

SURFACE: A PROTOTYPE EXPERT SYSTEM FOR SELECTING SURFACE ANALYSIS TECHNIQUES

Karl A. Smith* - Brian Farm** - Paul E. Johnson***

University of Minnesota

*Mineral Resources Research Center, Dept of Civil and Mineral Engineering

**Dept of Computer Science

***Department of Management Science

Abstract

SURFACE is a prototype expert system built for the personal computer to assist in the selection of surface analysis techniques for characterizing the surface of minerals or other materials. SURFACE uses a production system to represent the expertise required to reach a decision. The system knowledge base runs on an inference engine (shell) developed by Starfield and Adams.

SURFACE was built in a two stage process consisting of (1) development of a rule base for surface analysis from information in the literature and (2) testing and refinement of the rule base by several means, including interviews with human experts.

Introduction

Numerous factors must be considered in choosing appropriate surface analysis techniques for a particular sample. The major analytical techniques, Auger Electron Spectroscopy (AES), X-ray Photoelectron Spectroscopy (XPS), Ultraviolet Photoelectron Spectroscopy (UPS), and Secondary Ion Mass Spectroscopy (SIMS), all provide different types of information about the top five nanometers or less of the surface, and all affect the sample differently. The situation facing the novice user is graphically described by Sparrow & Drummond¹:

"Any newcomer to the field of surface science could easily become apprehensive deciding which technique to use for analysis of a particular sample or which instrumentation to invest in. The field is littered with acronyms, claims, and counterclaims which further confuse the decision-maker (p. 112)."

A surface may be described as that portion of a sample most directly affected by, or affecting, the external space around it. Generally, this includes a diffuse, weakly bonded layer of physically adsorbed gaseous material, a thin layer of chemisorbed atoms, and a reacted or treated specimen surface which is quite different chemically from the bulk of sample underlying it. The specimen surface often dictates the entire chemical and physical characteristics observed. Yet this sur-

face may only consist of a single layer of atoms several nanometers thick.

In general, problems involving surface analysis fall into one or more of the following categories: topography, morphology, elemental surface composition (including variations with depth), and chemical state. Elemental surface analysis of the specimen provides information such as:

1. Identification of the elements present.
2. Concentration of these elements.
3. Variation of concentration with surface depth.
4. Distribution of these elements across the surface.
5. Bonding nature of these elements.

Since information from these diverse areas cannot be obtained using a single instrumental technique, several different techniques may be required to characterize a particular surface or to solve a particular problem². The difficulty inherent in choosing appropriate surface analysis techniques coupled with the lack of available experts to assist in this choice stimulated our interest in building an expert system to address this problem.

Most knowledge-based expert systems developed to date have been relatively large-scale experimental prototypes, each requiring several man-years of effort to create^{3,4}. The procedure for building these large-scale expert systems has been described by Hayes-Roth⁵ and Hayes-Roth, Waterman, and Lenat⁶. Although there currently exist only a few references to small-scale expert systems (less than 100 rules), this area will surely be growing. An interesting microprocessor application of expert systems was described by Weiss, Kulikowski and Galen⁷ where they developed a scheme for interpreting the results from a scanning densitometer (a widely used medical instrument) using EXPERT and transferring the EXPERT interpretative system to a microprocessor environment. Another example of a relatively small expert system that has been used successfully is the DIPMETER ADVISOR developed by Schlumberger⁸. There are currently 90 rules in the DIPMETER ADVISOR system. It started with 30, increased to 150, and then dropped back to 90 with basically the same functionality as the larger system. As many as five or six of these rules work together to reach a

particular conclusion. DIPMETER ADVISOR took six years to develop, so that any improvement in the methodology for building small expert systems would be very helpful.

Background

The purpose of the SURFACE expert system building project was to develop a procedure for building an expert system from the literature and to apply this procedure to building a prototype system to provide prospective users of surface analytical services with guidance in selecting an appropriate elemental analysis tool (technique), or group of tools, and to provide information on how to acquire and use the recommended services.

Initially, the project was intended to provide information in three areas:

1. A catalog of elemental analysis tools for bulk and surface analysis, including information on the characteristics of the instrument, its location and availability, and the cost of the service.
2. The preparation of a prototype expert system, SURFACE, to serve as a decision aid to help users select an appropriate surface analysis technique wherein users specify the type of analysis desired, i.e., the aspect of the material to be characterized. This is accomplished through a series of questions asked by the expert system.
3. An analytical problem solver to aid the user in deciding how to characterize an unknown material, to help solve material fabrication and production problems, and to aid in studying the basic chemistry and physics of materials.

To date we have focussed on the first two steps of this overall objective. The third step involves poorly formulated, open-ended problems and was therefore quite difficult to tackle. Hence we concentrated our efforts on acquiring the information necessary to build a prototype expert system to assist in the selection of an analysis technique, given that the user was familiar with the type of material characterization information needed.

The users of the National Science Foundation Regional Surface Analysis Center at the University of Minnesota who could potentially benefit from an expert system to assist in the selection of surface analysis techniques include:

1. Graduate and undergraduate students learning about surface analysis methods or seeking analytical tools for their research.
2. Faculty researchers applying analytical services outside of their usual area of expertise.
3. Materials' developers and users seeking better ways to characterize their materials.
4. Industrial and state and federal agency researchers seeking to solve their surface analysis problems.

The rationale for the development of this

surface analysis expert system involved time saving and the maximization of resource utilization. It is anticipated that the use of SURFACE will reduce the time required to obtain appropriate analytical services and will minimize the use of inappropriate tests. SURFACE permits the choice of analytical tools from the complete range of available tools, not only from those tools with which the investigator is familiar. Finally, surface analysis is a rapidly growing area where expertise is distributed among too few experts. This expert system provides the prospective user assistance in selecting appropriate surface analysis techniques to solve routine problems, thereby permitting the few experts to spend more time on the tough and challenging problems where they are really needed.

Development of SURFACE

For the construction of the prototype expert system, SURFACE, the Starfield-Adams backward-chaining inference engine was used on an IBM Personal Computer. The Starfield-Adams shell and its applications are described in these proceedings⁹. Briefly, it is a rule-based expert system shell in which different knowledge bases can be stored in a flexible format in text files. The text files are called, parsed, checked and then interpreted by the shell. This format calls for a set of specific questions (each with a list of answers) and a set of rules (for drawing conclusions from the answers). This method of representing knowledge and storing it on file makes the expert system accessible to users with little or no experience in knowledge engineering. The shell can accommodate approximately 50 to 100 rules and can therefore be used to exercise small but non-trivial knowledge bases.

The procedure by which the Starfield-Adams shell accesses the knowledge base must be considered before and during the construction of the knowledge representation. The backward-chaining inference procedure works by validating decisions in the order they are listed in the text file. The order in which the questions are asked depends in turn on the order in which they are listed in the rules. The Starfield-Adams shell attempts to validate the first decision listed by looking for all the rules in which that decision appears, and then asking the questions in the order in which they appear in the rules.

The criteria for selecting appropriate surface analysis techniques are available in the literature. However, although most surface analysis problems are relatively routine and have been solved previously, the pertinent information is available in manuals, brochures and other technical reports that are not readily accessible and is not generally in the form needed by the user to make a decision. Moreover, the many different types of surface analysis equipment must be considered when determining whether a particular procedure is appropriate, complicating the decision-making process.

The first version of SURFACE was built from a representation, in table form, that contained

information on both imaging and analysis. The decisions (optical, SEM, AES, SAM, SIMS, and XPS) were listed with the following characteristics: sample requirements (size and form), resolution (depth and area analyzed), and analysis capabilities (elements, isotopes, detection limits, depth profile, quantitative capabilities and destructiveness). A production system with Decisions-Questions-Rules (DQR) was built from this representation in an intuitive manner¹⁰. It is interesting to note that the DQR file contained information that was not represented in the initial table. The performance testing of this version yielded some interesting results as described in the Performance Evaluation section. The advice of Hayes-Roth, Waterman, and Lenat⁶ to "throw away the Mark 1 version and start again" was followed for this work.

A reconstruction of the procedure that we used to develop the first version was recorded in an attempt to improve our understanding of the process. We then attempted to follow the procedure in order to build the second version of SURFACE and found some very interesting discrepancies. The task of rebuilding SURFACE revealed the need to revise the table representing the knowledge, one major revision being the separation of imaging and the surface analysis into two distinct sections. The rebuilding also revealed the implicit construction and use of a network starting with the questions and proceeding through the answers sequentially to arrive finally at the outcomes. The revised procedure was then used to create the second version of SURFACE and is described in the following section.

Procedure for Building Small Expert Systems from the Literature

Building a small expert system that will function with the DQR shell involves looking at or filtering the literature with a DQR template. Knowledge acquisition and representation from the literature for building SURFACE involved the following sequence of steps:

Task Definition:

1. Defining the task. Specifying the problem domain and listing the **decisions**, outcomes that the system will be expected to reach.

Information Collection:

2. Locating the relevant information in the literature by asking the question "What information is needed to reach the decisions listed?" The task is one of finding the information that forms the basis for (1) selecting techniques or making decisions, (2) formulating questions to be asked of the user and, (3) creating the rules in if-then statement form. Consulting many of the numerous articles which are available, e.g., "Comparison of surface analysis techniques," "Coordinated surface analysis," and "Surface analysis: selecting a technique," facilitated this task.

Information Organization:

3. Preparing a table of decisions in which all the characteristics found in the literature that are thought to be important in reaching a decision are listed.

4. Determining the possible **manifestations** of each characteristic, i.e., the various ways in which the characteristic can be defined or quantified.

5. Examining the table and selecting characteristics whose manifestations cover more than one decision and lead to a differentiation among the various decisions. These characteristics, that we call **traits**, can then be considered in formulating the rule base. The questions that will provide information on the manifestations of each characteristic can also be formulated at this time.

6. Constructing a network of all the questions with their associated answers, leading eventually to the final outcomes or decisions. This is a very helpful means of keeping track of all the possible routes through the knowledge representation. The construction of this network also simplifies the formulation of the rules.

Production System Preparation:

7. Connecting the questions and decisions by a series of rules. The preparation of the DQR text file in the order of Decisions, Questions, and Rules completed the preparation of SURFACE.

The details of the process that was used to construct SURFACE from the literature are described below. Accompanying this discussion is a detailed example of the procedure for developing a small subset of SURFACE, i.e., the procedure for selecting a surface imaging technique.

Task Definition

The task was defined in terms of the surface analysis techniques available at the University of Minnesota. These techniques include optical microscopy, SEM, AES, SAM, SIMS, and XPS. The objective was to acquire and represent the information necessary to decide among these techniques. Possible techniques for the imaging example are as follows:

- Optical microscopy
- Scanning electron microscopy (SEM)
- Scanning Auger microscopy (SAM)

Information Collection

Numerous articles were available in the literature to assist in the selection of surface imaging and analysis techniques. The first step was to locate the relevant literature from a database search supplemented by a manual search. Next the literature was scanned for descriptive and procedural information concerning the selection of

surface imaging and analysis techniques. Specifically, we looked for information that defined the characteristics for each technique which, in many cases, consisted of the specifications of the equipment involved. We also looked for procedural rules of the if-then form, or for information that could be converted to the if-then form.

Information Organization

When a human expert is consulted to build an expert system, the expert provides the questions that are typically asked when addressing a problem. However, a non-expert or pseudo-expert system builder (commonly called a knowledge engineer) will not know from experience which questions to ask, or when to ask them. Therefore, an external representation for storing and comparing the information and for imposing structure on the knowledge is needed to assist in determining appropriate questions. A table was used in the preparation of SURFACE, wherein all the decisions were listed along with each of the characteristics or events associated with each decision. The table included all of the aspects of each decision listed in the literature that we thought might be important in reaching a decision. A partial list of characteristics for the imaging example is as follows:

- sample size, form, and preparation requirements
- equipment cost, availability, ease of use, form of results, and ease of interpreting results
- resolution or magnification
- image type and features
- elemental analysis capabilities.

Following the completion of the characteristics table, the next step was to compare the characteristics for all decisions, looking for traits, which are characteristics that are defined or exist in some degree for more than one decision and are different for different decisions. Traits are characteristics that make a difference in reaching a decision, based on Bateson's definition of information as "a difference that makes a difference."¹¹ We considered two types of traits: (1) Traits of degree involving continuous variables, such as magnification, and 2) Traits of type involving information of a true-false nature, such as whether or not a machine has isotope analysis capability.

A new table was then formed by listing the decisions with the traits selected as likely to be useful in differentiating among the decisions. The traits selected for the imaging example are as follows:

- required sample size and form
- depth and lateral resolution
- magnification
- image type and features, and
- elemental analysis capabilities.

The knowledge representations, in the form of decisions by their traits, are listed in Tables 1

and 2 for surface imaging and surface analysis, respectively.

Representation Construction

Two steps were repeated to represent the knowledge required to reach a decision. First, the table constructed from the literature was examined to locate those traits that permitted differentiation among the decisions, as described above. Second, a network was constructed to illustrate the questions and their possible manifestations and to help trace or account for the possible decision routes through the representation.

The structured external knowledge representation (in table form in this case) was used to systematically create a knowledge network from which the production system was prepared. This process was begun by examining the traits, rather than the decisions or questions. The process that was used for the preparation of the network for SURFACE is as follows:

1. Each trait was examined leading to the selection of one trait that applied to all decisions-being-considered (for the first step this is "all decisions"). Traits were chosen which most evenly divided "decisions being considered," thereby producing a more effective search based on search algorithm efficiency. The first trait selected for the imaging example is:

- lateral resolution, which applies to all decisions at this point and whose manifestations have the interesting property that they overlap, i.e., more than one decision may be appropriate. For example, if a lateral resolution of 0.1 micrometer is acceptable, then SAM will be valid, and due to its greater resolution capability, SEM will also be valid.

A question was then written such that each manifestation of the selected trait could be selected as an answer. This step resulted in the following for the imaging example:

The trait "lateral resolution" applies to all decisions and its manifestations help to differentiate among the decisions. A question with answers corresponding to each manifestation for this trait is:

"What is the required lateral resolution?"

- greater than 1 micrometer
- less than 1 micrometer and greater than 0.1 micrometer
- less than 0.1 micrometer and greater than 0.025 micrometer

2. Under each manifestation were included all decisions that logically could be included as a subgraph. This information was recorded in a

network with a tree structure progressing from a question through its answers (manifestations) to the next level of decisions (see example in Figure 1). Note: if a decision appeared under more than one manifestation, it was written under each. However, the selection of traits for which each decision appears under only one manifestation will lead to a single decision more quickly.

3. Steps one and two were repeated for each new subgroup that still contained more than one decision. The **decisions being considered** at each step included only the undifferentiated decisions in the subgroup. Each subgroup was considered separately from others while applying the criteria in steps one and two for picking a trait. Note that it was not necessary or even desirable to ask the same question for all paths down the network at a given level. A new level was added to the network as each question was added. This process was continued along each path until a single decision resulted or until the applicable questions (traits) were exhausted. In the latter case, either the remaining subgroups have to be examined to find new information that will further distinguish among them until only one decision remains, or a compromise system that will only narrow down the number of decisions rather than arrive at a single decision will have to be accepted. The application of these steps to the imaging example is as follows:

The traits remaining for the example of selecting an imaging technique were required sample size and form, depth resolution, magnification, image type and features, and elemental analysis capabilities. The trait "image type" has several different manifestations that can be addressed in various ways. One can ask the following three questions, each of which has two possible answers: Is it necessary to detect color variations? Is it necessary to detect textural variations? Is a chemical map required? Alternatively, the information can be obtained from the following question "What type of imaging is required?" The possible answers to this question are:

- Physical mapping--color
- Physical mapping-- morphology
- chemical mapping.

These answers lead to a clear differentiation between the three imaging techniques being considered. However, if more than one type of image is sought, then more than one technique may be required. To determine whether this is the case, each question must be asked separately.

The knowledge representation network for the portion of the imaging example described above is shown in Figure 1. The repeated application of the procedure outlined above resulted in the knowledge

representation network for the second version of SURFACE.

Production System Preparation

Following the preparation of the knowledge representation in table and network form, the Decisions, Questions and Rules were formulated to run on the Starfield-Adams shell. The first step was to list all the outcomes, that is, all terminal nodes in the network (this list may include combinations of decisions) and enumerate each in order. The second step was to look at all the different questions, number each, and list each question with its answers. The rationale for each question may be listed here also. Finally, the rules were created by looking at each path in the network from initial question through final outcome. For each path a rule was written which contained each question and answers along the way.

The production system, in Decisions-Questions-Rules format for use on the Starfield-Adams shell, for the imaging example is as follows:

The four outcomes of this portion of the network are the terminal nodes listed above. They are then listed in the Decision format for the Starfield-Adams shell:

D1: "Optical, SEM and SAM are applicable".
D2: "SEM and SAM are applicable".
D3: "SAM applicable".
D4: "SEM applicable".

There are two questions for this small portion of the imaging example:

Q1: "What lateral resolution is required?"

Answers A1 "greater than 1 μm "
A2 "Between 0.1 and 1 μm "
A3 "Between 0.025 and 0.1 μm ".

Q2: "Is a chemical map required?"

Answers A1 "Yes"
A2 "No".

There are five rules for this small portion of the imaging example; the rule for the path in boldface print is:

R3 IF Q1A2 & Q2A1 THEN D3.

Initial Performance Evaluation

The performance of the first version of SURFACE has been tested by administering problems taken from the literature and tracing their progress from questions through decision(s). An initial set of problems was selected from a list of 21 problems prepared for administration to surface analysis experts. The four typical problems were:

1. Determine the composition and distribu-

tion (uniformity) of an organic coating on 7 micrometer diameter carbon fibers.

2. Determine the composition of red and blue dyes on an aluminum beverage container.

3. Determine the level of nitrogen in a sample of carbonitrided steel.

4. Determine the concentration and distribution (uniformity) of copper evaporated on the surface of graphite for a nominal 20 percent monolayer.

These four problems were administered to the first version of SURFACE. An average of eight questions was asked by SURFACE for the two problems where imaging was not requested (2 and 3) and an average of eleven was asked for the two problems where imaging was requested (1 and 4). For problem one, SURFACE recommended AES/SAM, whereas the literature recommended XPS. SURFACE did not arrive at a decision for problem two, and arrived at the same decision as recommended in the literature, AES/SAM, for problem three. For problem four, SURFACE recommended SIMS, whereas the literature recommended XPS. In three of the four cases, SURFACE prescribed either exactly the same procedure as recommended by the literature or by a human expert, or a procedure that was very close to that recommended.

The second version of SURFACE has not been tested as extensively as the first but a comparison between the two versions has revealed some interesting differences between them. The first version of SURFACE (1) was built intuitively, (2) was time-consuming and difficult to build, (3) does not always arrive at a decision, (4) combines imaging and analysis in a single knowledge representation, and (5) permits the validation of more than one decision. The second version of SURFACE (1) was built following the systematic procedure described earlier, (2) was very easy to build, (3) always arrives at a decision, (4) has separate knowledge representations for imaging and analysis, and (5) allows only one outcome to be validated.

Our next step is to build version three by combining the best features of each of the first two versions. We also expect to enhance our understanding by analyzing the hours of "thinking aloud protocol" that we have obtained from surface analysis experts. As we develop version three, we intend to revise the procedure to incorporate more of the features that expert problem solvers use intuitively.

Conclusions

The exercise of developing this prototype expert system pointed out the difficulty of acquiring and representing expertise for the routine problem of selecting surface analysis techniques which is done by experts every day. We feel there is a great potential for expert systems to contribute to higher quality problem solving and decision making, but feel that more work is needed to develop better procedures for knowledge acquisition and representation.

Acknowledgement

We gratefully acknowledge the financial support of the Mineral Resources Research Center. We thank the IBM Corporation for the use of personal computers made available under a contract to the Department of Civil and Mineral Engineering. Our thanks to David Potyondy, Ann Hossfeld and Leone Barnett for their thoughtful comments; to Pat Ryan, G.T. Swami, and Hilton Erskine for their participation; and to Carol Makkyla for assistance in preparing the manuscript.

References

- (1) G. R. Sparrow and I. W. Drummond, "Surface analysis: selecting a technique and an instrument," Industrial Research and Development, pp. 112-115, Aug. 1981.
- (2) T. P. Schreiber, "Coordinated surface analysis," American Laboratory, pp. 43-57, Mar. 1979.
- (3) R. O. Duda and E. H. Shortliffe, "Expert systems research," Science, no. 220, pp. 261-268, 1983.
- (4) R. Davis, "Amplifying expertise with expert systems," in The AI Business: Commercial Uses of Artificial Intelligence, P. H. Winston and K. A. Prendergast, eds., Cambridge, MA: MIT Press, pp. 17-40, 1984.
- (5) F. Hayes-Roth, "The knowledge-based expert system: A tutorial," Computer, vol. 17, no. 9, pp. 11-28, 1984.
- (6) F. Hayes-Roth, D. A. Waterman, and D. B. Lenat, Building Expert Systems, Reading, MA: Addison-Wesley, 1983.
- (7) S. M. Weiss, C. A. Kulikowski, and R. S. Galen, "Representing expertise in a computer program: The serum protein diagnostic program," The Journal of Clinical Laboratory Automation, vol. 3, no. 6, pp. 383-387, pp. 1983.
- (8) J. D. Baker, "DIPMETER ADVISOR: An expert log analysis system at Schlumberger, in The AI Business: Commercial Uses of Artificial Intelligence, P. H. Winston and K. A. Prendergast, eds., Cambridge, MA: MIT Press, pp. 51-66. 1984.
- (9) A. M. Starfield, S.R. Adams, and L. Bleloch, "A small expert system shell and its applications." Presented at this conference, 1984.
- (10) P. E. Johnson, "The expert mind: a new challenge for the information scientist," in Beyond Productivity: Information Systems Development for Organizational Effectiveness, Th. M. A. Bermelmans, ed., Holland: Elsevier, 1984.
- (11) G. Bateson, Mind and Nature: A Necessary Unity, New York: Morrow, 1979.

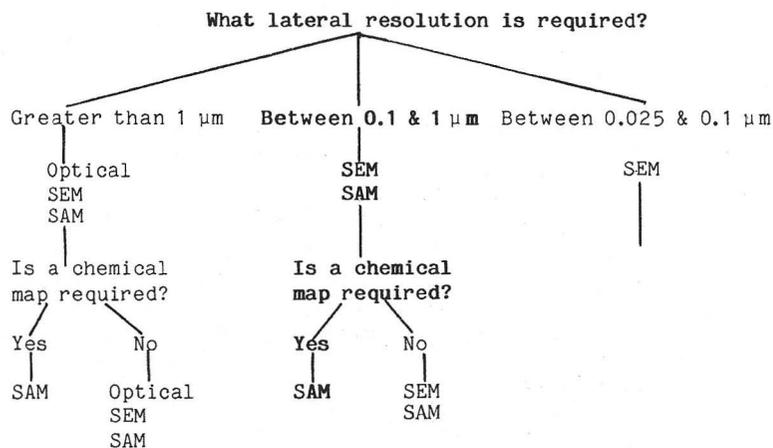


Figure 1. Knowledge Representation Network Example

TABLE 1
TRAITS FOR SURFACE IMAGING

Imaging Technique	Sample Requirements		Imaging					Chemical Analysis
	Size	Form	Lateral Resolution	Magnification	Image Type	Features		
Optical Microscopy	5x5x1 cm	Solid	10,000 Å (1 μm)	20x-1000x	Reflected or Transmitted Light	Color, Texture, and Grain Size	NA	
SEM (Scanning Electron Microscopy)	1x1x1	Solid Conductor or Solid Coated with Conductor	250 Å (0.25 μm)	20x-50,000x	Secondary Electrons	Morphology, Grain Size, Texture	Energy Dispersive x-ray Bulk Analysis Available for Elements with AN > 10. Sample penetration 10,000 Å.	
SAM (Scanning Auger Microscopy)	1x1x.1	Solid Conductor or Conducting	1000 Å (0.1 μm)		Secondary Electrons	Morphology, Grain Size and Texture	Surface Analysis (See Table 2)	

TABLE 2
TRAITS FOR SURFACE ANALYSIS

Analysis Technique	Sample Requirements			Resolution		Analysis Capabilities				
	Size	State	Other	Depth	Area	Elements	Detection Limits	Other Chemical Information	Quantitative Capability	Effect on Sample
AES (Auger Electron Spectroscopy)	1x1x0.1	Solid	Conductor or TF on Conductor	20-50 Å	0.05 cm	≥ Li	0.1%	None	Few %	Little, if Conductor
SAM (Scanning Auger Microscopy)	1x1x0.1	Solid	Conductor or TF on Conductor	5-50 Å	5x10 ⁻⁶ cm	≥ Li	0.1%	None	Few %	Little, if Conductor
SIMS (Secondary Ion Mass Spectrometer)	1x1x0.1	Solid	Semiconductor or Conductor	2-6 Å	0.1 cm	All	0.0001 %	Isotopes	Poor	Destroy
XPS (x-ray Photoelectron Spectroscopy)	1x1x0.1	Solid	None	20-50 Å	0.5 cm	≥ Li	1%	Bonding, Oxidation State, Compounds	Few %	None

A SMALL EXPERT SYSTEM SHELL AND ITS APPLICATIONS

A.M. Starfield* - S.R. Adams** - A.L. Bleloch***

*Dept of Applied Mathematics, University of the Witwatersrand and
Dept of Civil and Mineral Engineering, University of Minnesota

**Dept of Computer Science, University of the Witwatersrand

***Dept of Physics, University of the Witwatersrand

Abstract

A simple rule based expert system shell has been developed for use on a microcomputer. Different knowledge bases can be stored in a flexible format in text files which are called, parsed, checked and then interpreted by the shell. This format calls for a set of specific questions (each with a list of answers) and a set of rules (for drawing conclusions from the answers). This way of representing knowledge and storing it on file makes the expert system accessible to users with little or no experience of knowledge engineering. The shell can accommodate approximately 50 to 100 rules and can therefore be used to exercise small but non-trivial knowledge bases. It has been used by the authors in resource management workshops and by undergraduate engineering students to implement their own knowledge bases. These and other applications are discussed. The potential use of the shell as a vehicle for more effectively conveying information usually found in professional handbooks is also discussed.

Introduction

The field of knowledge engineering is dominated by large expert systems such as MYCIN, DEN-DRAL, R1 and PROSPECTOR.^{1,2} These systems have served the vital purpose of demonstrating that expert systems do indeed have important practical applications. However, because they are so well known, they have also moulded the perceptions of potential users of expert systems; in particular, they have created the impression that:

- (i) successful expert systems require large and complete knowledge bases, and
- (ii) they require many man-years of effort to construct.

Probably as a result of these perceptions, the idea of a small expert system appears to be almost a contradiction in terms. Given the explosive growth in microcomputer software over the last few years, relatively few expert systems have been developed for implementation on microcomputers. Duda and Gaschnig³ published an article that included a BASIC program that would run on a microcomputer. The purpose of the program was to illustrate some of the principles behind expert

systems rather than to provide a general purpose package. Teknowledge Inc. is marketing a program called M.1⁴ which runs on an IBM personal computer with 128 kilobytes of memory, but it is intended to be used as a tool for determining the feasibility of embarking on a large and expensive knowledge engineering project. It is not suggested that M.1 should be used for building working expert systems. Michie has written a microcomputer expert system called "Expert-Ease" which also runs on a 128K IBM PC. It is perhaps closest in concept to the system and ideas to be presented in this paper.

It is of course true that large systems are needed to tackle large and unfactorisable knowledge domains. The thesis of this paper, however, is that small systems (incorporating tens rather than hundreds of rules) are not just prototypes of larger systems; they have a number of important and exciting uses and are worth thinking about and studying in their own right, particularly from the point of view of knowledge acquisition.

This interest in small expert systems grew from the senior author's experiences in the field of ecological modelling and the frustration of trying to build conventional numerical models to assist wildlife managers in their decision making. On the one hand, the numerical models required information about processes and rates of change that the biologists could not provide; on the other hand, the biologists were in practice reacting to field observations that were very different from the substance of the models being built. For example, noticing when the leaves fall off deciduous trees can have important implications for the management of browsing animals, but building a simulation model which makes the leaves fall off in a realistic manner is a formidable task. What was needed was a form of qualitative modelling, and expert systems seemed to provide a format for just that.⁵ Efforts to explain this form of qualitative modelling to biologists led to the need for a small expert system shell, one that could run on a microcomputer and that would be easy to use for people with no experience of knowledge engineering and the minimum of experience with computers. Building the shell, in turn, led to thoughts about other potential uses for it.

In this paper we shall first describe the

shell itself and then discuss some of our experiences in using it and thoughts on its potential uses.

Description of the Knowledge Base

By knowledge base we mean a data set which is to be fed into the expert system shell; it contains three types of entity: decisions, questions and rules. The set of decisions defines the domain of the knowledge base by explicitly stating the "conclusions" that the expert system can reach. If a decision is declared without an explicit "conclusion," then that decision represents an intermediate or non-terminal conclusion.

The questions define the information that can be requested from the user. Each question has a list of possible answers. When the shell puts the question to the user, the user must choose an answer from this list. The answer is referred to by its numerical position in the list.

A rule consists of an antecedent and a consequent. The antecedent takes the form of a condition or logical expression, which is usually constructed from simpler logical expressions joined with the operations AND, OR and NOT. The consequent is a reference to a decision.

Each question and rule may have an explanation or reason attached to it.

The knowledge base is stored in a text file. The standard microcomputer text editor is used to create and modify this file. The syntax for the decisions, questions and rules has a loose format and is easily learnt. A novice thus only has to know how to use the editor and this simple syntax to be able to build a knowledge base. The knowledge base is then exercised via the expert system shell.

A Description of the Expert System Shell

The shell program was originally written on an Apple IIc microcomputer using UCSD Pascal. It is transportable and has been successfully moved to an IBM personal computer.

The shell program consists of four sections as depicted in Figure 1. They are:

1. A scanner which reads and classifies items in the input.
2. A parser which builds an internal representation of the knowledge base.
3. An auditor or semantic checker which tests the integrity of the knowledge base.
4. A decision validator which uses the knowledge base to advise/help the user.

The reader may notice the similarity between this structure and that of a compiler.

The scanner reads in the knowledge base and classifies the text into numbers, keywords and character strings. The output of the scanner is a

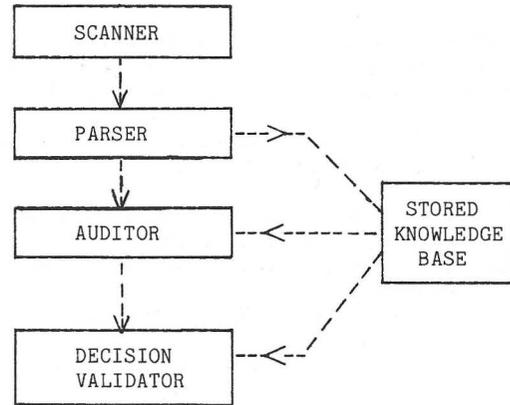


Figure 1. Functional diagram of the Expert System Shell.

sequence of tokens representing these items. Spaces, blank lines and comments are ignored by the scanner.

The parser builds an internal representation of the knowledge base which is stored as a collection of tree structures. Each decision, question and rule is stored in a separate tree. Since the knowledge base has a LL(1) grammar, the parser is of a simple recursive-descent design. The tree structures are literal representations of the parse trees generated by the grammar.

Syntax errors in the knowledge base are accurately detected, and, due to the simplicity of the grammar, reliably diagnosed. Recovery from errors is primitive but usually effective. The remainder of the faulty construct is ignored and parsing continues after the mandatory period (i.e., ".") that follows a decision, question or rule.

The auditor performs semantic checking on the knowledge base. Certain integrity tests are carried out, including tests for referential integrity (i.e., that all questions and decisions that are used in rules have actually been defined). Two or more decisions could be mutually exclusive, where if one decision is valid, then the others must be invalid. Currently the shell cannot check the integrity of the knowledge base in this respect; it has no way of determining which decisions are mutually exclusive.

The decision validator is the heart of the system. It may be used in one of two modes--single decision mode or general mode. In single decision mode the user of the shell program chooses a decision from the knowledge base as a candidate for validation. The decision validator will then use the rest of the knowledge base to determine whether or not that decision is valid. In general mode the shell will test decisions until either a valid decision is found or all the rules have been exhausted.

A decision is valid if there is at least one rule of the form

"IF condition THEN DECISION (number)."

for which the condition evaluates to true. If no rule referencing the decision has a true condition, then the decision is declared invalid.

The condition is a logical expression and may assume one of the following forms:

1. a reference to a question and answer
2. a reference to a decision
3. a disjunction (inclusive OR) of two expressions
4. a conjunction (AND) of two expressions or
5. the logical negation of an expression.

A reference to a question and answer is "true" if the question has that answer, and false if the question has another answer. The shell remembers the answer to a question so that the user is not unnecessarily asked the same question twice during the validation.

If an expression is a decision reference, then it is true only if that decision is valid. When the validity of the decision is known, it can be used immediately; otherwise, the shell will (recursively) attempt to validate the decision before continuing to validate the current decision.

In the case where an expression is a conjunction or disjunction of two expressions, the first expression is evaluated. The second expression is evaluated only if the result cannot be determined from the first expression. This prevents the asking of redundant questions. The following identities are used:

conjunction:	false AND x = false true AND x = x
disjunction:	false OR x = x true OR x = true.

As we shall see, an important feature of small expert systems is their ability to provide explanations. The shell has an explanation feature that is invoked whenever the user responds with "why" to a question. Repeated "why" responses trace back through the program's chain of logic. The shell obtains the explanations from the knowledge base in the following manner.

Each question and rule may have a reason associated with it. The reason is specified by the keyword WHY followed by a string. When the question or rule is used, the reason is pushed onto a stack, and it is only removed when the shell has finished with that question or rule. If the user responds with "why" to a question, then the shell inspects the stack and gives the reason on top of the stack. Successive reasons in response to repeated user queries are taken from successive elements of the stack until the bottom of the stack is reached. This approach has the advantage that the reasons are determined by the knowledge base writer and may be oriented towards the users of the system. The reasons may be

couched in terms of the users' problem rather than the solution process. It was simple to implement. One of the disadvantages is that the system cannot verify the reasons and poorly written reasons may give a deceptive impression of the chain of logic.

The expert system shell is a compact program and leaves a lot of room for extension. One possible extension would be the addition of a third truth value "Unknown." The shell could then operate on incomplete data. A more fundamental change would be to abstract away from questions and decisions towards a system of problem attributes which have values in a domain. We will not, however, explore these ideas here.

Applications in Resource Management

Expert systems usually call on expertise in a particular, possibly narrow, field. In contrast, decisions in resource management often call for expertise or information from a number of very different fields. We will consider two examples here: the first in the management of a herbivore population in a game preserve, the second in fixing the catch size for a fishing industry.

A decision on whether or not to harvest part of a herbivore population in a game preserve can depend on:

1. The objectives of the game preserve
2. The size and physiological condition of the herbivore population
3. The state of the vegetation
4. Competition between the target population and other rare or protected species
5. Weather or rainfall patterns
6. Economic considerations.

Interpreting the information that impinges on the size of a fish population in order to determine a fishing quota might depend on:

1. Predictions of computer models
2. Data from echo soundings
3. The previous year's catch
4. Data from sea-birds
5. Indications of egg-predation
6. Environmental factors, e.g., currents and winds
7. Political and economic factors.

Small expert systems can be built to address and combine the different issues and information sources relevant to either of these problems. They can most easily be constructed in small workshops, where participants not only contribute their particular expertise, but also interact with experts in different fields. The procedure in a typical workshop might be:

1. First introduce the concept of an expert system and demonstrate a working system on the microcomputer.
2. Explain the structure of questions, answers, decisions and IF-THEN rules.

3. Elicit a list of appropriate decisions; these define the scope of the system that is to be built.

4. Start collecting appropriate questions. A useful device here is to ask the participants what questions they would ask over the phone if they were on vacation and their stand-in called them for assistance.

5. Start constructing rules. This is where the structure of IF-THEN rules has very real advantages over devices such as decision trees: one does not have to grasp the nature of the whole decision process; all one has to do is get agreement on whether or not each particular rule is true and appropriate. Often bottlenecks in constructing rules can be overcome by suggesting a rule that is obviously too simple (e.g., If the echo-soundings indicate an increase in the size of shoals as compared with the previous year, then increase the fishing quota accordingly). Exploring the limitations of a set of trivial rules can lead to a non-trivial rule base.

6. When the rule base looks promising, implement the system via the shell and exercise it; it soon becomes apparent where the rule base is deficient.

At all times attention should be paid to the explanations given for both questions and rules.

There are a number of points to be made in connection with workshops of this type and the expert systems that emerge from them:

1. Decisions affecting resource management are often controversial. Building an expert system is a cooperative venture where workshop participants tend to argue constructively; the structure of the rules encourages compromise. Just building a system, even if it is never used, can be an extremely useful exercise.

2. The purpose of the expert system is to determine how to interpret and weigh different information, how to balance different considerations, and how to trade off different objectives. It draws on only specific aspects of each expert's knowledge. For example, the expert on sea-birds will have to learn during the workshop that nobody is interested in what he knows about their physiology or population dynamics; the important questions are likely to be of the form "if 50% of a gannet's diet is found to be anchovy, what does this tell us about the anchovy shoals?"

3. Our experience⁵ is that the response of workshop participants is always positive; invariably, going through the formal process of constructing an expert system forces clear thinking. We have, for example, conducted a workshop on game management at the end of a three-day meeting during which the participants had exhaustively discussed their management strategy; building the system highlighted a number of points they had overlooked.

4. The explanation facility can play a vital role in communication from one level of decision making to another (e.g., the scientists communicating to the managers or politicians) or from the professionals to the public (e.g., showing how controversial decisions are actually reached).

5. It is unlikely that a useable expert system will emerge at the end of a one or two day workshop, but a smaller team should be able to take away a prototype and improve it.

6. Finally, perhaps the most important point is that the objective is not to produce the definitive expert system; the approach is adaptive and the idea is to build a system and then refine and alter it with use and experience. Its purpose is not so much to produce answers as to focus arguments and promote sensible, logical discussion and decision making. Here again the explanation facility is vital.

Applications in the Classroom

In many professional subjects the student is taught facts, techniques and methods of analysis, but is not formally taught how to synthesize what he has learned. Thus an engineering student might be at a loss when he finally has to design something, while a medical student might be bewildered by the difference between what he sees on a ward round and what he reads in a textbook. Teaching (and testing) facts and techniques is relatively easy and can be accomplished via the usual routine of textbooks, lectures, homework assignments, mid-quarter tests and multiple answer quizzes. Teaching synthesis is more of an art.

Asking students to construct a small expert system provides a formal mechanism for teaching synthesis. The following has been found to be an effective approach⁶:

1. Introduce the concept of an expert system in the classroom, explain the format and the structure of IF-THEN rules, and demonstrate a small system on a microcomputer.

2. Divide the class into groups of 2 or 3 and ask each group to suggest a few suitable topics for expert systems. Discussing what they suggest can usefully highlight the difference between those problems that are best solved by formal analysis and those that lend themselves to a more qualitative, rule-based approach.

3. As a homework assignment (over one or two weeks) ask each group to build the knowledge base for an expert system and implement it, using a shell, on a microcomputer. Care should be taken to ensure that the scope of each assignment is not too broad. (For example, in a senior mining engineering class students built expert systems on such topics as what equipment to use to remove the overburden from an open-pit mine, and what type of explosive to use to break rock under various conditions.) The students should be told to pay particular care to the explanation facility.

4. Invite faculty to use and critique the systems produced by the students and also encourage students to run and critique the systems produced by their peers.

Students have been found to be unusually enthusiastic about this type of assignment; they feel they are doing something that is both challenging and worthwhile. The formal structure of the expert system imposes a discipline which encourages them to read, argue and consult faculty in a purposeful manner. It forces them to identify and concentrate on the essential features of their problem. Implementing the system on a microcomputer is far more exciting and meaningful than writing an essay. (It is also easier for faculty to grade!) Most importantly, though, the exercise introduces students explicitly to the thought processes they will have to acquire as professionals.

The Intelligent Handbook

Professional handbooks contain a variety of information, some of it factual and some of an algorithmic (how to do it or what to take into consideration) nature. Thus a mining engineering handbook might well contain a section on how to choose an appropriate explosive or a suitable method for the removal of the overburden for an open-pit mine. Both of these are examples that we quoted in the previous section to illustrate how small expert systems might be used to teach students how to synthesize information. What are the advantages of replacing the handbook text by small expert systems that address questions of this kind?

Articles in handbooks are often well written, but even then the user can have difficulties in relating what he reads to his particular problem. The difficulty is not so much with what is in the article, but what is left out. A small expert system, on the other hand, is unambiguous. If thoughtfully constructed, it should guide the user to his particular solution. Whereas the handbook article reads as though it is solving problems in general, the expert system behaves as though it is giving personal attention to the user. It can be more probing and specific than the handbook article. In fact, our experience in workshops suggests that the format of the expert system is likely to lead to a more useful algorithm; there is less chance of information being left out. Finally, more difficult problems could be addressed. An example is given in a companion paper by Smith, Farm and Johnson.⁷

It is worth looking at a scenario for a computerized handbook a little more closely. The end user would have a microcomputer with software containing a shell similar to the one described in this paper. A set of disks would then be available with up-to-date knowledge bases for specific purposes and the user would interact with them via the shell. An immediate advantage of this approach is that it becomes so easy to update the knowledge bases and to add to the set of disks. A second and more subtle advantage is psychological:

a user can interact with the computer program (particularly if the explanations given by the system are informative), but he can only read a handbook article. As a result, he is likely to find a more thoughtful solution to his problem via the expert system.

The above approach has more general applications than the intelligent handbook. The microcomputer with an expert system shell provides a powerful format for communication between specialists and generalists. It would have obvious uses in agricultural extension activities. It also provides an immediate link between researchers and practitioners in any organization. Imagine, for example, a large mining company with a central research organization and a number of mines scattered all over the world. A new idea is far more likely to reach the practicing engineer (and be used by him) if it is sent out from the research organization in the form of a knowledge base (with good explanations) on a disk that is read via an expert system shell than if it is sent out as a research report.

Concluding Remarks

We have described a shell for small expert systems and explored some of its uses. It should be noticed how the applications all exploit the two most important features of the shell, namely the explanation feature and the ability to enter a knowledge base via a text file using the standard microcomputer editor.

While the examples given in this paper have been drawn from the fields of resource management and engineering, they should be meaningful in many other areas. The last example, for instance, of a mining research organization communicating with engineers on individual mines could equally well apply to tax experts communicating with the accountants in a large company.

The examples we have given suggest a number of points about small expert systems that are worth emphasizing:

1. The process of constructing the knowledge base for a small expert system can be a useful end in itself, both in the classroom and as a technique (similar in intent to delphi techniques) to promote cooperation and intelligent compromise.

2. Knowledge acquisition for a small expert system is likely to be quite different from what has been learned about knowledge acquisition for large systems. The workshop is likely to be a useful device for designing a knowledge base and building a preliminary system. Small systems lend themselves to an approach of starting with a core set of rules and then modifying and adapting them via interactions between users.

3. Small expert systems are in essence communication devices rather than inference machines. They provide a powerful means of communication, be it two-way communication between con-

flicting interests in a workshop environment or one-way communication from the specialist to the generalist.

Undoubtably, there are many more potential uses for and attributes of small expert systems. This paper will have served its purpose if it provokes others into looking more carefully at the small expert system as a discipline in its own right.

References

- (1) R. O. Duda and E. H. Shortliffe, "Expert Systems Research," Science, vol. 220, no. 4504, pp. 261-268, 1983.
- (2) F. Hayes-Roth, D. Waterman, and D. Lenat, eds., Building Expert Systems. New York: Addison-Wesley, 1983.
- (3) R. O. Duda and J. G. Gaschnig, "Knowledge-based expert systems come of age," Byte, pp. 238-278, Sept. 1976.
- (4) J. Markoff and E. Shapiro, "A little knowledge," Byte, pp. 393-395, Sept. 1984.
- (5) A. M. Starfield and A. L. Bleloch, "Expert systems: an approach to problems in ecological management that are difficult to quantify," Int. J. Environmental Management, pp. 261-268, 1983.
- (6) A. M. Starfield, K.L. Butala, M.M. England, and K.A. Smith, "Mastering engineering concepts by building an expert system," Engineering Education, pp. 104-107, 1983.
- (7) K.A. Smith, B. Farm and P.E. Johnson, "SURFACE: A prototype expert system for selecting surface analysis techniques." Presented at this conference, 1984.