPERT SYSTEMS STRATEGIES AND SOLUTIONS IN MANUFACTURING AND DESIGN; A. Kusiak; Society of Manufacturing Engineers, 1984

25. ON THE ROAD TO AUTOMATIC KNOWLEDGE ENGINEERING; J. Patel; Proceeding of the Eleventh International Joint Conference on Artificial Intelligence, Volume 1, 1989

26. INTEGRATING KNOWLEDGE ACQUISITION AND PERFORMANCE SYSTEMS; A. Rappaport, B. Gaines; AAAI Workshop on Integration of Knowledge Acquisition and Performance Systems, 1988