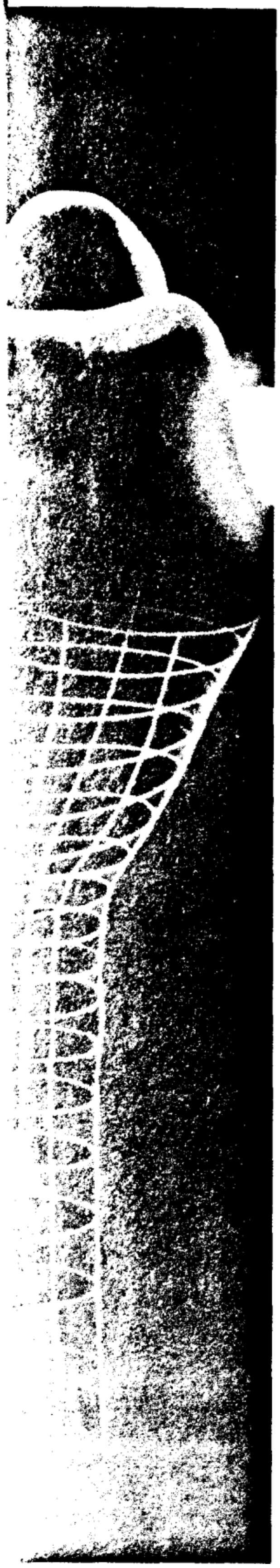
BY BRUCE MARCOT

TES I

testing your knowledge base

Now



In the development of a knowledge base, eventually the time comes to put its contents to the test. This article describes how knowledge bases can be tested and presents a set of validation criteria. Although knowledge bases take many forms, this article focuses on those founded on production rules. However, the general concepts and criteria discussed here apply to other types of knowledge bases as well.

I draw on my own experience developing and testing knowledge-based systems for Perkin-Elmer Corp., where I helped develop an expert system to monitor respiration conditions of patients.[1] I also worked for Oregon State University, Corvallis, Ore., and the U.S. Dept. of Agriculture - Forest Service developing expert systems to assess conditions of wildlife habitat[2] on USDA Forest Service lands.

Testing a knowledge base is not a trivial matter, regardless of the exactitude with which the problem was defined and the production rules (with associated probabilities) were devised. Testing cuts to the heart of why the knowledge base was initially developed and how one expects it to be used.

WHAT REQUIRES TESTING?

At best, testing a knowledge base is more an operant philosophy of programming than a discrete stage in the development of production rules. It is good programming practice to test any program in various ways throughout its development. Testing should be integrated into the development-application cycle.

Generally, there are two levels of testing any computer code: verification and validation. Verification involves insuring that the computer code—whether it is LISP code or a rule base written in some expert system shell—is written without bugs. This level of testing is relatively straightforward and may be pursued on a regular basis as the knowledge base is developed and revised. Many commercially available expert system shells have tracing functions or filters for catching common errors in syntax or rule redundancy. These tools are of immense value when the knowledge base approaches the size of even several dozen rules.

Validation, on the other hand, involves the more deceptively difficult task of insuring that the meaning and content of the rules meet some carefully defined criteria of adequacy. Defining such criteria is the key to successfully conducting a validation procedure and demonstrating the level of acceptability of the knowledge base.

Performance may be tested progressively as a knowledge base is built. For example, S.M. Weiss and C.A. Kulikowski[3] suggest testing the initial model design, knowledge base data, system performance, model refinements, and the effect of model changes on case conclusions.

Development of a knowledge base commonly proceeds through four phases: creating the prototype, developing the first generation rule set, testing and expanding the rule set to the second generation, and testing the second generation rule set. Attention must also be given to how well the system might be marketed and used. Specific criteria and procedures for testing the valid-

ING

0

Testing cuts to the heart of why a knowledge base was designed and how one expects it to be used

ity of the knowledge base (see the sidebar to this article entitled "Validation Criteria") should be integrated within each of these phases.

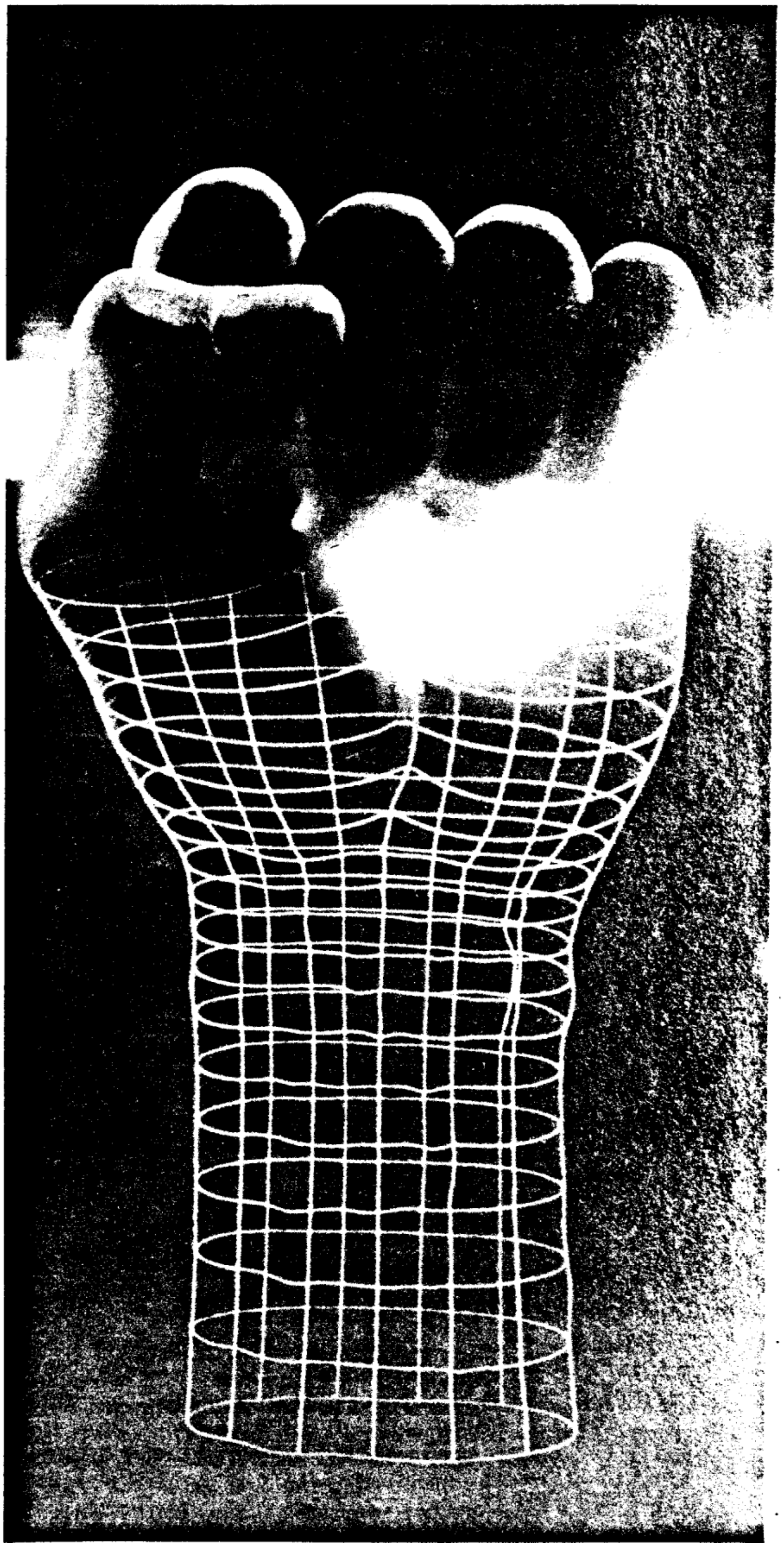
CREATING THE PROTOTYPE

Validation should be part of the initial stage of creating the prototype knowledge base. Prototypes that typically consist of only a dozen or so rules demonstrate the application and feasibility of a knowledge engineering approach to the problem at hand. Validating the prototype is integral to defining the original problem and developing the first set of rules to frame the inference structure.

When testing the prototype, it is important to determine the specific domain within which the system should operate. More fundamentally, testing the prototype should involve asking whether the original problem domain was defined narrowly enough. Knowledge base projects have a greater likelihood of succeeding—and, in a sense, of being valid—when they address a narrowly defined problem.

The prototype should also show a fair degree of usefulness to demonstrate the desirability of continuing work on a fuller system. A system is useful when it contains necessary and adequate parameters to solve at least some problems. Today's expert system shells are so user-oriented that bogus rule bases may be readily developed, creating what might be called "amateur systems." Although usefulness may be judged subjectively, it is more than just a trivial or self-evident criterion.

PHOTO BY MICHEL TCHEREVKOFF COURTESY OF BELL ATLANTIC



At the alpha testing stage it is important to determine the accuracy and adequacy of the system

THE FIRST GENERATION RULE SET After the prototyping stage, the first generation rule set typically consists of several dozen to a few hundred rules. The main goals of validation at this stage are to reexamine the original objectives, more precisely determine the problem domain, and establish the degree of detail desired in the system.

At this alpha-testing stage it is important to determine the accuracy and adequacy of the system. Accuracy is measured by comparing the number of correct predictions with known data. It may be assessed statistically with a ^{X2} or goodness-of-fit model that compares observed proportions of correct answers (the fraction of predictions that were observed to be empirically correct) with expected proportions (the desired rate of accurate predictions).

Adequacy, on the other hand, is a measurement of the fraction of actual conditions included in the system. For example, the breath gas monitoring system I helped develop diagnosed 12 waveform patterns of CO2. Each waveform pattern corresponded to a particular physiological condition. The adequacy of the system was judged according to the total desired number of conditions to diagnose.

Adequacy may be expressed as a simple fraction. The breath gas monitoring system was able to diagnose 12 of 27 important physiological conditions, so the system was 100 X (12/27) = 44% adequate. As an alternative, subjective weights may be added to particular conditions that are more important to recognize (such as cessation of breathing).

Another question at the alpha-testing stage concerns the degree of precision required. Precision may be measured as the capacity of the knowledge base to predict, diagnose, classify, or monitor within a specified statistical confidence interval. Precision is also a measure of the number of significant figures used in calculations and the error of estimate of the parameters. Precision error may be expressed as the standard error or confidence interval of the observed or inferred values of parameters.

The alpha-testing stage also entails statistical tests of the reliability of the inferences, diagnoses, or classifications made by the knowledge base. Evaluating reliability may involve a complex series of statistical tests that apply prior knowledge to empirical evidence, such as in the use of Bayesian statistics.[3,4] A discussion of such tests is beyond the scope of this article, but the reader is encouraged to explore the concepts.

EXPANDING THE RULE SET

Alpha-testing the first generation rule set should result in revision of the objectives for developing the fuller rule base. As rules are added and amended at this stage, performance standards that determine the characteristics of the full-scale system should be carefully described, especially for determining utility or ultimate marketability. These standards include the flexibility or adaptability of the system to future applications. An adaptable system can be readily enhanced and its user interface modified as contexts warrant.

Expanding the rule set to full scale should also identify the number of parameters necessary to address the problem domain (resolution) and the desired complexity (wholeness) of the knowledge base. Expansion typically involves recrafting inference structures, high-level control rules, and lower-level facts and relationships among facts. To this end, determining the robustness and sensitivity of conclusions to rules and variables helps direct the knowledge engineering efforts. Several workers have explored using empirical information for modifying the rule base (as used in SEEK).[5,6]

SECOND GENERATION RULE SET

Testing of the first full-scale knowledge base is often done directly on the site where the system will be used. Determining the accuracy, precision, and reliability of the full-scale rule base is necessary to ultimately determine the specific contexts within which the user should expect the system to work well (technical and operational validity). This beta-testing should also involve determining the usability of the system by the intended audience, including how well the system fits into existing procedural and administrative structures.

Testing the audience, so to speak, was one of the most important limitations in developing and implementing a knowledge engineering approach to solving problems of natural resource management in the USDA Forest Service. The weakest facet of the knowledge base approach for diagnosing and ameliorating the condition of wildlife habitat on National Forest lands was selling the basic idea to management.

Existing administrative processes for evaluating wildlife habitat simply did not allow for this new approach. Thus the knowledge base was, in an important sense, invalid. The solution in this case was to educate veteran supervisors and managers within the agency as to the utility of such a system and revise the general habitat evaluation process.

MARKETABILITY

In this article the term "marketability" refers not only to the specifics of selling a product in the marketplace but also to selling the idea and use of a system to people who may be reluctant to adopt it in the course of their work. Selling a system involves determining the practicability and utility of a system in specific already-functioning work environments. From the commercial perspective, this may also include market feasibility studies to determine desirability and availability. The system must also be adaptable to different work environments, changes in integrated hardware and software, and changing information needs over time.

If a system is to be used by technicians, professionals, administrators, or managers in the course of their work, it should be appealing and credible (face validity). In the development of the breath gas analysis system, we contacted recognized medical specialists in the field of breath gas analysis during the early phases of developing the prototype. The appeal and face validity of the prototype was important for securing research and development funds within the corporation and for helping sell the system to customers once the fuller system became available.

Another facet of these less-tangible criteria of validity concerns how the human user is to be integrated into the knowledge system. Users of knowledge-based monitoring systems will want to remain an integral part of the information-gathering/analysis/interpretation cycle. For example, the breath gas monitoring system will be more readily accepted and used if the clinician has a role in interpreting and at times overriding the system's diagnoses and conclusions.

How specifically can these various validation criteria be tested? The main steps in conducting validation tests under each of the phases of developing the knowledge base include:

- defining the domain and context within which a system is expected to perform well and thus the contexts in which its performance is poor or unknown.
- identifying specific performance criteria for validation.
- conducting the validation tests and analyzing and evaluating the results.

DOMAIN AND CONTEXT

Like any model, the domain and context of a problem determine how well a knowledge base can be expected to perform. Applying the model outside the arena for which it was intended will likely produce unreliable and poor performance.

The creator of the model should specify precisely the conditions under which the model is to be used. The audience must adhere to these conditions for the model to insure predictable levels of performance. This proved especially important in the expert system that evaluated wildlife habitat. Similar habitats in different geographic areas have vastly different species of wildlife associated with them, and the expert system failed to predict reliably outside of the area in which it was developed.

Testing a knowledge base may also involve assessing its generality or breadth. Generality is the range of contexts within which a system can be expected to perform reliably. Breadth is the number of conditions and parameters an expert system contains, proportional to the number of rules and clauses in its knowledge base.

A system should be broad when the range of conditions and contexts within which it should operate reliably is wide. Such testing involves applying the knowledge system to problems outside the specific confines of its domain under controlled circumstances and observing the accuracy or reliability of the outcomes.

Identifying a specific problem domain may also entail describing the number and kinds of variables chosen to represent each component of the knowledge structure (depth). This in turn helps identify the degree of realism of the knowledge model; that is, which relevant variables and relations have been included in the knowledge base. However, it may not be possible to simultaneously maximize realism and generality. Understanding how well a system performs under both criteria is essential for developing realistic expectations about its performance.

Although a system like the breath gas monitoring system must perform reliably, the realism of the production rules is equally important. Because a system like this one will be closely scrutinized by many medical experts, even if it performs reliably, the variables and relations expressed in each specific rule must conform to generally accepted expert understanding.

This is particularly important with knowledge bases such as medical monitoring systems, where legality and culpability are significant issues. For other systems, such as the wildlife habitat expert system, the degree of realism of the rules is secondary to insuring that the system provides reasonable and useful conclusions.

SPECIFIC PERFORMANCE CRITERIA

Performance criteria must be specified for each stage in the development of a knowledge base. For example, the degree of accuracy of the system may be assessed during testing of the second generation rule set.

Acceptable levels of each performance standard should be determined before the tests are conducted. What allowable fraction of predictions (outputs) may be in error? How accurate or precise do diagnoses or classifications have to be for a system to

If human experts incur a 15% error rate, should the system be expected to perform better?

be acceptable? How realistic should the variables and relations be within each rule?

In determining the overall validity of a system, it is instructive to determine how well human experts do in the problem area and to thereby create reasonable expectations of the system's performance. If human experts incur a 15% error rate, should the system be expected to perform better? If so, how much better, and why?

A Turing test of the output of a system may help determine its overall validity.[7] In a Turing test, human experts (evaluators) are given results of running the knowledge base model for a specific problem and results furnished by a human expert for the same problem. Both sets of results are unlabeled.

The degree to which the evaluators can distinguish between results from the model and the human expert is a test of how well the system mimicked human performance. Furthermore, having the evaluators explain exactly how they distinguished model from human performance can help pinpoint. which model parameters require further refinement.

Another performance criterion that may be tested is to analyze the type of errors a rule set produces. The conclusions of particular rules—or an entire knowledge base—for a given run may be wrong in two ways. They may fail to accept what is actually a correct conclusion (a type I error) or accept and report what is actually an incorrect conclusion (a type II error).

These two types of error may have vastly different implications, depending on the purpose of the knowledge base. For example, the breath gas monitoring system alerts the clinician when a serious problem, such as increased CO, content of breath gas or cessation of breathing (anoxia), has been detected. False alarms (type 11 errors) would be much more tolerable with such a monitoring system than failures to detect serious problems (type I errors). Thus the rules that detect and interpret anoxic conditions allow for a wide level of tolerance.

On the other hand, false alarms may be much less tolerable with other kinds of systems. For example, the wildlife habitat expert system predicts the effects on the distribution and abundance of bird species from various forest management activities. This knowledge base provides advice on additional and sometimes costly activities that can help protect wildlife species from detrimental impacts the result from the harvesting of timber.

In this case, false alarms (type II errors) are much less tolerable than in the medical monitoring system, because high costs may be incurred. Thus the rules that predict negative impacts on wildlife entail narrow tolerance levels.

Validation Criteria

The following set of criteria may be used to test and evaluate the validity of a knowledge base:

Accuracy- how well a simulation reflects reality. Compare inferences made by rules with historic (known) data, observe correctness of the outcome.

Adaptability: possibilities for future development and application. Keep 1/0 and control rules general; revise facts and rules when new information is available. Periodically review the desirability of integrating with existing or proposed hardware or software systems. Should the system be self-modifying or context sensitive? Can it be customized for particular user needs?

Adequacy: the fraction of pertinent empirical observations that can be simulated. Establish list of parameters (variables, conditions, and relations) that influence inference outcome, determine which to include in rule set.

Appeal: usability; how well the knowledge base matches our intuition and stimulates thought; practicability. Appeal is a potentially key criterion for marketability; test usability by assessing 1/0 friendliness relatively early in the development process. Test simulation and practicability on site in beta-development stage.

Availability- existence of other, simpler, validated knowledge bases that solve the same problem(s), important for determining eventual marketability. Will users perceive the need for a new rule-based system if other tools are already available and meet their needs?

Breadth: proportional to the number of rules used in the knowledge base. Determine the number of contexts within which the system should be expected to perform, and thus the number of pertinent parameters to account for in the rule set.

Depth: proportional to the number and kinds of variables chosen to describe each component in the model. Determine the range of conditions the system will address and which parameters are necessary to diagnose, classify, and/or advise for each condition. Depth will in turn determine necessary input data and user interface.

Face validity: model credibility. Have knowledge base, inference structure, and output reviewed by credible human experts during early development of prototype and later expansion of full-scale system. Compile and report results.

Generality: capability of a knowledge base to be used with a broad range of similar problems. Define the general contexts within which the system can be expected to perform at expert levels and provide strong caution that use beyond these contexts may not yield accurate results.

Precision: capability of a model to replicate particular system parameters; also the number of significant figures used in numeric variables and computations. Ensure that all pertinent variations of parameters are represented in the rule base and facts. Express numbers as floating point or real format as necessary; use double precision for calculations, especially those involving matrix or linear algebra calculations.

Realism: accounting for relevant variables and relations.

Establish parameters and functions in the rule base in the same terms and with the same conceptual models used by experts or end user audience. Realism is particularly important it when developing the full-scale knowledge base, and also involves the logical order with which queries are made.

Reliability: the fraction of model predictions that are empirically correct (actually, part of a complex statistical analysis of the accuracy and correctness of the entire rule base). Reliability includes conditional and posterior probabilities of correct diagnoses and classifications. Ultimately describes statistical utility of the likelihoods in the rules and outputs.

Resolution: the number of parameters of a system the model attempts to mimic. Identify which parameters need to be defined and represented in detail and which can be grouped into more general conditions or ignored.

Robustness: conclusions that are not particularly sensitive to model structure. Determine which input parameters are least and most significant in the form of the interim (diagnosis, classification) and final (advice, alarm) results and output. Be sure the latter are well defined in the rules and functions.

Sensitivity: the degree to which variations of knowledge base parameters induce outputs that match historical data. Specifically determine sensitivity of results to each input parameter by varying that parameter incrementally, holding all other parameters constant and matching model output with historical (known) data.

Technical and operational validity: identification and importance of all divergence in model assumptions from perceived reality. Carefully explicate the contexts, conditions. and assumptions that underlie the rules and relations. Discuss how each assumption limits model results. How do they affect model accuracy, reliability, robustness, and generality?

Turing test: assessing the validity of a knowledge base by having human evaluators distinguish between the model's conclusions on a specific problem and a human expert's conclusions solving the same problem.

Usefulness: validates that the system contains neeessary and adequate parameters and relationships for use in problem-solving contexts (if at least some model predictions are empirically correct). Usefulness is trivial for a full-scale system but important for prototyping and adding onto existing rule sets.

Validity: a knowledge base's capability of producing empirically correct predictions. Given the contexts within which the system is expected to operate well, determine how man) actual conditions the system can accurately diagnose, classify, and advise. Determine the level of correctness with human experts in the same area and set realistic objectives for correctness of the knowledge base.

Wholeness: the number of processes and interactions reflected in the model. How complex is the rule base? How many factors does it use? Consider wholeness in light of adaptability.

ANALYZING RESULTS

Analysis and evaluation entail drawing conclusions from the results and assessing the implications of each conclusion for further development, marketing, or revision of the system. The analytical steps involved should be clearly determined before each test proceeds, especially where quantitative tests are concerned.

For example, during the testing of the second generation rule set, assessing the accuracy of the system should involve determining the specific problem domain, selecting a representative spectrum of known cases against which to test the system (these should be cases that were not used in the initial development of the knowledge base), operating the system and recording its output (diagnosis, classification, advice), and comparing the output with the known conditions of each case. A statistical comparison between system predictions and known cases may help determine the frequency and types of errors the system made.

Finally, the results of the validation tests should be used to evaluate the development of the system. Does the system meet originally defined objectives and standards? Can it be used in the intended problem domain within an acceptable error rate? Can it be successfully placed in the market? What are the needs for adding to or revising the knowledge base? Are there additional considerations to address, such as legal or regulatory constraints? What can be learned to help in the development of the next system? Every knowledge engineering project is unique in its intent and application, and every project should embrace carefully designed tests of the validity of a knowledge base.

REFERENCES

1. Rader, C.D., V.M. Crowe, and B.G. Marcot. "CAPS: A Pattern Recognition Expert System Prototype for Respiratory and Anesthesia Monitoring." Presented at Wessex-87 IEEE Expert Systems Conference, June 1987, Anaheim, Calif.

2. Marcot, B.G. "Use of Expert Systems in Wildlife-Habitat Modeling." In *Wildlife 2000: Modeling Habitat Relationships of Terrestrial Vertebrates,* edited by J. Verner, M.L. Morrison, and CJ. Ralph. Madison, Wisc.: Univ. of Wisconsin Press, 1986, pp. 145-150.

3. Weiss, S.M., and C.A. Kulikowski. A *Practical Guide to Designing Expert Sysleins.* Totowa, NJ.: Rowman & Allanheld, 1984.

4. Rich, E. *Artificial Intelligence.* New York, N.Y.: McGraw-Hill, 1983.

5. Politakis, P., and S.M. Weiss. "Using Empirical Analysis to Refine Expert System Knowledge Bases." *Artificial Intelligence* 22(1984):23-48.

6. Politakis, P. "Empirical Analysis for Expert Systems." In *Research Notes in Artificial Intelligence 6.* Boston, Mass.: Pitman Advanced Publishing Program, 1985.

7. Chandrasekaran, B. "On Evaluating AI Systems for Medical Diagnosis." *AI Magazine* 4(2):34-37.

Bruce G. Marcot, Ph.D., is an AI consultant with Thought Services, Portland, Ore.