Expert Systems

Daniel E. O'Leary, University of Southern California

INTRODUCTION

In the early 1970's there was substantial interest in studying decisions by experts that did not use statistical or other mathematical tools, and determining if and how such decisions could be modeled in a computer. In particular, there was an interest with investigating conceptual and symbolic methods appropriate for modeling physician and other expert decision making e.g., Shortliffe (1). Out of this environment, the notion of an expert system evolved.

The concept of expert systems is almost magical: simply capture human expertise and put it into a computer program. Rather than worry about a person, a computer program that includes all of the relevant and appropriate knowledge could be developed and shipped around the world. For example, Rose (2) reported that Southern California Edison (SCE) had an expert whose trouble shooting had helped keep a dam safe. However, SCE was afraid their expert would retire or quit, and they worried that he might "get hit by a bus." As a result, SCE planned on using an expert system to try to "clone" one of their engineers, ultimately creating a computer program that captured his expertise.

With such hype, it is probably not surprising that, unfortunately, expert systems never lived up to their hype. Bobrow et al. (3) noted that the term "expert" may have created unrealistic expectations about what a computer program could do. Unfortunately, as noted by Business Week (4) "… grandiose promises of problem solving 'expert in a box' proved illusory." However, the term "expert" also generated "commercial" hopes for a discipline that had been academically based (e.g., Shortliffe (1)).

Electronic copy available at: http://ssrn.com/abstract=2349274

As a result, that same Business Week article also noted that expert systems had rapidly proliferated throughout finance in business applications, being used for a range of activities such as market analysis to credit evaluation. From the early to mid 1970's to the mid 1980's, expert systems application base seemed almost universal. Since then expert systems have been applied to just about every conceivable discipline, ranging from chemistry to medicine to business.

The term expert system apparently began to be replaced by the term "knowledge-based system" in the mid 1980's to mid 1990's (e.g., Hayes-Roth (5) and Davis (6)). The shift was one that would begin to remove the need for labeling a system with "expert," and reduce the hype, but still would require that the system be "knowledge-based." This name shift put less direct pressure on developers to build systems that were equivalent to experts, but also was sign of a commercial and research shift away from expert systems and an evolution to other forms of problem solving approaches.

Purpose and Scope

The purpose of this chapter is to review key concepts in expert systems, across the life cycle of expert system development. As a result, we will analyze the choice of the application area for system development, gathering knowledge through so-called knowledge acquisition, choosing a knowledge representation, building in explanation and verifying and validating the system.

Although it would be easy to focus only on the technical issues, a key finding in the expert systems literature was noted as businesses actually began to implement expert systems (e.g., Barker and O'Connor (7)): In order "… to successfully develop and provide ongoing support for expert systems and to integrate them into the fabric of one's business, … one must attend to the needs of the business and to human resource and organizational issues as well as to technical issues." One of the pioneers and one of the first developers of expert systems, E. Feigenbaum

Electronic copy available at: http://ssrn.com/abstract=2349274

was quoted as saying (Lyon (8)), "I'm not interested in theoretical concepts. I like to see my work used in the real world." Accordingly, we will not only consider technical issues, but also some of the non technical organizational and people issues, along with the applications.

Expert systems were a key part of pioneering artificial intelligence efforts to model human behavior. Expert systems have led to substantial additional and emerging research. Accordingly, this chapter also briefly investigates some of those additional and emerging issues.

Finally, in an article of this type it is inevitable that some key works and important researchers are omitted. There also is limited space, so some topics that might be addressed are not. The author apologizes in advance for any such omissions.

Outline of this Chapter

This chapter proceeds in the following manner. This first section has provided an introduction and statement of purpose and scope. The second section investigates expert systems and human reasoning, while the third section analyzes the structural nature of an "Expert System." The fourth section provides some definitions of an expert system. The following two sections analyze some characteristics of expert system applications and investigate some expert system applications. Then the following five sections trace expert systems through the life cycle of choosing an application that is likely to work, knowledge acquisition, knowledge representation, explanation and verification and validation of the system. The final three sections investigate, respectively, expert system strengths and limitations, extensions and emerging issues, followed by a brief conclusion.

EXPERT SYSTEMS AND HUMAN REASONING

Initially, computer scientists were interested in capturing non-quantitative decision making models in a computer, and they used expert systems to generate those models (e.g., Shortliffe (1)). What were some basic assumptions about human reasoning that drove expert systems? Perhaps the initial primary assumptions were:

- Experts know more than non experts.
- People use information and knowledge based on their past experience.
- People use heuristics.
- People use specific, a priori rules to solve problems.
- People use focused knowledge to solve problems.

Experts Know More Than Non-Experts

Expert systems assume that experts know more or at least something different than non-experts in a field. Accordingly, expert systems assume that experts are differentiated from non-experts by what knowledge they have. As a result, capturing that expert knowledge, can potentially change the knowledge of non experts.

People use Past Experience

People use their past experience (actions, education, etc.) as the basis of how to solve problems. As a result, system developers interested in building systems to solve problems can consult with people to try to capture that past experience, and use it to solve problems where that experience could be used.

People use Heuristics

Heuristics are so-called "rules of thumb." Past experience often is captured and summarized in heuristics. Rather than optimize every decision, people saticfice (e.g., Simon (9)) using heuristics that they have found from past experience drive them toward good, feasible solutions. In order to solve complex problems, expert systems assume that it is possible to capture those heuristics in a computer program and assemble them for re-use.

People use Rules

Much of everyday and business problem solving seems based on rules. For example, when choosing a wine for diner, simple rules such as "if the diner includes a red meat then the wine should be red," help guide dinners to the choice of a wine. People use rules to solve problems. As noted by Clancey (10), rules were seen as a simple and uniform approach to capturing heuristic information. Heuristics and other knowledge are captured and kept in a rule-based form. If people use rules then computer programs could use those same rules to solve problems.

Problem Solving Requires Focused Knowledge

Expert systems researchers, e.g., Feigenbaum (8) note that one view of human intelligence is that it requires knowledge about *particular* problems and how to solve those particular problems. Accordingly, one approach to mimicking human intelligence is to generate systems that solve only particular problems.

STRUCTURAL NATURE OF EXPERT SYSTEMS

Because it was assumed that human problem solvers used rules and knowledge could be captured as rules, rule bases and their processing were the critical component of expert systems. Accordingly, the structure of a classic expert system was designed to meet those needs. Ultimately, expert systems were composed of five components: data/database, user interface, user, knowledge base/rule base, and an inference engine to facilitate analysis of that knowledge base.

Data

The data used by the system could include computer-generated data, data gathered from a database, and data gathered from the user. For example, computer-generated data might derive from an analysis of financial statement data as part of a program to analyze the financial position of a company. Additional data might be selectively gathered straight from an integrated database. Further, the user might be required to generate some assessment or provide some required data. Typically, initial expert systems required that the user provide key inputs to the system.

User Interface

Since the user typically interacted with the system and provided it with data, the user interface was critical. However, an expert system user interface could take many forms. Typically, there would be a question from the system, and the user would select one or more answers from a list, as in a multiple choice test. From there, the system would go to another question, and ultimately provide a recommended solution. In some cases, the user would need to analyze a picture or a movie clip in order to answer the questions.

User

In any case, in a classic expert system, the user is a key component to the system, since it is the user who ultimately provides environmental assessments, generates inputs for the system and as a result, disambiguates the questions provided by the system to gather data from the user. Research has found that different user groups, e.g., novice or expert, given the same interrogation by the system, would provide different answers. As a result, generating that user interface and building the system for a particular type of user are critical.

Knowledge Base/Rule Base

The knowledge base typically consisted of a static set of "if … then …" rules that was used to solve the problem. Rules periodically could be added or removed. However, the knowledge needed for solving the particular problem could be summarized, and isolated. In addition, the very knowledge that was used to solve an inquiry also could be used to help explain why a particular decision was made. (Some researchers also include explanation facility as its own component.) Accordingly, gathering, explaining, and verifying and validating that knowledge is the focus of most of the rest of this discussion in this paper.

Inference Engines

Inference engines facilitate use of the rule-base. Given the necessary information as to the existing conditions provided by the user, inference engines allow processing of a set of rules to arrive at a conclusion by reasoning through the rule base. For example with the system “if a then b,” and “if b then c” would allow us to “reason” that a led to b and then to c. As rule-based systems became the norm, the inference engine saved each developer from doing the same thing, and allowed developers to focus on generation of the knowledge base. Developers became referred to as knowledge engineers. Ultimately, data was gathered, combined and processed with the appropriate knowledge to infer the matching solution.

Expert System Software

Because the expert system components were distinct, software could be designed to facilitate the ability of developers to focus on problem solution, rather than building the components themselves. As a result, a wide range of so-called “expert system shells” were generated, for example (e.g., Richter (11)), EMYCIN (from MYCIN, Buchanan and Shortliffe (12)), ART (Automated Reasoning Tool by Inference Corporation), M.4 (http://www.teknowledge.com/, (13)) or Exsys (http://www.exsys.com/).

WHAT IS AN EXPERT SYSTEM?

The term expert system has been broadly applied to a number of systems, apparently for a number of different reasons. At various points in time, the term “expert system” has implied a type of knowledge representation, a system to perform a particular task, the level of performance of the system,

Rule-based Knowledge Representation

As noted above, people appeared to use rules to reason to conclusions, and experts were seen as supplying rules that would be used to guide others through task solution. As a result, most of the so-called “expert systems” probably were “rule – based systems.” Researchers had seen that this type of reasoning apparently was often used by people to solve problems. So-called experts seemed to reason this way, so the systems were "expert."

Activity/Task of the system

Another rationale for labeling a system an "expert system," was because the system performed a specific task that human experts did. Experts seem to structured reasoning approaches that could be modeled to help solve various problems, e.g., choosing a wine to go with diner.

Level of performance of the system

One perspective was that a system was an "expert system" if it performed a task at the level of a "human expert." For example, Buchanan and Feigenbaum (14) argue that the DENDRAL system functioned at the same level as a human expert.

System Dependence

However, although the system was expert, it was generally still dependent on people for environmental assessments of conditions, and corresponding data input. Expert systems generally were dependent on the user for a range of activities and thus dependent on the user. For example, as noted in Hart et al. (15, p. 590) "… facts recorded in databases often require interpretation." As a result, most of the self proclaimed expert systems typically provided an interactive consultation that meant that the system was still dependent on people.

Definitions

Accordingly, over the years there have been a number of definitions of expert systems, including the following:

- A program that uses available information, heuristics, and inference to suggest solutions to problems in a particular discipline. (answers.com)
- "The term expert systems refer to computer programs that apply substantial knowledge of specific areas of expertise to the problem solving process." (Bobrow et al. (3), p. 880)

- " … the term expert system originally implied a computer-based consultation system using AI techniques to emulate the decision-making behavior of an expert in a specialized, knowledge-intensive field." (Shortliffe (1), p. 831)

As a result, we will call a system an expert system when it has the following characteristics:

- a rule-based approach is used to model decision making knowledge, and that those rules may include some kind of factor, to capture uncertainty
- interacts with a user from whom it gathers environmental assessments, through an interactive consultation (not always present)
- designed to help facilitate solution of a particular task, typically narrow in scope
- generally performs at the level of an informed analyst.

CHARACTERISTICS OF EXPERT SYSTEM APPLICATIONS

Since expert systems related to the ability of a computer program to mimic an expert, expert systems were necessarily about applications, and comparing those human experts and systems. The initial goal of expert systems at some level was often to show that the system could perform at the same level as a person. But as they put these systems in environments with people we began to realize a number of key factors. First, typically, in order for there to be a rule-base to solve the problem, the problem will need to be structurable. Second, systems may support or replace humans. Third, one of the key reasons that a system might replace a human is the amount of available time to solve the problem, not just knowledge.

Structured vs. Unstructured Tasks

Expert systems and their rule-based approaches rely on being able to structure a problem in a formal manner. Rules provided a unifying and simple formalism that could be used to structure a task. Thus, although the problem may not have had sufficient data to be analyzed statistically, or could not be optimized, there was still information that facilitated structuring the problem and knowledge about the problem in a formal manner.

Support vs. Replace

Expert systems were often seen as a vehicle to replace human experts (e.g., Rose (2)). Many systems apparently initially were designed to replace people. However, in many decision making situations, the focus was on providing a decision maker with support. For example, as noted by Kneale (16) in a discussion of an accounting system ExperTAX, the expert system is not designed to replace accountants, but instead enhancing and supporting advice for people.

Available Time

Another important issue in the support vs. replace question was how much time was available to make the decision. If a problem needed to be solved in real time, then perhaps support was out the question, particularly if there were many decisions to be made. Further, even if the system was to support an expert, perhaps it could provide insights and knowledge so the expert did not need to search for information elsewhere.

APPLICATIONS

Because of its focus on modeling and mimicking expertise, ultimately, the field of expert systems has been application oriented. There have been a large number of applications of expert systems, in a broad number of different areas, including Chemical Applications, Medical diagnosis, Mineral Exploration, Computer Configuration, Financial Applications, Taxation Applications. Applications have played an important role in expert system technology development. As expert system technologies and approaches were applied to help solve real world problems, new theoretical developments were generated, some of which are discussed below.

Chemical Applications

Some of the earliest applications of expert systems took place in this arena (e.g., Buchanan and Feigenbaum (14)). DENDRAL and Meta-DENDRAL are programs that assist chemists with interpreting data. The DENDRAL programs use a substantial amount of knowledge about mass spectrometry to help with the inference as to what a compound may be. The output from the program is a detailed list with as much detail as the program can provide. Ultimately, Buchanan and Feigenbaum argued that the program had a level of performance equal to a human expert.

Medical Diagnosis Expert Systems

Medicine was one of the first applications of expert systems. By 1984, Clancy and Shortliffe (17) were able to present a collection of papers covering the first decade of applications in this domain. Shortliffe (1) briefly summarized some of the contributions of medical expert systems to medicine. MYCIN was a success at being able to diagnose infectious diseases. Present Illness Program (PIP) generated hypotheses about disease in patients with renal disease. INTERNIST-1 was a system designed to assist diagnosis of general internal medicine problems. Since that time there has been substantial research in medical expert systems. One of the critical developments associated with medical expert system was using uncertainty on rules (e.g., (18)), which is discussed further below.

Geology Advisor

In geology, an expert system was developed to assist in the analysis of drilling site soil samples for oil exploration. PROSPECTOR I and II ((McCammon (19)) were built with over 2,000 rules capturing information about the geologic setting and kinds of rocks and minerals, to help geologists find hidden mineral deposits. PROSPECTOR I (Duda et al. (20) and Hart (15) was developed along with an alternative representation of uncertainty on rules that garnered substantial attention and is discussed further below.

Computer Configuration

Configuration was one of the first major industrial applications of expert systems. Perhaps the best known configuration expert system was XCON, also known as R1 (e.g., Barker and O'Connor (7)). XCON was touted as the first expert system in daily production use in an industry setting. At one point in time, XCON was only one of many expert systems in use in at the computer manufacturer “Digital” to configure hardware and software. As an expert system, XCON was used to validate the customer orders for technical correctness (configurability) and to guide order assembly. Barker and O’Connor (7) also describe a number of other expert systems that were in use at Digital during the same time as XCON, including

- XSEL, that was used interactively to assist in the choice of saleable parts for a customer order
- XFL, that was used to diagram a computer room floor layout for the configuration under consideration
- XNET, used to design local area networks to select appropriate components

Probably not surprisingly, these industrial applications had very large knowledge bases. For example, as of September 1988, XCON had over 10,000 rules, XSEL had over 3500 rules, XFL had over 1800 rules and XNET, a prototype had roughly 1700 rules.

Taxation Applications at the IRS

Beckman (21) reviewed and summarized the taxation applications expert systems literature, and provided a focus on applications at the Internal Revenue Service (IRS). Throughout the IRS’s involvement in artificial intelligence starting in 1983, the IRS focused on the ability of the technology to help solve real world problems. As reported by Beckman (21) a number of expert system projects were developed and tested, including the following. A “tax return issue identification” expert system was designed to help identify individual tax returns with “good audit potential.” A “reasonable cause determination” expert system was developed because it was found that the error rate by people was too high. As a result, the system was designed to improve the consistency and quality of so-called “reasonable cause determinations.” An “automated under-reporter” expert system that was designed to help tax examiners assess whether individual taxpayers properly reported income.

Auditing and Accounting

The fields of auditing and accounting have generated a substantial literature of applications. The notion behind the development of many such systems was inviting: auditors and accountants used rules to solve many problems that they faced. Expert systems were used to model judgment decisions made by the participants. Brown et al. (22) provides a recent survey of the field.

CHOOSING AN APPLICATION

There have been two basic perspectives on choosing an application to build an expert system. Prerau (23), Bobrow et al. (3) and others have analyzed what characteristics in the domain were important in the selection of a problem around which to build an expert system. Their perspective was one of how well the needs of the domain met the needs of the technology: chose the right problem so that the expert system technology can blossom. Alternatively, Myers et al. (24) and others have viewed it from the business perspective, stressing the need for making sure that the system was in an area that was consistent with the way the company was going to run their business: make sure that the expert system application met the objectives of the company developing it. In any case a number of issues were suggested as conditions that needed to be considered when the domain and expert system application were aligned, including the following.

Art and Science

Hart et al. (15, p. 590) note that “Mineral exploration is perhaps as much an art as science, and the state of this art does not admit the construction of models as rigorous and complete, as, say, those of Newtonian mechanics.” If there is a scientific model then there is no need for a rule-based approach; the scientific model can be used.

Expertise Issues

Since expert systems are dependent on human experts as a source of their knowledge, there must be experts that can work on the project. For example, Prerau (23) notes the importance of having access to an expert from which expertise can be gathered, and that expert must have sufficient time to spend on the project development. Other concerns such as willingness to work on the project also must be considered.

Benefit

In addition, the task should be one that provides enough returns to make it worth while. There is no sense in building a system if the value to the builders does not exceed the costs.

Testability

Since the system is to be categorized as an expert system, there is a need to see if the system performs appropriately. This requires that the results are testable.

KNOWLEDGE ACQUISITION

Early expert systems research was not so much concerned with knowledge acquisition or any other issues, per se. Instead the concern was mostly about the ability of the system to mimic human experts. However, over time as more systems demonstrated the feasibility of capturing expertise, there was greater attention paid to knowledge acquisition.

In general, expert system expertise was initially solicited in a team environment, where programmers and the expert worked hand-in-hand to generate the system. Faculty from multiple disciplines were often co-authors on research describing the resulting systems. However, as the base of applications broadened, it became apparent that interviews with experts, designed to try and elicit the appropriate knowledge was generally the most frequently used approach. Prerau (25) notes the importance of getting step-by-step detail, and that using some form of "quasi-English if-then rules" to document the findings. However, there have been a number of other creative approaches for gathering knowledge from experts, including the following.

ExperTAX

One particularly innovative approach was used by Coopers & Lybrand in the development of their expert system "ExperTAX" (e.g., Shpilberg et al. (26) and Kneale (16)) The goal of the project was to try and understand how partners in a professional services firm analyzed tax planning problems. Ultimately, in order to gather the knowledge necessary to solve a particular tax problem, they had a team of three partners behind a curtain. On the other side of the curtain was a beginner, with a large number of documents. While videoing the process, the partners guided the beginner toward a solution. The camera captured what questions were asked, what documents were needed and what information was used. Ultimately, each of the partners spent a total of over 50 hours working on the system.

Problems with Gathering Knowledge from Experts

Various problems have been reported associated with gathering knowledge from experts. First, knowledge is power. As a result, unfortunately, experts do not always have incentives to cooperate. For example, one consultant noted in Orlikowski (27, p. 246), as to why expert consultants at one company were not interested in participating in knowledge acquisition, "Power in this firm is your client base and technical ability …. It is definitely a function of consulting firms. Now if you put all of this in a … database, you will lose power. There will be nothing that's privy to you, so you will lose power. It's important that I am selling something that no one else has. When I hear people talk about the importance of sharing expertise in the firm, I say, 'Reality is a nice construct.'" As a result, it has been suggested that experts may withhold secrets (e.g., (2)).

Second, as noted in Rose (2), oftentimes experts do not consciously understand what they do. As a result, any attempt to interview them will not result in the quality or quantity of knowledge that is necessary for a system to work. In an example discussed in Rose (2), SCE had their programmers study dam safety and construction engineering reading some before the knowledge acquisition. Then the programmers met one on one with the expert in a windowless conference room. Their first meeting lasted seven hours. They captured all of the interaction using a tape recorder. Unfortunately, the attempts to build the system ran into difficulties. Early versions of the program indicated problems. Virtually every scenario ended with the recommendation to pack the problem wet area with gravel and keep it under observation. They narrowed the focus to a single dam in an effort to generate sufficient detail and insights. However, even after months of work, the knowledge base had only twenty different rules.

KNOWLEDGE REPRESENTATION

There are a number of forms of knowledge representation in artificial intelligence. However, expert systems typically refer to so-called rule-based systems. However, there have been some extensions to deterministic rules to account for uncertainty and ambiguity.

"If … then …" Rules

"If … then …" rules are the primary type of knowledge used in classic expert systems. As noted above those rules are used to capture heuristic reasoning that experts apparently often employ. However, over time researchers began to develop and integrate alternative forms of knowledge representation, such as frame-based or case-based reasoning, into their systems. Systems that included multiple types of knowledge sometimes were referred to as hybrid systems, or labeled after a particular type of knowledge representation, e.g., case-based.

Uncertain Knowledge

Unfortunately, not all statements of knowledge are with complete certainty. One approach to capturing uncertainty of knowledge was to use some form of probability on each of the rules. As expert systems were developed, a number of different approaches were generated, oftentimes depending on the particular application. For example, Buchanan and Shortliffe (12, p. 248) for rules of the sort "if e then h," generated certainty factors (CF) for a medical expert system. MYCIN attributes a "meaning" to different certainty factors (Buchanan and Shortliffe (12), p. 91). The larger the weight, the greater the belief in the specific rule. If CF = 1.0 then the hypothesis is "known to be correct." If CF = -1.0 then that means that the hypothesis ". . . has been effectively disproven." "When CF = 0 then there is either no evidence regarding the hypothesis or the supporting evidence is equally balanced by evidence suggesting that the hypothesis is not true."

Duda et al. (20) and Hart et al. (15) developed a different approach for Prospector, an expert system designed to aid geological exploration. They used the specification of “if E then H (to degree S,N).” S and N are numeric values that represent the strength of association between E and H. S is called a sufficiency factor, since a large S means that a high probability for E, is sufficient to produce a high probability of H, N is called a necessity factor, since a small value of N means that a high probability for E is necessary to produce a high probability of H, where $S = P(E|H)/P(E|H')$ and $N = P(E'|H)/P(E'|H')$. S and N are likelihood ratios. This approach was extended to include the reliability of the evidence (e.g., (28)).

In addition to probability-based approaches additional approaches emerged and found their way into expert systems. For example, fuzzy sets (Zadeh (29)) and Dempster-Shafer belief functions (Shafer (30)) were used to provide alternative approaches.

Interaction of Knowledge Acquisition and Representation

Unfortunately, it does not appear that knowledge acquisition and representation are independent of each other. For example, recent research (31) illustrates that the two are tightly intertwined. An empirical analysis of logically equivalent, but different knowledge representations, can result in different knowledge being gathered. That is, soliciting knowledge in one knowledge representation can generate knowledge perceived as different than a logically equivalent one. As a result, if the developer wants “if … then …” rules then they should use those rules as the form of knowledge in the acquisition process, and throughout system development.

EXPLANATION

Researchers were able to develop techniques, so that given complex rule bases or other structured forms of knowledge representation, systems could analyze the knowledge to find a solution. However, a human user of the system might look at the systems and not understand “why” that particular solution was chosen. As a result, it became important for systems to be able to provide an explanation as to why they chose the solution that they chose.

Importance of Explanation Facilities

Arnold et al. (32) did an empirical analysis of the use of an explanation facility. They found that novice and expert users, employed the explanation capabilities differently. In addition, they also found that users were more likely to follow a recommendation if there was an explanation capability. As a result, explanation is an important strand of expert system research, that includes the following.

Trace Through the Rules

Perhaps the first approach toward generating a system that could provide an explanation for the choice was to generate a trace of the rules. The trace was simply a listing of which rules were executed in generating the solution. Much of the research on explanation leveraged knowledge and context from the specific application area. Although primitive, this approach still provided more insight into why a decision was made, as compared to probability or optimization approaches.

Model-based Reasoning

In general, explanation is facilitated by the existence of a model that can be used to illustrate why a question is being asked or why a conclusion was drawn. One model-based domain that has gathered a lot of attention is the financial model of a company that depends on a number of accounting relationships. This financial model has been investigated by a number of researchers as a basis of explaining decisions (e.g., (33)).

Dialog-Based Systems

Quilici (34) had an interesting approach to explanation, suggesting that in the long-run that expert systems must be able to participate in dialogs with their users. Quilici suggested that providing a trace was not likely to be enough, but instead the system needed to know when and how to convince a user. This would require that the system understand why its advice was not being accepted.

Explanation as to what Decisions were Made in Building the Program

Swartout (35) argued that as part of explanation, a system needs to be able to explain what its developers did and why. Accordingly, he built XPLAIN to provide the user with insights about decisions made during creation of the program in order to get insight into the knowledge and facilitate explanation.

VERIFICATION AND VALIDATION

As noted above, one of factors that makes a system an expert system, is the level of performance of a system. As a result, perhaps more than any other type of system, verification and validation that some system functions at a particular level of expertise is important in establishing the basic nature of the system. Accordingly, an important set of issues is ensuring that the system developed works appropriately and that the knowledge contained in the system is correct. Assuring those conditions is done using verification and validation.

Verification

Verification is more concerned with the syntactical issues. As noted by O'Keefe et al. (36), verification refers to building the system right. Verification refers to making sure that the technology has been correctly implemented. Accordingly, verification is concerned that the structural nature of the "if … then …" rules is appropriate. For example, verification is concerned that there are no loops in the rule base ("if a then b" and "if b then a") or that there are no rules that conflict (e.g., "if a then b," "if a then c"). Preece et al. (37) examine these structural issues in greater detail. Verification also is concerned that any weights on rules have been done correctly. For example, O'Leary (38) provides a number of approaches to help determine if expert system weights on the rules have been put together appropriately or if there are any anomalies that should be investigated.

Validation

Validation is more concerned with the semantic issues. As noted by O'Keefe et al. (36) validation refers to building the right system. O'Leary (39) lays out some of the critical issues regarding validation of expert systems and ties his approach to a structure based on research methods. O'Leary (39) suggests that some of the key functions of validation, all consistent with the nature of expert systems, are

- ascertaining what the system knows, does not know or knows incorrectly
- ascertaining the level of decision making expertise of the system
- analyzes the reliability of the system.

While (39) is concerned with the theory and basic guidelines, O'Leary (40) provides a number of practical methods for expert system validation.

EXPERT SYSTEM STRENGTHS AND LIMITATIONS

Unfortunately, the mere term "expert" has put much pressure that the system performs at an appropriate level. This label is both a strength and a weakness. This section lists some other of the strengths and limitations of expert systems.

Strengths

Expert systems have provided the ability to solve real problems using the manipulation of syntactic and semantic information, rather than quantified information, providing a major change in the view as to what computer could do. In particular, if the problem being posed to the system is one for which rule based knowledge is effective, then the system is likely to be able to provide a recommended solution.

Further, expert systems can be integrated with other computer-based capabilities. As a result, they can do substantial “pre-analysis” of the data. For example, in the case of financial systems, financial ratios can be computed and analyzed, saving much time and effort.

Limitations

However, there are also some limitations associated with expert systems. One of the biggest “complaints” against expert system has been the extent to which they are limited in scope and that the systems do not know their limitations. Classic expert systems have rules that focus only on the problems that it is designed to solve, resulting in their limited scope. Generally expert systems do not know when a problem being posed by the user is outside of scope of the system.

Further, as noted by (4), from a practical perspective, expert systems “… require complex and subtle interactions between machines and humans, each teaching and learning from other.” Rather than being static, systems and people need to learn and change to accommodate each other.

In addition, Clancy (10) was an early investigator who noted that people, other than the authors of the rules may have difficulty modifying the rule set. Clancy (10) also had concerns with the basic rule formalism for capturing knowledge. For example, Clancy noted “ … the view that expert knowledge can be encoded as a uniform … set of if/then associations is found to be wanting.”

Getting and keeping up-to-date knowledge is another potential limitation. For example, in the area of United States taxation, the tax rules change every year. Some rules are new and some rules are no longer valid. Such rule-base changes are not unusual in any setting where technology is involved that must change often more than once a year. For example, imagine developing a system to help someone choose the right mobile phone.

Finally, a primary limitation to expert systems is illustrated by comment from Mike Ditka, a hall of fame American Football player. On a radio interview on Los Angeles Area radio (September 18, 2007), while talking about evaluating football players, he noted "… the intangibles are more important than the tangibles." Viewed from the perspective of expert systems, this suggests that although we can capture (tangible) knowledge, that other (intangible) knowledge is out there, but not captured, and in many cases that additional knowledge may be the most important.

EXTENSIONS TO EXPERT SYSTEMS AND EMERGING RESEARCH ISSUES

The basic model of the expert system presented to this point is one where knowledge is gathered from a single expert and that knowledge is categorized as "if … then …" rules, as a basis for mapping expertise into a computer program. However, there have been some extensions to that basic model, including the following.

Multiple Experts or Knowledge Bases

Ng and Abramson (41) discussed a medical system named "Pathfinder" that was designed around multiple experts. Rather than having the system designers try to merge knowledge gathered from multiple experts into a single knowledge base, the design concept was to allow the system to put together knowledge from the multiple experts, when it needed it.

Knowledge from Data

Gathering knowledge from experts ultimately became known as a "bottle neck." In some cases data was available, so rather than capturing what people said they did, an analysis of the data found what they actually did. Some researchers began to try to get knowledge from data, rather than going through classic interview processes. Ultimately, the focus on generating knowledge from data ended up creating the notion and field of knowledge discovery.

Neural nets also provided a vehicle to capture knowledge about data. Ultimately, neural nets have been used to create rules that are used in expert systems and expert systems have been built to try to explain rules generated from neural networks.

Alternative Forms of Knowledge Representation

As researchers studied reasoning and built systems they found that rules apparently were not the only way that people thought, or the ways that the researchers could represent knowledge. For example, one line of reasoning suggested that people used cases or examples on which to base their reasoning. As another example, researchers built frame-based reasoning systems. Frames allow researchers to capture patterns, that allow heuristic matching, e.g., as was done with GRUNDY (Rich 42). As a result, case-based reasoning and other forms of knowledge representation helped push researchers to forms of knowledge representation beyond rules in an attempt to match the way that people use knowledge (Hayes 43).

Alternative Problem Solving Approaches

Not only knowledge representation changed, even other types of problem solving approaches were used. For example, as noted by Shortliffe (1, p. 831), "The term (expert systems) has subsequently been broadened as the field has been popularized, so that an expert system's roots in artificial intelligence research can no longer be presumed … any decision support system (is) an expert system if it is designed to give expert level problem specific advice …"

Expertise

Since expert systems were intent on capturing human expertise in a computer program, this led to a need to better understand expertise and what it meant to be an expert. As a result, since the introduction of expert systems there has been substantial additional research in the concept of expertise, not just how expertise can be mapped into a computer program..

Uncertainty Representation

Generating expert systems for different domains ended up facilitating the development of a number of approaches for representing uncertainty. However, additional research has focused on moving toward Bayes' Nets and influence diagrams (e.g., Pearl 44) and moving away from the MYCIN certainty factors and the Prospector likelihood ratios.

The Internet and Connecting Systems

Generally, the expert system wave came before the Internet. As a result, the focus was on systems for a specific computer, and not networked computers. As a result, there was limited research about networks of expert systems. However, since the advent of the Internet, expert system concepts were extended to knowledge servers (e.g., 45) and multiple intelligent agents. In addition, technologies such as extensible mark-up language (xml) are now used to capture information containing rules and data, and communicate it around the world (e.g., xpertrule.com).

Ontologies

Further, developers found that as expert systems grew or were connected and integrated with other systems that more formal variable definition was necessary. Large variable sets needed to be controlled and carefully managed, particularly in multilingual environments. As a result, extending those expert system capabilities led to some of the work on ontologies.

Embedded Intelligence vs. Stand Alone Systems

Increasingly, rather than highly visible stand alone applications, rule-based intelligence was built into other production applications. Because the systems were not stand alone expert systems, users did not even "see" the embedded expertise: People don't go check on what the expert system has to say – programs now are just more intelligent. For example, fixing spelling errors and grammar errors in Word, requires a certain amount of intelligence.

Business Rules

As another form of evolution, there is now interest by businesses in so-called "business rules." As might be anticipated, business rules assume that businesses use rules in their interaction with other businesses. Rather than wait for people to make decisions business rules capture those decision making capabilities. Business rules have virtually all of the same concerns as we saw in expert system rules, in terms of knowledge acquisition, knowledge representation, verification and validation, etc.

CONCLUSION

Expert systems have provided an important starting point for understanding and mimicking human expertise. However, they were only a start. Expert systems focused on heuristic decision making and rules, generally as manifested in "if – then" rules, possibly employing weights on the rules to capture uncertainty or ambiguity. Expert system provided the foundations on which many other developments have been made.

BIBLIOGRAPHY

E. Shortliffe, Medical Expert Systems, Knowledge Tools for Physicians, *The Western*
1 *Journal of Medicine,* December 1986, 830-839.

F. Rose, An 'Electronic' Clone of a Skilled Engineer is very hard to create, *The Wall*
2 *Street Journal*, August 12, 1988.

D. Bobrow, S. Mittal, and M. Stefik, Expert Systems: Perils and Promise,
3 *Communications of the ACM*, Volume 29, Number 9, 880-894, September 1986.

4 Business Week, The New Rocket Science, *Business Week,* November, 1992.

F. Hayes-Roth, The Knowledge-based Expert System, *IEEE Computer,* September 1984,
5 11-28.

6 R. Davis, Knowledge-based Systems, *Science*, 231: 4741, February 28, 1986.

V. Barker and D. O'Connor, Expert Systems for Configuration at Digital: XCON and
7 Beyond, Communications of the ACM, March 1989, Volume 32, Number 3, 98 - 318.

8 D. Lyons, Artificial Intelligence gets Real, *Forbes*, November 30, 1998.

9 H. A. Simon, *Administrative Behavior,* Second Edition, The Free Press, 1965.

W. J. Clancey, The Epistemology of a Rule-based Expert System, Stanford CS – 81-896,
10 1981.

11 M. Richter, AI Tools and Techniques, Norwood NJ : Ablex Publishing, 1989.

B. Buchanan and E. Shortliffe, *Rule Based Expert Systems: The Mycin Experiments of*
12 *the Stanford Heuristic Programming Project***,** Addison – Wesley, 1984.

13 Cimflex Teknowledge, *M4 User's guide*, Cimflex Teknowledge, 1991.

B. Buchanan and E. Feigenbaum, Dentral and Meta-Dentral: Their Applications
14 Dimension, Heuristic Programming Project Memo, 78-1, February 1978.

P. Hart, R. Duda and M. Einaudi, PROSPECTOR-A computer-based Consultation
15 System for Mineral Exploration, *Mathematical Geology*, Volume 10, Number 5, 1978.

D. Kneale, How Coopers & Lybrand Put Expertise into its Computers, *The Wall Street*
16 *Journal*, November 14, 1986.

W. J. Clancey and E.H. Shortliffe (Eds): *Readings in Medical Artificial Intelligence:The*
17 *First Decade. Reading, Mass. Addison-Wesley, 1984*

P. Szolovits, Uncertainty and Decisions in Medical Informatics, *Methods of Information*
18 *in Medicine*, 34: 111-134, 1995.

R. McCammon, Prospector II: Towards a Knowledge Base for Mineral Deposits,
19 *Mathematical Geology,* Volume 26, Number 8: 917-937, 1994.

R. Duda, J. Gaschnig. and P. Hart, "Model Design in the Prospector Consultant System
for Mineral Exploration," in D. Mitchie, Ed., *Expert Systems for the Micro Electronic*
20 *Age*, 153-167, Edinburgh University Press, Edinburgh, 1979.

T. J. Beckman, AI in the IRS, *Proceedings of the AI Systems in Government Conference*,
21 1989.

22 C. Brown, A.A. Baldwin and A. Sangster, "Accounting and Auditing," in *The Handbook of Applied Expert Systems*, Edited by J. Liebowitz, CRC Press, pp. 27-1 - 27-12, 1998.

23 D. Prerau, Selection of an Appropriate Domain for an Expert System, AI Magazine, Volume 6, Number 2, Summer 1985, 26-30.

24 M. Myers, A. Detore, S. Siegel, K. Curley, The Strategic Use of Expert Systems for Risk Management in the Insurance Industry, ACM, 1990.

25 D. Prerau, Knowledge Acquisition in the Development of a Large Expert System, *AI Magazine*, Volume 8, Number 2, 43-51, Summer 1987.

26 D. Shpilberg and L. Graham and H. Schatz, Expertax: An Expert System for Corporate Tax Accrual and Planning,"*Expert Systems*, 1986.

27 W. Orlikowski, Learning from Notes, *The Information Society*, 9: 237-250, 1993.

28 D. O'Leary, On the Representation and Impact of Reliability of Expert System Weights, *International Journal of Man-Machine Studies*, 29: 637-646, 1988.

29 L. Zadeh, Fuzzy Sets, *Information and Control* 8: 338–353, 1965.

30 G. Shafer *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ, 1976.

31 D. O'Leary, Knowledge Representation of Rules, *Intelligent Systems in Accounting, Finance and Management*, Volume 15, Number 1-2, 73-84, 2007.

32 V. Arnold, N. Clark, P. Collier, S. Leech, and S. Sutton, The Differential Use and Effect of Knowledge based System Explanations in Novice and Expert Judgment Decisions, *MIS Quarterly*, 30: 79-97, 2006.

33 W. Hamscher, Explaining Financial Results, International Journal of Intelligent Systems in Accounting, Finance and Management, 3:1-20, 1994,

34 A. Quilici, Recognizing and Revising Unconvincing Explanations, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 3:21-34, 1994.

35 W. Swartout, XPLAIN: A System for Creating and Explaining Expert Consulting Programs, *Artificial Intelligence*, 40: 353-385, 1989.

36 R. O'Keefe, O. Balci, E. Smith, Validating Expert System Performance, *IEEE Expert*, Volume 2, Number 4, 81-90, 1987.

37 A. Preece and R. Shinghal, Foundation and Application of Knowledge Base Verification, International Journal of Intelligent Systems, 9: 683-702, 1994.

38 D. O'Leary, Verification of Uncertain Knowledge-based Systems, *Management Science*, 42: 1663-1675, 1996.

39 D. O'Leary, Validation of Expert Systems, *Decision Sciences*, Volume 18, Number 3, 468-486, 1987.

40 D. O'Leary, Methods of Validating Expert Systems, *Interfaces*, Volume 18, Number 6, 72-79, 1988.

41 K. Ng and B. Abramson, Probabilistic Multi-knowledge base systems, *Journal of Applied Intelligence*, Volume 4, Number 2, 219-236, 1994.

42 E. Rich, "User Modeling via Stereotypes," *Cognitive Science*, 3, pp. 329-354, 1989.

43 P. Hayes, "The logic of frames," In D. Metzing (editor), *Frame Conceptions and Text Understanding,* pages 45-61, de Gruyter, 1979.

J. Pearl, *Probabilistic Reasoning in Intelligent Systems,* Morgan Kaufman, San Mateo,
44 CA, 1989.

N. Abernethy, J. Wu, M. Hewitt and R. Altman, Sophia: A flexible Web-based
45 Knowledge Server, *IEEE Intelligent Systems*, 79-85, July/August 1999.

READING LIST

B. Buchanan and E. Shortliffe, *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*, Addison – Wesley, 1984.

F. Hayes-Roth, D. Waterman and D. Lenat, *Building Expert Systems*, Addison – Wesley, 1983.

J. Liebowitz, The Handbook of Applied Expert Systems, CRC, 1997.

Shu-Hsien Liao, Expert system methodologies and applications—a decade review from 1995 to 2004, *Expert Systems with Applications*, January 2005, Volume 28, Number 1, pp. 93-103.