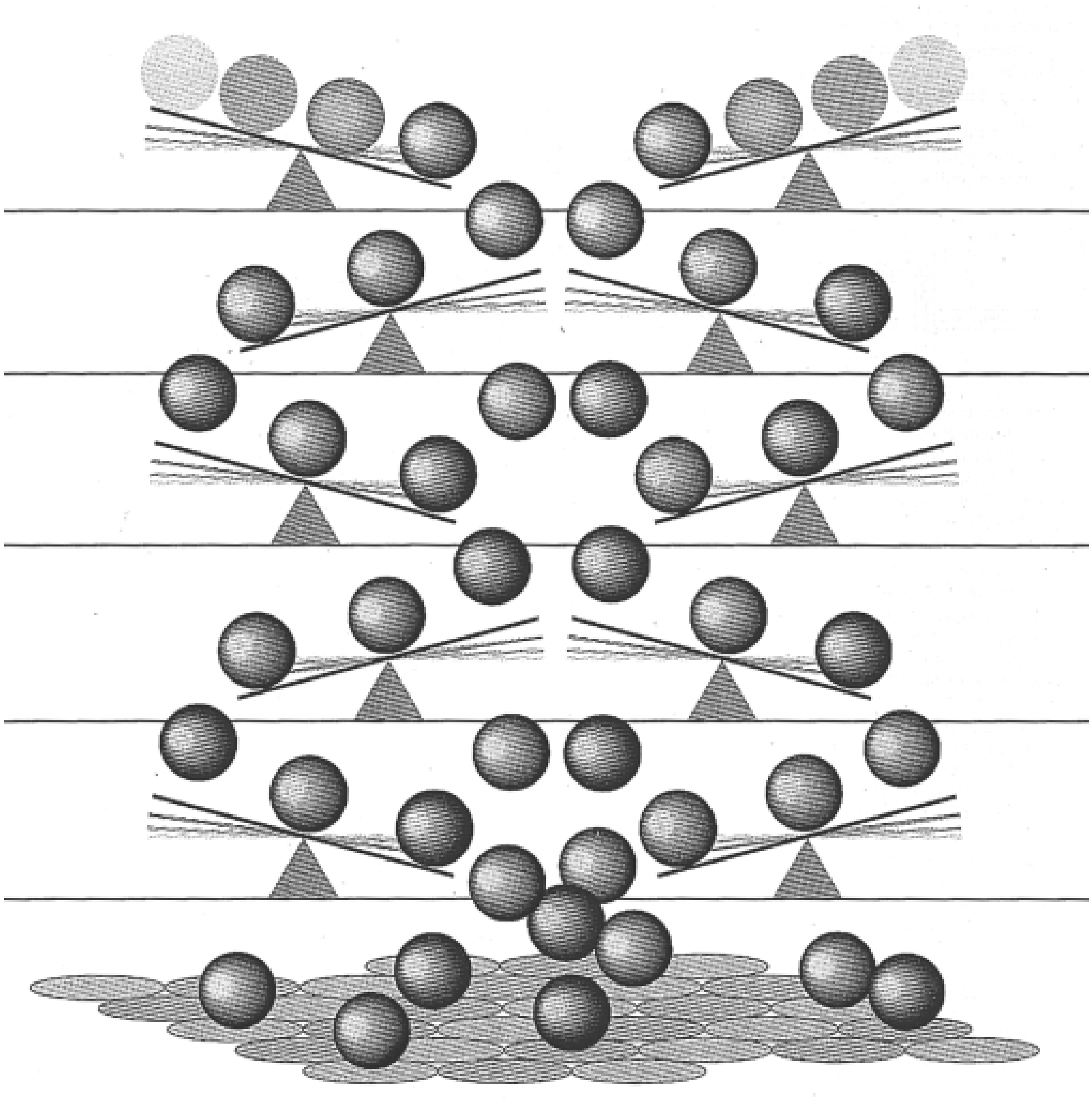
AI Magazine Volume 9 Number 4 (1988) (© AAAI)

A Method for Evaluating Candidate

James R. Slagle and Michael R. Wick



Expert System Applications

We built on previous work to develop an evaluation method that can be used to select expert system applications which are most likely to be successfully implemented. Both essential and desirable features of an expert system application are discussed. Essential features are used to ensure that the application does not require technology beyond the state of the art. Desirable features help point to the applications that stand the greatest chance for successful implementation. Advice on helpful directions for evaluating candidate expert system applications is also given.

Expert systems are entering a critical stage as interest spreads from university research to practical applications (Harmon and King 1986; Hayes-Roth, Waterman, and Lenat 1983; Waterman 1986). If expert systems are to withstand this transition, special attention must be paid to the applications that are selected to test the real-world impact of this technology. It is essential that expert systems avoid the problems which plagued AI in general during its critical stage. Expectations of expert system technology must be kept realistic. To this end, the selection of potential applications must be guided by strict consideration not only of the abilities of expert system technology but also of the current limitations.

Over the last four years, we developed an expert system shell (AGNESS) at the University of Minnesota (Slagle, Wick, and Poliac 1986). Numerous real-world applications were evaluated as test cases for the AGNESS system with varying degrees of success (Long et al. 1987; Slagle 1987; Slagle et al. 1988; Slagle et al. 1987; Slagle et al. 1986). We combined the knowledge gained from our experience into an efficient and convenient method of evaluating potential expert system applications. Our method is an extension and reformulation of similar work by Prerau (1985). We built on his work in three ways: (1) we extended the list of evaluation criteria to reflect experience gained during our research; (2) we reorganized the entire list of evaluation criteria around two axes, namely, the essential-desirable features axis and the users and their management-the task-the expert axis; and (3) we defined an analytic method for using the identified features.

Gathering Candidate Applications

The initial choice of the candidate applications to evaluate is a critical step in the application selection process. The key to a full and rich set of candidate applications is *divergent thinking*, which means finding as many candidates as possible. Analyze all aspects of your daily routine for applications. Such areas could include technical functions, manufacturing functions, scientific analysis and experimentation, iterative or redundant tasks, administrative and accounting tasks, and management tasks. Remember the obvious: An application cannot be selected if it is not a candidate.

Evaluation of Candidate Applications

The evaluation of a candidate application is based on the notion of scoring the features of the application and combining these scores for an overall candidate value. The candidate with the highest value should then be selected as the expert system application. By using this method, each candidate is treated consistently, and documented reasoning exists to explain the choice of the expert system application. Several features of expert system evaluation depend on the quality of the knowledge engineers and the expert system building tools used. Our discussion does not explicitly attack these issues; however, the evaluation method does allow for such concerns (using the feature weights).

The key to successful application evaluation is *convergent thinking:* Be

Term	Definition	Range
Feature Weight	The weight of this feature in the overall decision.	0..10
Feature Value	The degree to which each feature is present.	0..10
Feature Score	The Feature Weight times the Feature Value.	0..100
Candidate Value	The overall value of a candidate.	0..10

Table 1

as critical as possible in trying to narrow the field of candidate solutions. Play the role of devil's advocate. Remember, a little extra work during this process can save extra work and money during expert system construction.

The Basic Evaluation Method

Each feature of expert system development (discussed in the following subsections) is assigned a weight from 0 to 10 depending on its relative importance (for an example, see the section entitled Two Sample Evaluations). Each feature weight should be treated as a constant, remaining the same for all candidate applications. Treating each feature weight as a constant avoids the tendency to tailor the evaluation toward the application instead of tailoring the application toward the evaluation. The features are partitioned into essential versus desirable. The distinction between these partitions is explained later.

Once the feature weights are established, the method proceeds as follows for each candidate application:

1. Assign a feature value from 0 (absent) to 10 (fully present) for each essential feature.
2. If a feature value is low (from 0 to 3), and the corresponding feature weight is relatively high (above 5), try to vary the application to increase the feature value. If the application can not be varied to increase the value (past 3), discard the application.
3. Multiply each essential feature value by its corresponding essential feature weight to obtain each essential feature score.
4. Sum the essential feature scores and essential feature weights over all essential features.
5. Assign a feature value from 0 (absent) to 10 (fully present) for each desirable feature.
6. Multiply each desirable feature value by its corresponding desirable feature weight to obtain each desirable feature score.
7. Sum the desirable feature scores and desirable feature weights over all desirable features.
8. Divide the sum of all scores by the sum of all weights to obtain the overall candidate value (cvalue).
9. Try slightly varying each candidate application to improve its cvalue.
10. Select a candidate with the highest cvalue.

Table 1 further defines the critical elements of the evaluation process. The evaluation sheets in this article provide a guide for performing the candidate evaluation. Overall, the process evaluates the following formula for each candidate and chooses a candidate with the highest cvalue:

$$\text{cvalue} = \frac{\sum_{i=1}^{40} \text{weight}_i * \text{value}_i}{\sum_{i=1}^{40} \text{weight}_i}$$

Essential Features

An *essential feature* is a feature that current expert system technology requires in order for the application to be a success. Each essential feature is described in one of the following subsections. All features, both essential and desirable, are classified into one of three groups: (1) the users and their management, (2) the task, and (3) the expert. Different expert system project goals can result in a different partitioning of the features than is presented here.

The Users and Their Management. *The recipients of the expert system agree that the payoff is high.* The application performs a needed and useful task. Most importantly, the expert system is planned for actual use. The developers, management, and users should agree that the use of an expert system to perform the application is feasible and important.

The recipients have realistic expectations of the system's scope and limitations. The recipients of the expert system are not too optimistic or too pessimistic. AI has been plagued from the beginning with false claims of mystical systems. The users and their management should be aware that the system might not always give correct answers and at best can act as a narrow and fragile version of the human expert. In general, users and management should be educated on the strengths and weaknesses of expert system technology (Harmon and King 1986).

The project has management commitment. Management has committed not only time and money but also interest in seeing the project through to the end.

The Task. *The application task is not natural language intensive.* Given current expert system technology, both natural language generation and understanding are highly difficult tasks. If the application task requires extensive natural language capabilities, it is not an appropriate application.

The application task is knowledge intensive. Knowledge plays a key role in the problem-solving process. The amount of knowledge required should be large enough to result in a useful system but small enough to be captured. This knowledge can include domain-specific knowledge as well as knowledge of how to intelligently manipulate volumes of data.

The application task is heuristic in nature. The task requires the use of rules, heuristics, and strategies. Usually, this type of knowledge is necessary to overcome the enormous size of the search space. If the task can be completely performed using an algorithmic approach, then it is not an appropriate expert system application. This is not to say that portions of the problem cannot have algorithmic solutions. In fact, intelligently coordinating data from various algorithms often makes an excellent application for an expert system. However, a major portion of the task should be heuristic.

Test cases ranging from easy to difficult are available for the application. Testing is a vital stage in the development of any system, and an expert system is no exception. For each test case, an expert solution must also be available so that the expert system can be compared to a meaningful norm.

The system is able to undergo incremental growth. An application that requires the entire problem be attacked at once is unpromising. The application should be divisible into chunks that can be attacked relatively independently of one another. This approach allows work and progress to be judged in stages, allows prototype implementation and a gradual introduction into the workplace.

The application task requires little or no common sense. Although researchers are continuing to study the representation of commonsense knowledge, no practical systems have been developed to date (Lenat, Prakash, and Shepherd 1986). A problem requiring common sense on the part of the expert should be left to a research project.

The task does not require optimal results. Because heuristic methods are most likely to be used, the task should not require 100 percent correct or optimal results. A reasonable number of errors should be acceptable. An error estimate is often hard to obtain for an expert system. Such a situation should be considered before attempting to build the expert system.

The task will be performed in the future. Expert system development is generally expensive and time consuming. Be sure that the task being undertaken will still be performed well into the future. The longer the potential need for the task to be done, the higher the potential payoff for constructing an expert system to do the task.

The task is not essential to an approaching deadline. Expert system development is a difficult process for which to give a precise completion date. Another essential project should not depend on the successful development of the expert system within some constrained time limit.

The task is easy but not too easy. It is vital to the success of the project that the application task require the power of an expert system, no more and no less.

The following four characteristics, among others, show that a task is easy enough to be implemented. First, the task can be performed by an expert in a reasonable amount of time. A rule often forgotten during the evaluation of a candidate task is that an expert must exist who can perform the application task. Avoid a problem for which no human expert exists. Also, the time that it takes the expert to solve the problem should be consistent with current expert system abilities. As a guide, the task should take between 15 minutes and 8 hours for the expert to complete.

Second, the problem domain of the task is stable. This means that the domain should be well established and unlikely to undergo vast changes during the life of the expert system project. This stability does not require that the problem-solving process required to perform the task be well understood, simply that the basics of the task domain be established.

Third, the task is self-contained. Solving the problem should not require enormous volumes of knowledge. For example, tasks that require extensive knowledge of world politics should be avoided.

Fourth, the task is definable. If the problem cannot be defined clearly and precisely, it is probably not well understood. In addition, an explicit task description is essential in describing the scope of the project.

Characteristics that indicate the problem is not too easy and requires the power of an expert system include the following three elements. First, performing the task requires expertise. If a trained technician can perform the task with the same efficiency and skill as an experienced expert, the task does not require expertise and will unlikely make a good expert system application. Expert systems are designed to be knowledge driven; thus, performance of the task should increase with experience (knowledge).

Second, the task involves many factors. The number of factors that must be considered is a good indication of the difficulty of the task. If the problem requires only a half dozen or so factors, it is probably not a good application for an expert system.

Third, traditional methods are inadequate for doing the task. If an efficient algorithm or efficient approximation algorithm exists, the domain is inappropriate for an expert system.

The Expert. *An expert exists.* The strength of expert system technology is in the ability to capture human expert knowledge in a computer program. In order to capture such knowledge, it must exist. Be sure you have access to an expert before even considering expert system development.

The expert is a genuine expert. Because an expert system aims at capturing the knowledge of a human expert, it is essential that the knowledge used be as good as possible. Two characteristics of the domain expert can help determine the degree of expertise. First, the expert is highly respected by experienced people in the domain field. Because the goal of the project is often to simulate the expert's performance, this expert should be viewed by others as a genuine authority whose solutions are of high quality. Second, the expert has considerable experience in the problem domain. This characteristic gets at the point that the expert's knowledge should have been derived from experience (thus the title expert) and not from standard book knowledge.

The expert is committed to the project for its duration. The development of an expert system usually involves months of extensive interaction with the domain expert. It is detrimental to the project if the expert leaves before the completion of the expert system. The expert must commit both time and interest if the project is going to be a success.

The expert is cooperative. Much of the project time is consumed by interactions with the domain expert. As with any endeavor, the investigating parties must have a good working relationship. The expert must have a personality that is easy to work with.

The expert is articulate. During each knowledge engineering session, the expert is generally asked to work through sample problems speaking aloud personal thoughts and

Score	=	Weight	x	Value		Feature
100	=	10	x	10	:	Recipients agree on high payoff.
100	=	10	x	10	:	Recipients have realistic expectations.
100	=	10	x	10	:	Project has management commitment.
100	=	10	x	10	:	Task is not natural language intensive.
56	=	7	x	8	:	Task is knowledge intensive.
80	=	8	x	10	:	Task is heuristic in nature.
100	=	10	x	10	:	Test cases are available.
70	=	7	x	10	:	Incremental growth is possible.
100	=	10	x	10	:	Task requires no common sense.
80	=	8	x	10	:	Task does not require optimal solution.
100	=	10	x	10	:	Task will be performed in future.
70	=	7	x	10	:	Task not essential to deadline.
80	=	8	x	10	:	Task easy, but not too easy.
100	=	10	x	10	:	An expert exists.
100	=	10	x	10	:	Expert is a genuine expert.
100	=	10	x	10	:	Expert is committed to entire project.
80	=	8	x	10	:	Expert is cooperative.
64	=	8	x	8	:	Expert is articulate.
80	=	8	x	10	:	Expert has successful history.
64	=	8	x	8	:	Expert uses symbolic reasoning.
70	=	7	x	10	:	Hard to transfer expertise.
70	=	10	x	7	:	Expert does not use physical skills.
100	=	10	x	10	:	Experts agree on good solutions.
100	=	10	x	10	:	Expert does not need creativity.
2064		214				

Table 2. Essential Features Evaluation Form for the Dam Domain.

justifications. It is from this information that the knowledge base is built. The more articulate the expert, the more valuable each session is. However, the use of an expert who is also a teacher can be dangerous. A teacher has a tendency to tutor the knowledge engineer and not reveal the true mechanism that is used during problem solving. Although the problem of an expert "saying one thing and doing another" is common in all knowledge engineering projects, the use of an expert who is a teacher can significantly complicate the process. This feature (as well as that preceding it) depends on the quality and personality of the knowledge engineering team. For example, a skilled knowledge engineer might be able to overcome a teacher's tutoring. The feature weights can be used to adjust for the level of experience and the personalities of the knowledge engineering team.

The expert has a history of successful task performance. An expert who sometimes needlessly abandons a project before completion should be avoided. An expert who tends to become sidetracked during a project should also be avoided. The input of the domain expert can be required during all phases of expert system development.

The expert uses symbolic reasoning when performing the application task. Again, this point relates to the heuristic nature of an expert system. The explicit use of symbolic knowledge has several advantages, including ease of explanation. The majority of the knowledge captured by the expert system should be symbolic.

It is hard but not too hard to transfer the expert's expertise to another human. This point is related to the use of heuristics. If the knowledge used to solve the problem is so trivial that it can easily be passed to another person, then the application does not require the power of an expert system. However, because knowledge engineering is an important part of expert system development, the task of knowledge transfer should be possible. A good measure is the difficulty with which the expert can record the knowledge used in doing the task. If the expert can easily write a reasonable set of knowledge that can solve the problem, the expertise is probably not appropriate for an expert system.

The expert uses cognitive, not physical, skills in performing the task. An application task that requires the use of physical sensory organs should be avoided. Concentrate on a problem that is knowledge intensive, not sensory intensive.

Experts agree on what constitutes a good solution for the application problem. A problem domain with a wide variance in the answers supplied by experts seldom results in a useful expert system application. The system must be recognized as able to give good answers if it is to be used.

The expert does not need to be creative in solving the problem. Creativity is a poorly understood concept and, as such, is nearly impossible to capture in a computer program. A task can be said to require creativity on the part of the expert if steps exist in the problem-solving process for which no a priori justification can be found; in other words, the only justification for a problem-solving step is the quality of the final solution.

Desirable Features

Desirable features are those which are not necessarily required by current expert system technology; however, experience shows that without these features the difficulty of the expert system project can significantly increase. Each desirable feature is discussed in the following subsections.

The Users and Their Management. *Management will commit time, money, and interest in follow-on efforts if the system is successful.* A serious commitment of resources for a prolonged investigation of expert system development is invaluable. Several important lessons will be learned with each expert system project, and it is important to have an environment that allows these lessons to be used in future work.

The insertion into the workplace is smooth. The expert system does not interfere in the user's routine any more than necessary. An important consideration is whether the user wants an expert system. The user should be available to test early versions of the system to allow for feedback on what features of the system are useful and what features are more of a burden than can be tolerated. As has been learned in traditional software development, no matter how good the system, if the user isn't happy, it doesn't get used.

The system interacts with the user. The ability to involve the user in the problem-solving process is often critical to the acceptance of an expert system. A user is reluctant to use a system that simply takes the input and generates the output. In general, an expert system is good at handling interaction with the user, and this ability should be exploited whenever possible. As is described in the next paragraph, an expert system can use explanation to achieve a type of transparency that allows the user to follow and understand the problem-solving process. This ability can significantly contribute to the interactive nature of the system.

The system is able to explain its reasoning to the user. The user's confidence in, and patience with, an expert system can be significantly increased by allowing the user to query the system during the problem-solving process (Swartout 1983; Wick and Slagle 1988). A system that can not explain its actions is of little practical use.

Score	=	Weight	x	Value		Feature
56	=	8	x	7	:	Management committed to follow-on.
36	=	4	x	9	:	Insertion into work place smooth.
40	=	4	x	10	:	System interacts with user.
40	=	4	x	10	:	System can explain reasoning.
40	=	4	x	10	:	System can intelligently question user.
40	=	4	x	10	:	Task identified as problem area.
40	=	4	x	10	:	Solutions are explainable.
40	=	5	x	8	:	Task does not require real-time response.
80	=	8	x	10	:	Similar expert systems built before.
50	=	5	x	10	:	Task performed in many locations.
0	=	3	x	0	:	Task performed in hostile environment.
32	=	4	x	8	:	Task involves subjective factors.
0	=	3	x	0	:	Expert unavailable in future.
32	=	4	x	8	:	Expert intellectually attached to project.
40	=	5	x	8	:	Expert does not feel threatened.
20	=	2	x	10	:	Expertise loosely organized.
586		71				

Table 3. Desirable Features Evaluation Form for the Dam Domain.

The system is able to intelligently question the user. A long, exhaustive questioning scheme tends to bore the user, resulting in decreased use. The system should be equipped with a scheme that allows the most relevant question to be asked first. Thus, the system is able to move to a solution faster, possibly avoiding unnecessary questions (Slagle and Hamburger 1985).

The key point of these features is that the final system must be friendly to use. Expert system development is expensive and time consuming, and as such, every precaution must be taken to ensure that the final system is used in its intended environment.

The Task. *The task was previously identified as a problem area.* If the task was considered before as a potential area for automation, chances increase for many of the essential features such as management commitment to be realized.

Solutions are explainable and interactive. Again, acceptance by the users is significantly increased if the solution requires interaction during problem solving. It is not enough that the system has the facilities to explain solutions; the task must also allow for easy to understand explanations.

The task does not require real-time response. Although significant progress is being made in the development of real-time expert systems, the extra burden can be detrimental to an expert system project.

The task is similar to tasks performed successfully by previous expert systems. Expert systems are relatively new, and it is often hard to judge the difficulty of a given project. Previous expert system projects should be used to help gauge the

Score	Weight	=	CValue
2064	214		
586	71		
2650	/ 285	=	9.30

Table 4. Overall Evaluation Form for the Dam Domain.

Score	=	Weight	x	Value		Feature
40	=	10	x	4	:	Recipients agree on high payoff.
50	=	10	x	5	:	Recipients have realistic expectations.
50	=	10	x	5	:	Project has management commitment.
70	=	10	x	7	:	Task is not natural language intensive.
49	=	7	x	7	:	Task is knowledge intensive.
80	=	8	x	10	:	Task is heuristic in nature.
100	=	10	x	10	:	Test cases are available.
42	=	7	x	6	:	Incremental growth is possible.
50	=	10	x	5	:	Task requires no common sense.
64	=	8	x	8	:	Task does not require optimal solution.
100	=	10	x	10	:	Task will be performed in future.
70	=	7	x	10	:	Task not essential to deadline.
40	=	8	x	5	:	Task easy, but not too easy.
80	=	8	x	10	:	An expert exists.
80	=	8	x	10	:	Expert is a genuine expert.
100	=	10	x	10	:	Expert is committed to entire project.
32	=	8	x	4	:	Expert is cooperative.
56	=	8	x	7	:	Expert is articulate.
80	=	8	x	10	:	Expert has successful history.
48	=	8	x	6	:	Expert uses symbolic reasoning.
70	=	7	x	10	:	Hard to transfer expertise.
50	=	10	x	5	:	Expert does not use physical skills.
80	=	10	x	8	:	Experts agree on good solutions.
100	=	10	x	10	:	Expert does not need creativity.
1581		214				

Table 5. Essential Features Evaluation Form for the Introduction Domain.

scope of the problem.

The task is performed in many locations. One advantage of expert system technology is that it allows the knowledge of a rare expert to be captured in a computer program. If the task is performed in many locations, this program can then be used to duplicate the problem-solving ability of this one rare person at every location. The more locations, the higher the potential payoff for capturing the expertise.

The task is performed in a hostile environment. It is to everyone's advantage to replace a human with a machine in a hostile environment. Even in a domain where the system can not be used to replace the human, the use of a computer can ensure that human factors such as panic and fear do not affect the problem-solving process.

The task involves subjective factors. Expert system technology is especially tailored to deal with subjective judgment. As such, a task that involves subjective factors often makes a better application than a task which does not.

The Expert. *The expert is unavailable in the future.* An application domain in which the expert is retiring from the company makes an excellent expert system application. The unavailability of the expert not only forces use of the expert system but also gives an opportunity to demonstrate the high payoff of capturing the knowledge of a human expert.

The expert is able to become intellectually attached to the project. Few things motivate cooperation more than intellectual interest. An expert who is interested in understanding the expert system development process is likely to be willing to spend extra time on the project to make it work. Sessions explicitly designed to capture the expert's intellectual attention should be an important part of expert system development.

The expert does not feel threatened by the expert system. The expert should be aware of the advantages and limitations of an expert system. An expert who feels threatened is likely to resist full involvement in the project. Special attention should be paid to educating the expert so that it is understood the system is not meant as a replacement but rather as an intelligent assistant. In the case where the expert is retiring from the company, the expert system can act as an assistant to the person replacing the expert.

The expertise used by the expert in solving the problem is loosely organized. This organization allows partitioning of the problem-solving process into smaller pieces that are easier to handle and debug; for example, the use of specific strategies to solve particular problem instances often leads to efficient expert systems.

Two Sample Evaluations

This section presents two sample expert system evaluations to illustrate our evaluation method. Although space does not allow a complete discussion of the evaluation features, key items are discussed for each domain application. The complete forms are provided in tables 2 to 7.

The first step in the evaluation process is to determine the feature weights. Each weight must be set to reflect the relative importance of the feature within the goals of the particular expert system team. The weights we chose are for an expert system group that has an average of seven years experience per member.

The first domain to be evaluated involves building an expert system to aid in the analysis of potential concrete dam failures. Our group was asked by an independent power company to investigate whether an expert system could be built to allow an on-field inspection to be done more efficiently and by a less experienced inspector. The overall structure of the system would be as an intelligent assistant to the inspector, guiding the individual through the significant questions.

Table 2 shows the Essential Features Evaluation Form for this domain. Two values require explanation. First is the feature that the expert uses symbolic knowledge. In our preliminary study of the domain, we found that the expert uses several numeric processes during the problem-solving activity. However, as the integration of these processes appears to take place through the use of symbolic knowledge, this flaw is not fatal. The second key feature—the expert does not use physical skills—is slightly more serious. Although the inspector uses cognitive skills to evaluate the evidence, a significant amount of physical skill is required to collect this evidence. For example, the inspector must be able to accurately judge the degree of cracking or discoloring in the dam foundation. Errors in measuring these elements can have an effect on the quality of the answer. Even though physical skill is required, the task is still mainly cognitive. For this reason, a value of 7 is assigned to this feature.

Table 3 shows the Desirable Features Evaluation Form for this domain. Although more fluctuation exists in the values for this set of features, most values remain relatively high. However, two features receive zeros. The task is not performed in a hostile environment, nor is the expert unavailable for future needs. The effect of these low scores is minimized by the relatively low weights assigned to these features.

Table 4 shows the overall value for the dam candidate expert system. Notice that the overall value for a candidate application falls in the range 0 to 10 (just like each feature value). This position allows the overall meaning of the value to be easily judged.

At the same time that the power company approached our group, an unrelated company proposed to use our services. As is often the case, a decision had to be made based on the overall quality of the applications. The second domain was that of a matchmaker for an introduction (dating) service.

Tables 5, 6, and 7 show the Essential Features Evaluation Form, the Desirable Features Evaluation Form, and the Overall Evaluation Form, respectively. Because many of the values for this domain are low, we do not

Score	=	Weight	x	Value		Feature
0	=	8	x	0	:	Management committed to follow-on.
8	=	4	x	2	:	Insertion into work place smooth.
40	=	4	x	10	:	System interacts with user.
40	=	4	x	10	:	System can explain reasoning.
40	=	4	x	10	:	System can intelligently question user.
16	=	4	x	4	:	Task identified as problem area.
40	=	4	x	10	:	Solutions are explainable.
50	=	5	x	8	:	Task does not require real-time response.
56	=	8	x	7	:	Similar expert systems built before.
50	=	5	x	10	:	Task performed in many locations.
0	=	3	x	0	:	Task performed in hostile environment.
40	=	4	x	10	:	Task involves subjective factors.
15	=	3	x	5	:	Expert unavailable in future.
12	=	4	x	3	:	Expert intellectually attached to project.
0	=	5	x	0	:	Expert does not feel threatened.
6	=	2	x	3	:	Expertise loosely organized.
413		71				

Table 6. Desirable Features Evaluation Form for the Introduction Domain.

Evaluation Steps:

Essential Evaluation:

1. Essential Score = $\sum_{i=1}^{24} weight_i \bullet value_i$
2. Essential Weight = $\sum_{i=1}^{24} weight_i$

Desirable Evaluation:

1. Desirable Score = $\sum_{j=1}^{16} weight_j \bullet value_j$
2. Desirable Weight = $\sum_{j=1}^{16} weight_j$

Candidate Evaluation:

1. CValue = $\frac{\text{Essential Score + Desirable Score}}{\text{Essential Weight + Desirable Weight}}$

Table 7. Overall Evaluation Form for the Introduction Domain.

Score		Weight	=	CValue
1581		214		
413		71		
1994	/	285	=	7.00

discuss the individual features. The underlying reason for the low ratings can be seen by looking at the attitude of the potential customer. We found the introduction service personnel to be resistant to, and highly suspect of, the expert system approach. In fact, they argued that the customers of their service would not like the notion of a computer matching people. They favored the old-fashioned approach of one-to-one contact with the clients. This fatal flaw caused several low values during the evaluation. For example, features such as recipients agree on high payoff, recipients have realistic expectations, and smooth insertion into workplace were strongly affected.

As these examples illustrate, the method can be used to guide the process of expert system candidate selection in the most promising direction. In the next section we present some helpful hints on other ways of using our method to improve expert system selection.

Helpful Hints

Organize an evaluation form that can be used to ensure consistent evaluation of all applications. A standard spreadsheet is valuable in organizing this form. The evaluation form is helpful for both the evaluation process and the understanding of the features outlined in this article. Retain the form for each candidate so that you have a permanent evaluation for future reference. This information can be valuable for conducting retrospective evaluations in order to revise the feature weights for other projects.

An important advantage of our method is that it can be used to highlight those features which are most likely to be trouble spots during development. When varying an application in an attempt to increase its value, remember that increasing a feature value to 10 (the maximum) can cause a change in the candidate's value equal to the feature weight times 10 minus the feature value divided by the sum of the feature weights (all other feature values constant). Thus, to determine which features to attempt to vary, simply compute the following value for each feature:

$$potential_i = \frac{weight_i * (10 - value_i)}{\sum_{j=1}^{40} weight_j}$$

Call the resulting value the feature potential. Then, select the features with the highest feature potential as those features to attempt to vary. This method focuses attention on the features with the greatest potential for impact on the candidate's value.

As a result of our method, two immediate projects could be used to further develop an understanding of the candidate expert system evaluation process. First, develop an expert system (or a knowledge-based system) to evaluate each feature; that is, write an expert system to fill in the feature values. Although the project might not result in a usable system, the process should educate its creators about what characteristics influence each feature. Second, implement an expert system that matches the abilities of certain expert system paradigms (such as forward versus backward chaining) with the characteristics of the problem under consideration. For further discussion, see Basden (1983) and Klein and Dolin (1986).

Conclusion

The use of our method for selecting a candidate expert system application can result in work that focuses on that problem for which the current expert system technology is best equipped. By distinguishing between essential and desirable characteristics, the evaluation process can not only prune out applications that cannot be solved but can also point to applications which have a high likelihood of being easier to complete and providing high payoffs.

Other advantages of our method include the following: (1) it provides a permanent and consistent means by which to evaluate candidate expert system applications; (2) the explicit use of weights allows the method to be tailored to the specific goals of the expert system team; (3) the analytic nature of the method allows for convenient identification of potential trouble spots; and (4) the explicit evaluation process can provide knowledge engineers with a set of probing questions that can quickly differentiate good applications from bad. It is imperative that as expert system development moves into the critical stage of real-world testing, only those practical applications which are promising are attempted.

Acknowledgments

This research is sponsored by the National Science Foundation, grant number DCR8512857, and the Microelectronics and Information Sciences Center of the University of Minnesota.

References

Basden, A. 1983. On the Application of Expert Systems. *International Journal of Man-Machine Studies* 19(5): 461-477.

Harmon, P., and King, D. 1986. *Expert Systems: Artificial Intelligence in Business.* New York: Wiley.

Hayes-Roth, F.; Waterman, D. A.; and Lenat, D. B. 1983. *Building Expert Systems.* Reading, Mass.: Addison-Wesley.

Klein, P. J., and Dolin, S. B. 1986. Problem Features That Influence Design in Expert Systems. In Proceedings of the Fifth National Conference on Artificial Intelligence, 956-962. Menlo Park, Calif.: American Association for Artificial Intelligence.

Lenat, D. B.; Prakash, M.; and Shepherd, M. 1986. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks. *AI Magazine* 6(4): 65-85.

Long, J. M.; Slagle, J. R.; Leon, A. S.; Wick, M. R.; Fitch, L. L.; Matts, J. P.; Karnegis, J. N.; Bissett, J. K.; Sawin, H. S.; and Stevenson, J. P. 1987. An Example of Expert Systems Applied to Clinical Trials: Analysis of Serial Graded Exercise {ECG} Test Data. *Controlled Clinical Trials* 8: 136-145.

Prerau, D. S. 1985. Selection of an Appropriate Domain for an Expert System. *AI Magazine* 6(2): 26-30.

Slagle, J. R. 1987. Expert System Shells and Their Applications. In Proceedings of the Navy Regional Data Automation Center/Oak Ridge National Laboratory Conference on Expert Systems Technology. Forthcoming.

Slagle, J. R., and Hamburger, H. 1985. An Expert System for a Resource Allocation Problem. *Communications of the ACM* 28(9): 994-1003.

Slagle, J. R.; Finkelstein, S. M.; Leung, L. W.; and Warwick, W. J. 1988. MONITOR: An Expert System that Validates and Interprets Time-Dependent Partial Data Based on a Cystic Fibrosis Home Monitoring Program. *IEEE Transactions on Biomedical Engineering.* Forthcoming.

Slagle, J. R.; Long, J. M.; Irani, E.; Hunter, D.; Matts, J. P.; and San Marco, M. 1987. ESCA, An Expert System for Evaluation of Serial Coronary Angiograms: The POSCH A.I. Project, Technical Report, Computer Science Dept., Univ. of Minnesota.

Slagle, J. R.; Long, J. M.; Wick, M. R.; Matts, J. P.; and Leon, A. S. 1986. The ETA Project: A Case Study of Expert Systems for the Analysis of Serial Clinical Trial Data. In Proceedings of the Medinfo'86 Fifth World Conference on Medical Informatics, 155-159. Elsevier Science Publishers, Holland.

Slagle, J. R.; Wick, M. R.; and Poliac, M. O. 1986. AGNESS: A Generalized Network-Based Expert System Shell. In Proceedings of the Fifth National Conference on Artificial Intelligence, 996-1002. Menlo Park, Calif.: American Association for Artificial Intelligence.

Swartout, W. R. 1983. XPLAIN: A System for Creating and Explaining Expert Consulting Programs. *Artificial Intelligence* 21:285-325.

Waterman, D. A. 1986. *A Guide to Expert Systems.* Reading, Mass.: Addison-Wesley.

Wick, M. R., and Slagle, J. R. 1988. An Explanation Facility for Today's Expert Systems. *IEEE Expert.* Forthcoming.