

Selection of an Appropriate Domain for an Expert System

David S. Prerau

Computer Science Laboratory, GTE Laboratories, Incorporated, 40 Sylvan Road, Waltham, Massachusetts 02254

Abstract

This article discusses the selection of the domain for a knowledge-based expert system for a corporate application. The selection of the domain is a critical task in an expert system development. At the start of a project looking into the development of an expert system, the knowledge engineering project team must investigate one or several possible expert system domains. They must decide whether the selected application(s) are best suited to solution by present expert system technology, or if there might be a better way (or, possibly, no way) to attack the problems. If there are several possibilities, the team must also rank the potential applications and select the best available. To evaluate the potential of possible application domains, it has proved very useful to have a set of desired attributes for a good expert system domain. This article presents such a set of attributes. The attribute set was developed as part of a major expert system development project at GTE Laboratories. It was used recurrently (and modified and expanded continually) throughout an extensive application domain evaluation and selection process.

This article discusses the selection of the domain for a knowledge-based expert system. In particular, it focuses on selecting an expert system domain for a corporate application. The choosing of the domain is a critical task in

I would like to thank the management and members of the Knowledge Based Systems Department at GTE Laboratories for their continuing support in this work. Especially, I would like to thank Ralph Worrest of the Department for his work with me on the application evaluation and selection phase of our expert system development project and for his suggestions throughout the course of the work that led to this paper. Also, I'd like to thank Shri Goyal, Alan Gunderson, Alan Lemmon, Robert Reinke, and Roland Van der Meer of the Department, and Charles Rich of MIT for their suggestions on the contents of this paper.

the development of an expert system, and thus a significant amount of effort should go into the selection process.

Background

Interest in artificial intelligence by the corporate business community has been growing dramatically in the last few years, and many corporations have set up AI groups or are in the process of doing so. One of the prime areas of corporate interest is expert systems. Though the number of expert systems actually functioning in a corporate environment is still relatively small, the number of projects looking into expert system development is growing rapidly.

The knowledge engineering project team working on an expert system development must investigate possible application domains. In some cases there is a very specific application, chosen by management, for which an expert system is to be developed. In this situation, it is likely that those who selected the application area had little technical knowledge of artificial intelligence or expert systems. Thus, the project team must decide whether the selected application is one that is best suited to solution by present expert system technology, or if there might be a better way (or, possibly, no way) to attack the problem.

In other cases, the project team is asked to select one of several corporate problems or to survey corporate concerns to find a good application of expert system technology. Here, the project team must not only decide if an application is suited to present expert system technology, but must also rank potential domains and select the best available application.

To evaluate the potential of a possible application, it has proven very useful to have a set of the attributes desired in a good expert system domain. This article provides such a set of attributes. The set includes technical

attributes as well as attributes related to non-technical corporate issues

An Application Domain Evaluation Process

The set of desired expert system domain attributes was developed as part of a major expert system development project at GTE Laboratories. It was used recurrently (and modified and expanded continually) throughout an extensive application domain evaluation and selection process. Over 50 corporate managers and "experts" were interviewed, and over 30 extremely diverse possible expert system applications areas were considered, at least briefly. This list was narrowed to eight major possibilities, and these were further analyzed and ranked. Two primary candidate areas were studied in great detail. Finally, one application area was chosen, and our system development was begun.

At each stage of the selection process, the set of attributes proved very useful. In initial interviews, a discussion of the attributes was an excellent way to give our interviewees, who usually knew nothing about artificial intelligence or expert systems, some quick idea of the sort of application area for which we were looking. As each potential application surfaced, a brief check through the desired attribute list enabled us to identify possible problems related to the candidate area, and then to focus our further questions. When the set of major possibilities was determined, we were easily able to highlight the good and bad points of each potential application. Finally, when the actual application area was decided upon, we used the attribute list to justify the decision. One further point: at each step, the list proved very useful to justify the dropping of politically favored candidate areas.

Desired Properties of the Domain

This section presents a set of desired attributes for the domain of an expert system for a corporate application. Though many of these attributes are applicable to all expert systems, there are some that are specific to the development of an expert system in a corporate environment. These involve, for example, the likelihood of corporate acceptance of a system, the support for the system development by corporate management, etc. There are probably analogous points that apply to an academic or other environment, but these are not addressed here.

The attribute set was developed from the perspective of providing a real working expert system to solve a corporate problem, using state of the art expert system techniques. The discovery of new or better methods for expert system development was not an objective. In fact, a domain that requires a major breakthrough in expert system methodology is probably not a good domain to choose if the goal is to maximize the likelihood of success. Yet, any project that is the first to attack a particular domain is

likely to find some unique properties of the domain that may require new approaches.

There may be a degree of commonality among some of the attributes listed in this section. However, to encourage consideration of the different aspects of domain selection, these commonalities were not eliminated.

Very few of these desired attributes are absolute, and it is unlikely that any domain will meet all of them completely. Furthermore, in each different situation the weighting of the factors will be different, and additional factors may apply. This set does provide, however, a fairly extensive list of aspects to consider in domain selection.

Basic Requirements

- *The domain is characterized by the use of expert knowledge, judgment, and experience.* The goal of the project is to extract a portion of an expert's knowledge, judgment and experience, and put it in a program
- *Conventional programming (algorithmic) approaches to the task are not satisfactory.* If a conventional approach will work well, there is usually less technical risk to using it rather than an expert system approach. Note, however, that expert system methodology may offer some additional advantages over conventional techniques, such as the expected ease of updating and maintaining a knowledge base and the ability to explain results.
- *There are recognized experts that solve the problem today.* If an area is too new or too quickly changing, there may be no real experts. However, these are often the areas that are suggested for expert system developments.
- *The experts are probably better than amateurs in performing the task.* Thus, the task does require expertise.
- *Expertise is not or will not be available on a reliable and continuing basis, i.e., there is a need to "capture" the expertise.* Thus, there is a need for the expert system. For example: (1) expertise is scarce, (2) expertise is expensive, (3) there is a strong dependence on overworked experts, and/or (4) expertise is available today, but will be unavailable, or less available, in the future.
- *The completed system is expected to have a significant payoff for the corporation.*
- *Among possible application domains, the domain selected is that one that best meets overall project goals regarding project payoff versus risk of failure.* For example, a conservative approach would be to attempt to develop a system that would meet some criterion for minimum payoff if successful, and that seems to offer the best chance of success.

Type of Problem

- *The task primarily requires symbolic reasoning.* For a task that primarily involves numerical computation, consideration should also be given to other programming approaches
- *The task requires the use of heuristics, e.g., rules of thumb, strategies, etc.* It may require consideration of an

extremely large number of possibilities or it may require decisions to be based upon incomplete or uncertain information. A strength of expert systems is their ability to handle heuristics. Problems with very large numbers of possibilities or with incomplete or uncertain information are difficult to attack by conventional approaches, but may be amenable to expert system methodologies

- *The task does not require knowledge from a very large number of areas* If it did, the amount of knowledge needed for the expert system would probably be beyond acceptable limits. Also, there are difficulties in combining very heterogeneous knowledge.

- *The system development has as its goal either to develop a system for actual use or to make major advances in the state of the art of expert system technology, but does not attempt to achieve both of these goals simultaneously.* Doing both simultaneously is laudable, but more difficult

- *The task is defined very clearly.* At the project outset, there should be a precise definition of the inputs and outputs of the system to be developed. This is a good attribute of any task. However, it is not necessary that the task definition be fixed for all time. As the system evolves and as situations change, it should be possible to change the task definition accordingly.

The Expert

- *There exists an expert to work with the project.* This is the source of expertise

- *The expert's knowledge and reputation must be such that if the expert system is able to capture a portion of the expert's expertise, the system's output will have credibility and authority.* Otherwise, the system may not be used. (This may not be necessary in a domain where an accepted test for "goodness" of result exists.)

- *The expert has built up expertise over a long period of task performance.* Thus, the expert has had the amount of experience necessary to be able to develop the insights into the area that result in heuristics.

- *The expert will commit a substantial amount of time to the development of the system.* This is often a problem. The best experts, in the most important corporate areas, are usually the ones that can be least spared from their usual position

- *The expert is capable of communicating his knowledge, judgment, and experience, and the methods used to apply them to the particular task.* It is important to find an expert that has not only the expertise, but also the ability to impart it to the project team, whose members probably know little or nothing about the subject area. The expert should be able to introspect to analyze his reasoning process, and then should be able to describe the reasoning process clearly to the project team, and to discuss it with them.

- *The expert is cooperative.* The expert should be eager to work on the project or, at worst, nonantagonistic.

- *The expert should be easy to work with.* The project team and the expert will be spending a lot of time together

- *The expertise for the system, at least that pertaining to one particular sub-domain, is to be obtained primarily from one expert.* This avoids the problem of dealing with multiple experts whose conclusions or problem-solving techniques do not agree. However, there may be some advantages to using multiple experts—e.g., strength of authority and breadth of expertise in sub-domains.

- *If multiple experts contribute in a particular sub-domain, one of them should be the primary expert with final authority.* This allows all the expertise to be filtered through a single person's reasoning process. (Note that some techniques have been developed, in disciplines such as economic modeling and technological forecasting, to allow combining inputs from multiple experts.)

Problem Bounds

- *The task is neither too easy (taking a human expert less than a few minutes) nor too difficult (requiring more than a few hours for an expert).* If the task is too easy, the development of the system may not warrant the effort; if too difficult, the amount of knowledge needed may be beyond the state of the art in knowledge base size.

- *The amount of knowledge required by the task is large enough to make the knowledge base developed interesting.* If it is too small, the task may be more amenable to another approach—e.g., a decision tree.

- *The task is sufficiently narrow and self-contained: the aim is not for a system that is expert in an entire domain, but for a system that is an expert in a limited task within the domain.* This more tightly bounds the task, which should help keep the size of the knowledge base bounded.

- *The number of important concepts (e.g., rules) required is bounded to several hundreds.* This is a reasonable size for an expert system, though the number can go into the thousands.

Domain Area Personnel

- *Personnel in the domain area are realistic, understanding the potential of an expert system for their domain, but also realizing that thus far few expert systems have resulted in actual production programs with major industrial payoff.* The system recipients should not be overly optimistic nor overly pessimistic. The project team may have to educate them to understand what are reasonable expectations.

- *Domain area personnel understand that even a successful system will likely be limited in scope and, like a human expert, may not produce optimal or correct results 100% of the time.* The expert system will probably be no better than a limited version of the expert—this must be enough.

- *There is strong managerial support from the domain area, especially regarding the large commitment of time by the expert(s), and their possible travel or temporary relocation, if required.* This should all be agreed upon up front.

- *The specific task within the domain is jointly agreed upon by the system developers and the domain area personnel.* This helps ensure that the system, if successful, will be useful and will be used.
- *Managers in the domain area have previously identified the need to solve the problem which the system attacks.* This is strong evidence that the system is needed and makes managerial support more likely.
- *The project is strongly supported by a senior manager, for protection and follow-up.*
- *Potential users would welcome the completed system.* If not, will the system ever be used? The project team should consider how to make the system unthreatening to the users and welcomed by them.
- *The system can be introduced with minimal disturbance of the current practice.* This will make the users' acceptance of the system more likely
- *The user group is cooperative and patient.*
- *The introduction of the system will not be politically sensitive or controversial.* If not, the potential resulting problems should be considered in advance. One typical problem: The control or use of the system goes across existing organizational boundaries.
- *The knowledge contained by the system will not be politically sensitive or controversial.* For example, there may be certain practices, embodied in heuristics, which may prove embarrassing if written down, such as how certain customers are treated relative to other customers.
- *The system's results will not be politically sensitive or controversial.* If there will be corporate parties who will challenge the system if its results do not favor them politically (e.g., on appropriation of funds), then it will be much harder to gain system acceptance

Other Desirable Features

- *The system can be phased into use gracefully. Some percentage of incomplete coverage can be tolerated (at least initially), and the determination of whether a sub-problem is covered by the present system is not difficult.* If the system does not have to do everything in order to do something, it can be put in place much sooner. The more difficult problems can be solved later, if at all.
- *The task is decomposable, allowing relatively rapid prototyping for a closed small subset of the complete task, and then slow expansion to the complete task.* This makes development much easier.
- *The task is not all-or-nothing: Some percentage of incorrect or nonoptimal results can be tolerated.* The more toleration for incorrect results, the faster the system can be deployed and the easier it will be to win system acceptance. For example, in a domain where even the best experts are often wrong, system users will not be as upset by an incorrect result from the system
- *The skill required by the task is taught to novices.* Thus, the task is not "unteachable," and there is some experience

with teaching the domain knowledge to neophytes, such as the project team (and, ultimately, the system). Furthermore, this usually means that there is an organization to the knowledge that can prove useful (at least initially) in building the system.

- *There are books or other written materials discussing the domain.* If this is true, then an expert has already extracted and organized some of the domain expertise. As in the previous point, this organized knowledge might prove useful (at least initially) in building the system. Note, however, that one benefit of capturing an expert's domain knowledge might be to make a step toward formalizing a domain that has not been treated in a formal manner before
- *The task's payoff is measurable.* If not, it is harder to demonstrate success to skeptics
- *Experts would agree on whether the system's results are good (correct)* If not, the system's results are open to challenge, even if the system accurately embodies the expert's knowledge.
- *Test cases are available* This makes development much easier.
- *The need for the task is projected to continue for several years.* The need must exist enough beyond the period of system development to generate the payoff.
- *The domain is fairly stable. Expected changes are such that they utilize the strengths of expert systems (e.g., ease of updating or revising specific rules in a knowledge base), but will not require major changes in reasoning processes* An unstable domain may yield a situation where a large number of previously developed knowledge structures (e.g., rules) are no longer valid but cannot easily be changed without redoing the entire development process.
- *The effects of corporate developments that will significantly change the definition of the task can be foreseen and taken into account*
- *No alternative solution to the problem is being pursued or is expected to be pursued.* However, if a project goal is to compare expert system technology to other technologies, this may be just what is desired
- *The project is not on the critical path for any other development, and has no absolute milestones for completion* The use of expert system technology for real corporate applications is still relatively new, and so any development has some risk. Thus, the less dependent other activities are, the better.
- *At the outset of the project, the expert is able to specify many of the important concepts.* This gives good promise of project success.
- *The task is similar to that of a successful existing expert system.* This also makes success more likely.
- *Any requirement for real-time response will not involve extensive effort.* Though it is certainly possible to develop a system for a problem with a real-time requirement, the considerations involved divert effort from the primary task:

knowledge acquisition.

• *The user interface will not require extensive effort.* As with a real-time requirement, if the work required is excessive, it could divert effort from knowledge acquisition.

References

- d'Agapeyeff, A (1983) *Preparing for Fifth Generation Computing*. Fifth Generation World Conference, London, U K.
- Davis R (1982) Expert Systems: Where Are We? And Where Do We Go From Here? *AI Magazine* Vol. 3, No. 2: 3-22
- Gevarter W (1983) Expert Systems: Limited but Powerful *IEEE Spectrum* Vol. 20, No. 8:39-45
- Hayes-Roth F, D. Waterman, & D. Lenat. (1983) *Building Expert Systems* Reading MA: Addison-Wesley
- McDermott J (1981) R1: The Formative Years *AI Magazine* Vol. 2, No. 2:21-29.
- McDermott J (1984) R1 Revisited: Four Years in the Trenches *AI Magazine* Vol. 5, No. 3:21-32
- Roberts S (1983) Computers Simulate Human Experts. *Micro-Min Systems* 16/9
- Smith R. (1984) On the Development of Commercial Expert Systems. *AI Magazine* Vol. 5, No. 3:61-73

PROLOG-86™

Become Familiar in One Evening

Thorough tutorials are designed to help learn the PROLOG language quickly. The interactive PROLOG-86 Interpreter gives immediate feedback. In a few hours you will begin to feel comfortable with it. In a few days you are likely to know enough to modify some of the more sophisticated sample programs.

Sample Programs are Included like:

- an EXPERT SYSTEM
- a NATURAL LANGUAGE INTERFACE
(it generates a dBASE II "DISPLAY" command)
- a GAME (it takes less than 1 page of PROLOG-86)

PROTOTYPE Ideas and Applications QUICKLY

1 or 2 pages of PROLOG is often equivalent to 10 or 15 pages in "C" or PASCAL. It is a different way of thinking.

Describe the FACTS and RULES without concern for what the computer will have to do. Maybe you will rewrite in another programming language when you are done.

Programming Experience is not required but a logical mind is. PROLOG-86 supports the de facto STANDARD established in "Programming in Prolog."

AVAILABILITY: PROLOG-86 runs on MSDOS, PCDOS, IBM AT or CPM-86 machines. We provide most formats. The price of PROLOG-86 is **only \$125.**

Full Refund if not
satisfied during
first 30 days.

800-821-2492

**Solution
Systems™**

335 Washington Street
Norwell, MA 02061
617-659-1571

CMD, INC.

Is Seeking AI Professionals in:

- Natural Language Processing
- Expert Systems
- Intelligent User Interfaces

CMD, INC. is located in Kendall Square, Cambridge's high-tech center across the Charles River from downtown Boston.

CMD, INC. has been profitable consistently, and is not a venture capital firm.

CMD, INC. is dedicated to high quality and well targeted custom software development for commercial applications, using AI technology.

CMD, INC. is staffed with very talented and very well trained professionals. We get involved closely with client staff in all phases of application identification, functional specs, system development, system installation, and training of designated staff. Our skill and experience mix allows us to interact effectively with clients' senior executives, domain experts, and systems people.

CMD, INC. cherishes and promotes talent and excellence, emphasizing effective career development opportunities for its staff. Salaries are more than competitive. Work is challenging.

Applicants should have:

- Experience with LISP and LISP machine environments.
- Strong academic preparation in Computer Science.
- Awareness of knowledge engineering and language representation techniques.
- Creativity and follow-through in problem solving.
- Commitment to developing quality software.
- Ability to work well with others.
- Experience preferred in implementing complex systems for turnkey delivery.

Send your resume to:

President

CMD, INC.

Post Office Box 1100

Cambridge, Massachusetts 02142

