Report 81-09
Stanford -- KSL

Scientific DataLink

Evaluating Expert Systems.
Edward H. Shortliffe,
Jul 1981

#64

HPP-81-9

EVALUATING EXPERT SYSTEMS

Edward H. Shortliffe

Heuristic Programming Project

Departments of Medicine and Computer Science

Stanford University

Stanford, California 94305

July 1981

This paper is the author's contribution to Chapter 6 in the volume BUILDING EXPERT SYSTEMS, edited by R. Hayes-Roth, D. Lenat, and D. Waterman.* The full article is entitled "Evaluation of expert systems: issues and case studies", and is authored by J. Gaschnig, P. Klahr, H. Pople, E. Shortliffe, and A. Terry. The volume is the result of a Workshop on Expert Systems held in San Diego in August 1980 and sponsored by the Rand Corporation, ARPA, and the NSF.

Hayes-Roth, Waterman & Lenat, BUILDING EXPERT SYSTEMS, Copyright © 1983, Addison-Wesley, Reading, Massachusetts. Parts of Chapters 7 & 8. Reprinted with permission.

1 Issues in the Evaluation of Expert Systems

We have been discussing the reasons for doing evaluations of expert systems, or for having reservations about getting involved in the evaluation process, but we have not addressed the nature of the evaluation process itself. In this section we define many of the parameters that determine an appropriate design for an evaluation experiment.

1.1 Dependence on Task, System, Goals, and Stage of Development

When one examines the literature on computer performance evaluation, it is clear that the term is used with a variety of meanings depending upon the perspective of the authors. Each tends to focus on the specific performance issues that have been most central to the design of the system in question. Other aspects warranting formal evaluation are often ignored. One of the goals of this chapter is to make it clear that there are diverse components to the evaluation process; validation is most appropriately seen as occurring in stages as an expert system develops over time.

Most computing systems are developed in response to some human need, and it might therefore be logical to emphasize the system's response to that need in assessing whether it is successful. Thus there are those who would argue that the primary focus of a system evaluation should be the task for which it was designed and the quality of its corresponding performance.

It must be recognized, however, that the goals of system builders are not necessarily directed at the long term utility of their system. Many

expert system researchers are responding to challenges related to the basic science of intelligent computing, and the task domain they choose may well be selected primarily because it allows them to exercise their developing techniques optimally. In this case, it may be inappropriate to assess the expert system's success purely on the basis of its level of performance. The basic science insights which have resulted, and the concomitant stimulation of new research avenues, may justify a piece of work in expert systems research and provide a pertinent "evaluation" of its success given the designer's goals.

In the remainder of this chapter, however, we will be emphasizing the evaluation of systems which are ultimately designed to perform a real world task, typically to be used by a community of non-computer scientists. Our assumption is that one major goal is the development of a useful system that can have an impact on society by becoming a regularly used tool in the community for which it was designed. Although we recognize that many basic science problems may arise during the development of such systems, our discussion will emphasize the staged assessment of the developing tool, and not techniques for measuring its scientific impact as a stimulus to further research.

We shall organize our discussion by looking at the "what?", "when?", and "how?" of expert systems evaluation. When possible, we shall draw on examples from some of the few expert systems that have actually undergone formal evaluations. These programs have provided important insights into ways of tackling the unusual problems inherent in assessing validity and

acceptability of an expert system's performance. They accordingly provide useful guidelines for designing the evaluations of the expert systems of tomorrow.

1.2 What to Evaluate?

As mentioned above, at any stage in the development of a computing system several aspects of its performance could be evaluated. Typically some are more appropriate than others at a particular stage. However, by the time a system has reached completion it is likely that every aspect will have warranted formal assessment. The timing for such evaluations will be discussed in section 1.3.

1.2.1 Decisions/Advice/Performance

Expert systems developed to date have tended to emphasize the program's performance at its decision making task in their evaluation studies. Since accurate reliable advice is an essential component of an expert consultation system, it is usually the area of greatest research interest and is logically an area to emphasize in evaluation. As we shall point out below, however, the mechanisms for deciding whether a system's advice is appropriate or adequate may be difficult to define or defend. Expert systems tend to be built precisely for those domains in which the decisions of human experts are highly judgmental and non-standardized. However, it is clear that no expert system will be accepted by its intended users if they fail to be convinced that the decisions made and the advice

given are pertinent and reliable. Thus, some approach to performance verification is typically mandatory.

1.2.2 Correct Reasoning

Not all designers of expert systems are concerned about whether their program reaches decisions in a "correct" way, so long as the advice that it offers is appropriate. However, as has been discussed earlier in this book, there is an increasing realization that expert level performance may require heightened attention to the mechanisms by which human experts actually solve the problems for which the expert systems are typically built. It is with regard to this issue that the interface between knowledge engineering and psychology is the greatest, and, depending upon the motivation of the system designers and the eventual users of the expert program, some attention to the mechanisms of reasoning that the program uses may be appropriate during the evaluation process. The issue of deciding whether or not the reasoning used by the program is "correct" will be discussed further below.

1.2.3 Discourse (I/O Content)

Although the reliability of the reasoning processes of an expert system is crucial for ultimate success, knowledge engineers now routinely accept that there are a variety of other parameters that will play major roles in determining whether their system is accepted by the intended users. The nature of the discourse between the expert system and the user is particularly important. Here we mean such diverse issues as: (1) the choice

of words used in the questions and responses generated by the program; (2) the ability of the expert system to explain the basis for its decisions and to customize those explanations appropriately for the level of expertise of the user; (3) the ability of the system to assist the user when he is confused by what is required of him or needs assistance for any other reason when using the program; and (4) the ability of the expert system to give advice and/or to educate the user in a congenial fashion so that the frequently cited psychological barriers to computer use are overcome or avoided. It is likely that issues such as these are every bit as important to the ultimate success of an expert system as is the quality of its advice. For this reason, such issues also warrant formal evaluation. Many current expert systems have made some effort to develop capabilities along these lines, but the techniques for assessing their utility, and for separating one variable from the others in a study design, are still rudimentary.

1.2.4 Hardware Environment (I/O Medium)

Much effort has gone into the development of congenial terminal interfaces between novice computer users and computer systems. Although some users, particularly when pressed to do so, can become comfortable with a conventional typewriter keyboard as the basis for the interaction with the machine, in many settings this is a skill that would have to be learned and the potential users are not motivated to do so. For that reason we have seen the development of light pen interfaces, touch screens, and specialized keypads, any of which may be adequate to facilitate a simple interaction between intended users and the expert system. Typically the details of the

hardware interface will influence the design of the system software as well. The intricacies of this interaction cannot be ignored in system evaluation, nor can the mundane details of the user's reaction to the terminal interface. Once again, it can be difficult to design evaluations in which dissatisfaction with the terminal interface is isolated as a variable, independent of discourse adequacy or decision making performance. As we shall point out below, one purpose of staged evaluations is the resulting ability to allow certain variables to be eliminated during the evolution of the system.

1.2.5 Efficiency

The impact of an expert system on the "process" of decision making in the user's environment must also be analyzed during the system's evaluation. A system that requires an excessive time commitment by the user, for example, may fail to be accepted even if it excels at all the other tasks we have mentioned. Similarly, technical analyses of system behavior are generally warranted. Underutilized CPU power or poorly designed disk-seeking behavior, for example, may introduce resource inefficiencies that severely limit the system's response time or cost effectiveness.

1.2.6 Cost Effectiveness

Finally, and particularly if it is intended that an expert system become a marketable product, some detailed evaluation of its cost effectiveness is necessary. No AI systems have reached this stage in system

evolution, but there is a wealth of relevant experience in other computer science areas. Expert systems must be prepared to embark on similar studies once they reach an appropriate stage of development.

1.3 When to Evaluate?

The evaluation process is a continual one that should begin at the time of system design, extend in an informal fashion through the early stages of development, and become increasingly formal as a developing expert system begins to achieve real world implementation. It is useful to cite nine stages of system development which summarize the evolution of an expert system[1]. They are itemized in Table 1 and discussed in some detail in the paragraphs below.

[1] These implementation steps are based on a discussion of expert systems by Shortliffe and Davis in the SIGART newsletter, No. 55, December 1975, pp. 9-12.

Table 1

Steps In The Implementation Of An Expert System

1. Top level design; define long-range goals
2. First version prototype, showing feasibility
3. System refinement in which informal test cases are run to generate feedback from the expert and from users
4. Structured evaluation of performance
5. Structured evaluation of acceptability to users
6. Service functioning for extended period in prototype environment
7. Follow-up studies to demonstrate the system's large scale usefulness
8. Program changes to allow wide distribution of the system
9. General release and marketing with firm plans for maintenance and up-dating.

As was mentioned above, it is important for system designers to be very explicit about the nature of their motivations for building an expert system. The long-range goals must also be outlined clearly. Thus the first stage of a system's development (Step 1), the initial design, should be accompanied by explicit statements of what the measures of the program's success will be and how that failure or success will be evaluated. It is not uncommon for system designers to ignore this issue at the outset because of the overwhelming complexity of the initial challenges which their expert system will have to overcome. If the evaluation stages and long range goals are explicitly stated, however, they will necessarily impact on the early design of the expert system. For example, if formal explanation capabilities are deemed to be crucial for the user community in question, this will have important implications for the underlying knowledge representation that the

expert system must utilize. Thus the evaluation process ideally starts with the birth of the expert system so that its anticipation may help shape the early design and uncover issues that it would otherwise be tempting to overlook or ignore.

The next stage in the development of an expert system (Step 2) is a demonstration that the performance task which has been selected is feasible. At this stage there is no attempt to demonstrate expert level performance. The goal is, rather, to show that there is a representation scheme appropriate for the task domain and that knowledge engineering techniques can lead to a prototype system which shows some reasonable (if not expert) performance on some sub-task of that domain. An evaluation of this stage will typically be very informal, and will simply consist of showing that a few special cases can be handled by the prototype system. This result suggests that with increased knowledge, and refinement of the reasoning structures, a high performance expert system is possible.

The next stage (Step 3) is familiar to all knowledge engineers; in fact, it is as far as many systems ever get. This is the period in which informal test cases are run through the developing system, the system's performance is observed, and feedback is sought from expert collaborators and potential end users. This feedback serves to define the major problem areas in the system's development and guides the next iteration in the research endeavor. This iterative process may go on for months to years, depending on the complexity of the knowledge domain, the flexibility of the knowledge representation, and the availability of techniques adequate to cope with the

domain's specific control or strategic processes. The point can be made, however, that evaluation of an informal nature is part of this iteration. The question constantly being asked is: how did this system do on this case? Detailed analyses of strengths and weaknesses lead back to further research; in this sense evaluation is an intrinsic part of the system development process.

Once the system is performing well on most cases with which it is presented, it is appropriate to turn to a more structured evaluation of its decision making performance. This evaluation can be performed without assessing the program's actual utility in a potential user's environment. Thus Step 4 is undertaken if the test cases being used in Step 3 are found to be handled with skill and competence, and there accordingly develops a belief that a formal randomized study will show that the system is capable of handling almost any problem from its domain of expertise. Only a few expert systems have reached this stage of evaluation. The principal examples are the PROSPECTOR Program developed at SRI International [1], and the MYCIN System from Stanford University Medical School [4],[5]. It should be emphasized that a formal evaluation with randomized case selection may show that the expert system is in fact not performing at an expert level. In this case, new research problems or knowledge requirements are defined and the system development returns to Stage 3 for additional refinement. A successful evaluation at Stage 4 is generally required before a program is introduced into a user environment.

The fifth stage (Step 5), then, is system evaluation in the setting

where the intended users have access to it. The principal question at this stage is whether the program is acceptable to the users for whom it was intended. Essentially no expert systems have been assessed at this stage. The emphasis in Step 5 is on the discourse abilities of the program, plus the hardware environment that is provided. That is the reason that Step 4 must have been successfully completed before Step 5 can be attempted. Otherwise, failure to accept the program by the end-users may result from decision making errors rather than problems with the discourse or the hardware environment. If, on the other hand, expert level performance has been demonstrated at Step 4, failure of the program to be accepted at Step 5 can be assumed to be due to one of these other human factors.

If a system is formally shown to make good decisions and to be acceptable to users, it is appropriate to introduce it for extended periods in some prototype environment (Step 6). This stage is intended largely to gain experience with a large number of test cases and with all the intricacies of real-world functioning that may not have been adequately addressed in system design. Careful attention during this stage must be directed towards problems of scale, i.e., what new difficulties will arise when the system is made available to large numbers of users outside of the direct control of the system developers? The evaluation tends to be rather informal and based upon careful observation of the program's performance and the changing attitudes of those who interact with it.

After service functioning has proceeded in a prototype environment and seems to be running smoothly, it is appropriate to begin some follow-up

studies to demonstrate the system's large scale usefulness (Step 7). These formal evaluations tend to require the measurement of pertinent parameters prior to introducing the system into a large user community (different from the original prototype environment). Then, after the system is made available in the new setting, careful observation and measurement is required to determine the system's impact. Pertinent issues are the system's efficiency, its cost effectiveness, its acceptability to users who were not involved in its early experimental development, and its impact on the execution of the task with which it was designed to assist.

During Step 7 it is not uncommon to discover new problems that require attention before the system can be marketed (Step 8). These may involve programming changes or modifications required to allow the system to run on a smaller or exportable machine.

Finally, the last stage in system development is its general release as a marketable product (Step 9). Inherent at this stage are firm plans for maintaining the knowledge base and keeping it current. One might argue that the ultimate evaluation takes place at this stage when it is determined whether or not the system can succeed in the open marketplace. However, a system's credibility is likely to be greater if good studies have been done in the first eight stages so that there are solid data supporting any claims about the quality of the program's performance.

1.4 How to Evaluate?

It would be folly to claim that we can begin to suggest detailed study designs for expert systems in one section of a book chapter. There is a wealth of information in the statistical literature, for example, regarding the design of randomized controlled trials (RCT's), and much of that experience is relevant to the design of expert system evaluations. Our intention here, therefore, is to concentrate on those issues that complicate the evaluation of expert systems in particular and to suggest pitfalls that must be considered during study design.

We also wish to distinguish between two senses of the term "evaluation". In computer science, system evaluation often is meant to imply optimization in the technical sense -- timing studies, for example. Our emphasis, on the other hand, is on a system's performance at the specific consultation task for which it has been designed. Unlike many conventional programs, expert systems do not deal with deterministic problems for which there is clearly a right or wrong answer. As a result, it is often not possible to demonstrate in a straightforward fashion that a system is "correct" and then to concentrate one's effort on demonstrating that it reaches the solution to a problem in some optimal way.

1.4.1 Need for an Objective Standard

Evaluations of new techniques typically require some kind of "gold standard" -- a generally accepted correct answer with which the results of the new methodology can be compared. In the assessment of new diagnostic techniques in medicine, for example, the gold standard is often the result of

an invasive procedure which physicians hope to be able to avoid, even though it may be 100% accurate (e.g., operative or autopsy results, or the findings on an angiogram). The sensitivity and specificity of a new diagnostic liver test based on a blood sample, for example, can best be assessed by comparing test results with the results of liver biopsies from several patients who also had the blood test; if the blood test is thereby shown to be a good predictor of the results of the liver biopsy, it may be possible to avoid the more invasive procedure in future patients. The parallel in expert system evaluation is obvious; if we can demonstrate that the expert system's advice is comparable to the gold standard for the domain in question, it may no longer be necessary to turn to the gold standard itself if it is less convenient, less available, or more expensive.

3.4.1.1 Can the Task Domain Provide a Standard?

In general there are two views of how to define the gold standard for an expert system's domain: 1) what eventually turns out to be the "correct" answer for a problem, or 2) what does a human expert, presented with the same information as is available to the program, say is the correct answer. It is unfortunately the case that for many kinds of problems with which expert systems are designed to assist, the first of these questions cannot be answered or is irrelevant. Consider, for example, the performance of the MYCIN program [2]. MYCIN's charge is to predict the bacteria causing infection in a patient and accordingly to suggest optimal antibiotic therapy. One might therefore suggest that the gold standard in this domain should be the identity of the bacteria that are ultimately isolated from the patient,

or the patient's outcome if he is treated in accordance with (or in opposition to) the program's recommendation. Suppose, for example, that MYCIN suggests therapy that covers for four possibly pathogenic bacteria but that the organism that is eventually isolated is instead a fifth rare bacterium that was totally unexpected, even by the experts involved in the case. In what sense should MYCIN be considered "wrong" in such an instance? Since expertise in such a domain is largely based on probabilistic considerations, might it not be reasonable to suggest that MYCIN performed at an expert level and was in fact "correct" if it agreed with the experts, even if both MYCIN and the experts turned out to be wrong? Similarly, the outcome of patients treated for serious infections is not 100% correlated with the correctness of therapy; patients treated in accordance with the best available medical practice may still die from fulminant infection, and occasionally patients will improve on their own despite inappropriate or unnecessary antibiotic treatment. Because of considerations such as these, the studies of MYCIN's performance used expert opinions rather than patient outcomes or bacterial identification as the gold standard [4],[5].

1.4.1.2 Are Human Experts Evaluated?

A related issue that arises if domain experts are used as the objective standard for performance evaluation is whether the human experts themselves are subjected to rigorous evaluations of the quality of their decisions. If so, such assessments of human expertise may provide a useful set of benchmarks against which to measure the expertise of a developing consultation system. An advantage of this approach is that the technique for

evaluating human experts is usually a well-accepted basis for assessing expertise and thus lends credibility to an evaluation of the computer-based approach.

1.4.1.3 Informal Standards

Typically, however, human expertise is accepted and acknowledged using less formal criteria (e.g., level of training, recommendations of previous clients, years of experience in a field, number of publications, and the like). Testimonials regarding the performance of a computer program have also frequently been used as a catalyst to the system's dissemination, but it is precisely this kind of anecdotal "selling" of a system against which we are arguing here. Many fields will not accept technological innovation without rigorous demonstration of the breadth and depth of the new product's capabilities. This point may be particularly true in domains in which a computer system is taking on decision making tasks previously performed by human beings. Both MYCIN and PROSPECTOR encountered this cautious attitude in potential users and designed their evaluations largely in response to a perceived need for rigorous demonstrations of performance.

1.4.2 Biasing and Blinding

In designing any evaluation study, considerations of sources of bias are of course important. This lesson was learned again by MYCIN's evaluators and explains many of the differences between the bacteremia evaluation [4] and the meningitis study [5]. During the first of these evaluations, the

expert physicians who were assessing MYCIN's performance knew they were examining the output of a computer program. Many of their comments and criticisms reflected their own biases regarding the proper role for computers in medical settings (e.g., "I don't think the computer has an adequate sense of how sick this patient is. You'd have to see a patient like this in order to judge."). As a result, the meningitis study design mixed MYCIN's recommendations with a set of recommendations from nine other individuals asked to assess the case (ranging from infectious disease faculty members to medical students). When national experts later gave opinions on the appropriateness of therapeutic recommendations, they did not know which proposed therapy was MYCIN's and which recommendation came from the faculty members. This "blinded" study design removed an important source of potential bias, and also provided a sense of where MYCIN's performance lay along a range of expertise from faculty member to medical student.

1.4.3 Remove Variables

As we pointed out in the discussion of "when" to evaluate an expert system, one advantage of a sequential set of studies is that each can assume the results of the experiments that preceded it. Thus, for example, if a system has been shown to reach optimal decisions in its domain of expertise, one can assume that the system's failure to be accepted by its intended users in an experimental setting is a reflection of inadequacies in an aspect of the system other than its decision making performance. One key variable that could account for system failure has been "removed" in this way.

1.4.4 Realistic Standards of Performance

Before assessing the capabilities of an expert system, it is necessary to define the minimal acceptable standards that are acceptable for the system to be called a success. It is ironically the case that in many domains it is difficult to decide what level of performance qualifies as "expert". Thus it is important to measure the performance of human experts in a field if they are assessed by the same standards to be used in the evaluation of the expert system. This point was demonstrated in the MYCIN evaluations. In the bacteremia studies [4], MYCIN's performance was approved by experts in approximately 75% of cases, a figure that seemed disappointingly low to the system developers. They felt that the system should be approved in at least 90% of cases before it was made available for actual clinical use. The blinded study design for the subsequent meningitis evaluation [5], however, showed that even infectious disease faculty members received at best a 70-80% rating from other experts in the field. Thus the 90% figure originally sought may have been unrealistic in that it inadequately reflected the level of disagreement that can exist even among experts in a field such as clinical medicine.

1.4.5 Sensitivity Analysis

A special kind of evaluation procedure that is pertinent for expert systems work is the analysis of a program's sensitivity to slight changes in knowledge representation, inference weighting, etc. Similarly, it may be pertinent to ask which interactive capabilities were necessary for acceptance

of an expert consultant. Experiments that compare two versions of the system, one with and one without (or with a different version) of the feature under consideration provide one approach to assessing these issues. Identical results from two parallel studies tend to suggest that the feature may not be crucial to system performance afterall. An example of studies of this kind are the experiments that MYCIN's developers have done in assessing their certainty factor (CF) model of inexact reasoning [3]. Clancey and Cooper showed, in unpublished experiments, that the decisions of MYCIN changed minimally from those reported in the meningitis evaluation [5] over a wide range of possible CF intervals for the inferences in the system. This "sensitivity analysis" experiment helped MYCIN researchers decide that the details of the CF's associated with their rules mattered less than the semantic and structural content of the rules themselves.

1.4.6 Interaction of Knowledge: Preserving Good Performance When Correcting the Bad

An important problem, discussed earlier in this book as well, can be encountered when an evaluation has revealed system deficiencies and new knowledge has been added to the system in an effort to correct these. In complex expert systems, the interactions of new knowledge with old can be unanticipated and lead to detrimental effects on problems that were once handled very well by the system. An awareness of this potential problem is crucial as system builders iterate from Step 3 to Step 4 and back to Step 3 (see Table 1). One method for protecting against the problem is to keep a library of old cases available on-line for batch testing of the system's

decisions. Then, as changes are made to the system in response to Step 4 evaluations of the program's performance, the old cases can be run through the revised version to verify that no unanticipated knowledge interactions have been introduced (i.e., to show that the program's performance on the old cases does not deteriorate).

1.4.7 Realistic Time Demands on Evaluators

A mundane issue that must still be considered because it can lead to failure of a study design or, at the very least, to unacceptable delays in completing the program's assessment, is the time required for the evaluators to judge the system's performance. If expert judgments are used as the "gold standard" for adequate program performance, the opinions of the experts must be gathered for the cases used in the evaluation study. A design that picks the most pertinent two or three issues to be assessed, and concentrates on obtaining the expert opinions in as easy a manner as possible, will therefore have a much better chance of success. MYCIN's staff experienced over a year delay in obtaining the evaluation booklets back from the experts who had agreed to participate in the bacteremia evaluation [4]; by focussing on fewer variables and designing a checklist that allowed the experts to assess program performance much more rapidly, the meningitis evaluation was completed in less than half that time [5].

References

1. Gaschnig, J. Preliminary performance analysis of the PROSPECTOR consultant system for mineral exploration. Proceedings of the 6th IJCAI, pp. 308-310, Tokyo, Japan, August 1979.

2. Shortliffe, E.H. Computer-Based Medical Consultations: MYCIN. Elsevier/North Holland, New York, 1976.

3. Shortliffe, E.H. and Buchanan, B.G. A model of inexact reasoning in medicine. Math. Biosci. 23:351-379 (1975).

4. Yu, V.L., Buchanan, B.G., Shortliffe, E.H., et al. Evaluating the performance of a computer-based consultant. Comput. Prog. Biomed. 9:95-102 (1979).

5. Yu, V.L., Fagan, L.M., Wraith, S.M., et al. Antimicrobial selection by a computer: a blinded evaluation by infectious disease experts. J. Amer. Med. Assoc. 242:1279-1282 (1979).

Copyright © 1985 by KSL and
Comtex Scientific Corporation

FILMED FROM BEST AVAILABLE COPY