U S L / D B M S N A S A / R E C O N

W O R K I N G P A P E R S E R I E S

Report Number

DBMS.NASA/RECON-19

The USL/DBMS NASA/RECON Working Paper Series contains a collection of reports representing results of activities being conducted by the Center for Advanced Computer Studies of the University of Southwestern Louisiana pursuant to the specifications of National Aeronautics and Space Administration Contract Number NASW-3846. The work on this contract is being performed jointly by the University of Southwestern Louisiana and Southern University.

For more information, contact:

Wayne D. Dominick

Editor
USL/DBMS NASA/RECON Working Paper Series
Center for Advanced Computer Studies
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504
(318) 231-6308

(NASA-CR-184527) AN HIERARCHICAL APPROACH TO PERFORMANCE EVALUATION OF EXPERT SYSTEMS Final Report, 1 Jul. 1985 - 31 Dec. 1987 (University of Southwestern Louisiana, Lafayette. Center for Advanced Computer

N89-14967

Unclas

G3/82 0183565

AN HIERARCHICAL APPROACH TO PERFORMANCE EVALUATION OF EXPERT SYSTEMS

Srinu Kavi

Center for Advanced Computer Studies
University of Southwestern Louisiana
P. O. Box 44330
Lafayette, Louisiana 70504

May 6, 1985

AN HIERARCHICAL APPROACH TO PERFORMANCE EVALUATION OF EXPERT SYSTEMS

ABSTRACT

The number and the size of expert systems is growing rapidly. Formal evaluation of these systems - which is not done for many systems - increases the acceptability by the user community and hence their success. Hierarchical evaluation that had been conducted for computer systems is applied for expert system performance evaluation. Expert systems are also evaluated by treating them as software systems (or programs). This paper reports many of the basic concepts and ideas in the Performance Evaluation of Expert Systems Study that is being conducted at USL. Future report(s) will provide details and/or explanations of many of these ideas and issues identified in this report.

TABLE OF CONTENTS

AN HIERARCHICAL APPROACH TO PERFORMANCE EVALUATION OF EXPERT SYSTEMS

1. EVALUATION OF EXPERT SYSTEMS

1.1 Introduction

Now it is more than 10 years since the MYCIN research group at Stanford demonstrated the potential of the Artificial Intelligence discipline to undertake problem-solving in complex, real-world domains. In contrast to much of the work being conducted at the time, namely, to develop powerful, general purpose problem-solving methods, the MYCIN group took the alternative approach of incorporating the domain knowledge actually used by experts. Thus, the MYCIN program successfully achieved its objectives of diagnosing bacterial infections by using explicit knowledge of bacteremia (bacteria in the blood).

The incorporation of explicit domain knowledge into problem-solving programs proved to be of great practical importance. First, it enabled AI to solve many real-world problems that were previously beyond the scope of "conventional" programming methods and came to known as "knowledge engineering" (or "applied" AI). Secondly, the knowledge-based approach created its own problems, thus extending the theoretical interests in AI. Some of the issues raised include knowledge representation, representation of uncertainty.

There are now a substantial number of "expert systems" which can claim expert, or near expert, performance in a wide range of domains, undertaking such problem-solving tasks as medical diagnosis, data analysis and planning. Some of the best known systems include:

(1) MYCIN, a system for diagnosing bacterial infections.

(2) DENDRAL, a system for inferring the structure of chemical compounds from mass-spectral data.

(3) CASNET/glaucoma, a system for diagnosing the eye disease glaucoma.

(4) R1, a system for configuring VAX computers

(5) DIPMETER ADVISOR, a system for oil well log interpretation.

(6) MOLGEN, an automated "scientist's assistant" in the field of molecular genetics.

1.2 Why Evaluate Expert Systems?

The systems listed above have been very successful within their narrow domains of application, and some systems are moving from research and development environment into the marketplace. DENDRAL, R1, and MOLGEN all are routinely used by users who are not connected to the designers of the system. Therefore, the developers are expected to provide some objective demonstration that the system performs as well as they claim.

Existing techniques for evaluating the ESs are few and primitive. Much more effort has been devoted to designing and constructing ESs than to measuring their resulting performance. There is no consensus about how to evaluate ESs (or when or why).

The criteria like correctness, efficiency, or friendliness that are used to evaluate other computer-based systems can be used to evaluate ESs. But they are not enough, because ESs use human expertise and are usually compared with human performance. But this raises an important issue: whether a correct solution (for an ES) is one that a human expert would give, one that a group of experts would agree upon, or that represents the ideal solution (after testing and analyzing) [Hayes-Roth, et al, 83].

No one has developed a method to evaluate human expertise objectively and adequately. Though there are many kinds of tests for human experts, few of these methods seem to apply directly to the issues faced in evaluating an ES. (The last few paragraphs are taken from [Kavi, 84], pages 255-256)

This report is a initial attempt to develop a methodology in evaluating expert systems. Many issues will need to be explained and discussed in more detail in future versions of this report.

1.3 What to Evaluate and by Whom?

ESs are evaluated by various individuals at various stages of their development as shown in Figure 1.

 | N A S A |

```
+----------+      +-----------+
| Domain   |      |  End      |
| Experts  |      |  Users    |
+----------+      +-----------+
     |                 |
     V                 V
+-------------------------+       +------------+
|    EXPERT  SYSTEM       |<--- -| AI         |
+-------------------------+       | Researcher |
                                  +------------+
     |                 |
+----------+      +------------+
| System   |      | Knowledge  |
| Analyst  |      | Engineer   |
+----------+      +------------+
```

FIGURE 1 Expert System and its Evaluators

1. Domain Experts

Since accurate, reliable advice is essential for any ES, it is usually the area of greatest research interest and is an area to emphasize in evaluation. However, it is not easy to decide whether a system's advice is appropriate or accurate, mainly because ESs are built for those domains in which decisions are highly judgmental. Yet, it would be difficult for the intended ES users to accept the ES if they are not convinced that the decisions made and the advice given are appropriate or accurate.

Some experts also evaluate an ES not just to determine whether or not the ES produces "correct" advice, but to determine whether it reaches decisions in a "correct" way. Though MYCIN was not intended to simulate human problem solving in any formal way, there is an increasing realization that expert-level performance may require attention to the mechanisms by which human experts actually solve the problems for which the ESs are being built [Buchanan, et. al, 84].

2. End Users

End users evaluate ESs for their discourse (I/O content) and hardware medium (I/O medium) [Hayes-Roth, et. al, 83]. Some of the issues related to discourse are:

(1) The choice of words used in the questions and responses generated by the program.

(2) The ability of the ES to explain the basis of its decisions and to customize those explanations appropriately for the level of expertise of the user.

(3) The ability of the system to assist the user when he or she is confused or wants help.

(4) The ability of the ES to give advice and to educate the user in such a way that the psychological barriers

_ to computer use are eliminated.

One of the most important issue when end-users evaluate the hardware environment of an ES is the interface that facilitate interactions between the user and the ES. Light pen interfaces, touch screens, specialized keyboards, etc. will greatly simplify the users' interaction with ESs. Details of the hardware interface often influence the design of ES software. Thus, when one is evaluating an ES for its decision-making performance or its discourse ability, one cannot ignore the user's reaction to the terminal interface.

3. Knowledge Engineers

Knowledge engineers evaluate an ES throughout its life cycle. Some of the issues that are important for knowledge engineers during system development are:

(1) Knowledge Representation Scheme (production rules, semantic nets, or frame representation).

(2) Knowledge chunk size.

(3) Control Structures (inference engine strategies).

(4)_ Certainity Factors (CFs).

There are two useful metrics to evaluating the usefulness or effectiveness of a particular control strategy. They are:

(1) Branching factor.

(2) Penetrance.

Various other metrics or criteria could be used to evaluate an ES. For example, the number (or percentage) of rules in the knowledge base that are used for a number (or percentage) of time or the number (or percentage) of rules that change frequently. This measure could be used to organize (or store) a knowledge base in a more efficient manner.

4. Systems Analysts

System analysts evaluate an ES for its efficiency and cost effectiveness:

(1) Efficiency. The issues considered here include response time, CPU usage, memory allocation, etc.

(2) Cost effectiveness. The issues that are important here are whether the ES is a "deep and narrow" type or a "broad and shallow" type.

1.4 When to Evaluate?

The evaluation of an ES is a continual process that should begin at the time of system design, continue through the early stages of development and become increasingly formal as a developing system moves towards real-world implementation. Table 1 summarizes the steps in the evolution of an ES and the steps are discussed briefly below [Gashing, et. al, 83].

1. Top-level design; definition of long-range goals.

2. Implementation of prototype, showing feasibility.

3. Refinement of system, usually by
   a. Running informal test cases to generate feedback from the expert, resulting in refined prototype.
   b. Releasing the prototype to friendly users and soliciting their feedback.
   c. Revising system on basis of user's feedback.
   d. Releasing revised prototype to users and returning to stage 3b.

4. Structured evaluation of performance.

5. Structured evaluation of acceptability to users.

6. Service functioning for extended period in prototype environment.

7. Conducting follow-up studies to demonstrate the system's large-scale usefulness.

8. Making program changes to allow wide distribution of the system.

9. General release and marketing with firm plans for maintenance and updating.

TABLE 1 Steps in the Implementation of an Expert System

[Hayes-Roth, et. al, 83 p. 258]

(1) The first stage of a system's development (Step 1), the initial design, should contain explicit statements of what the measures of the program's success will be and how success or failure of the system will be evaluated along with the long-range goals for building the ES. If evaluation plans and long-range goals are clearly defined, they will influence the early design of the ESs. For example, if explanation capabilities are deemed to be crucial for the user community for whom the system is intended, this will have important implications for the system's underlying knowledge representation.

(2) In Step 2, the feasibility of the design of the ES is demonstrated. At this stage, there is no attempt to demonstrate expert-level performance. The goal is to show that there is a representation scheme appropriate for the task domain, and that knowledge engineering techniques can be applied to build a prototype system which shows some reasonable performance of some subtask of that domain.

(3) In Step 3, some informal test cases are run through the developing system, the system's performance is observed, and feedback is sought from expert

collaborators as well as some potential users. This feedback will be useful in identifying major problem areas in the system's development. Revisions will be made to the system and it will be released again to expert collaborators and users. This iterative process may go on for months or years, depending on the complexity of the knowledge domain, the flexibility of the knowledge representation and the control strategies.

(4) After Step 3 is successfully completed, i.e., after the system is performing well on most cases with which it is presented, a more structured evaluation of the ES's decision making ability should be conducted (Step 4). In this phase, the emphasis is to test the ES's ability to solve random cases (within its domain) with expert-level performance. A formal evaluation with randomized case selection may reveal that the ES is not performing at an expert level, in which case one should go back to Step 3.

(5) In Step 4, actual utility of the ES is not emphasized, only its expert-level problem solving. In Step 5, the end-users will evaluate the system in a formal way. The purpose of this phase is to determine whether or

not the ES is acceptable to the users for whom it was intended. The emphasis at this phase is on the ES's discourse abilities and the hardware environment that is provided. A successful completion of this phase will ensure that the ES does make expert-level decisions and that it is acceptable to users.

(6) Step 6 is "field testing". During this phase, a large number of cases (along with the associated peculiarities of the environment) are tested by the ES. Careful attention during this stage must be directed towards problems of scale, i.e., what new difficulties will arise when the system is made available to a large number of users.

(7) In Step 7, follow-up studies to demonstrate the ES's large-scale usefulness are conducted. Some of the issues of concern at this phase are the systems's efficiency, its cost effectiveness, its acceptability to users (who are not involved in its early experimental development), and its impact on the execution of the task with which it was designed to assist.

(8) Before the system can be distributed (Step 8), some

modifications may be required to allow the system to run on a smaller or portable machine.

(9) Step 9 is general release of the ES as a marketable product or in-house tool. Firm plans for maintaining the knowledge base and keeping it current are necessary at this phase.

The methodology that is considered for evaluation of expert systems is applicable at Step 4, the formal evaluation of the ES. It will be described in detail in a future report.

| N A S A |

2. EXPERT SYSTEM AS A COMPUTER SYSTEM

In this chapter, an ES is considered as a computer system and some basic performance evaluation and system modelling concepts are presented. Actual models for expert system performance evaluation will be discussed in a future report.

2.1 Performance Evaluation Concepts

The performance of a computer system is defined as the effectiveness with which the system handles a specific application. Various measures can be used to describe the performance of a computer system. For example, the system throughput, the most commonly used, is defined as the number of tasks processed by the system in a unit of time.

Performance evaluation can be viewed from two different aspects:

(1) The determination of the performance function F, such that

System performance = F(av1, ..avm, wv1, ..wvn)

where the avi are the system architecture parameters, and the wvj are the system workload parameters.

(2) The estimation of values of the above performance function for a specific set of system parameter values

(av1, ..avm, wv1, ..wvn)

2.2 System Modeling Concepts

Any analysis of a system is only an analysis of a model of the system. This is true of system performance evaluation, which is an analysis of one aspect of the system - its performance. A model of a system can be defined as an abstraction that contains only the significant variables and relations of the system [Zeigler, 76].

2.3 Types of Models

Models used for system performance evaluation can be divided into three broad classes [Svobodova, 76]:

(1) _Structural Models_ describe aspects of individual system components and their interaction. They usually serve as the basis for more abstract models, by providing an interface between the real system and the more abstract models. An example is a block diagram model of a system in which each block is a system component.

(2) _Functional Models_ define the operation of the system such that the model can be analyzed mathematically or studied empirically. Examples of functional models are queueing models that have mathematical solutions for the performance measures of interest, and simulation models that provide empirical evaluations of performance measures.

(3) _Analytical Performance Models_ formulate the dependence of performance on the system workload and architectural variables. Such models are usually functions that are fitted to data obtained from functional models.

2.4 Characteristics of Models

Three main aspects of models are: validity, cost, and amount of information obtainable from the model.

1. Validity

A model is said to be valid when the performance measure values generated by it agree with the actual observations of system performance to within a desired range of accuracy. The range of validity of a model is the region in the multi-dimensional space of system parameters over which the model is valid.

There are varying degrees of rigor to the validity of models [Zeigler, 76]:

(1) At the least rigorous level, a model is replicatively valid if it matches the performance values already acquired from the real system.

(2) At a more rigorous level, a model is predictively valid when its predictions of performance are corroborated by observations of the system.

(3) At the most rigorous level, a model is <u>structurally valid</u> if it not only reproduces the observed system behavior, but truly reflects the way in which the real system operates to produce this behavior.

2. <u>Cost</u>

The cost of a model is usually related to the computational complexity of the model, i.e., the work involved in using the model to make a single evaluation of system performance. Thus simulation models are usually quite expensive in their computational demands, while mathematical models such as queueing models and analytical performance models are quite inexpensive.

3. <u>Amount of Information Obtainable from a Model</u>

During a performance evaluation study, the systems analyst is interested in more than one measure of system performance. For example, in a system which is an interconnection of resources, resource utilization is as important a measure as system throughput, since it can point to system bottlenecks. Models with higher structural validity are capable of yielding more information than models of merely predictive validity.

The detail with which the information can be obtained also depends on the models. A queueing model may attempt to yield accurately only the average utilization of a resource, while a simulation model can yield an entire histogram of resource utilization.

2.5 Model Building Process

Regardless of the type of model chosen, there are certain common features in the process of building the model as a tool for performance evaluation. The following are some of the basic phases of the model building process [Zeigler, 76]:

(1) Choice of Experimental Frame. The experimental frame characterizes a limited region of the entire system parameter space, in which the system is to be modelled. All the characteristics discussed in the previous section of a model are only with respect to the experimental frame for which the model is constructed. Thus a model may be invalid in an experimental frame other than the one chosen, but only its validity in the chosen frame is of importance.

(2) Model Calibration. Calibration is the process of estimating the parameters that describe the model in the experimental frame. For example, the parameters of an analytical model that expresses performance as a linear function of system parameters,

$$P = Bo + \sum_{i=1}^{m} Bi \cdot avi + \sum_{j=1}^{n} rj \cdot wvj$$

are the coefficients Bi (i = 0 through m) and rj (j = 1 through m). The calibration of such a model may involve the fitting of a linear regression equation to observed values of system performance for varying values of the system parameters.

(3) Model Validation. Once a model has been calibrated, it can be used to predict system performance. Validation is the process of establishing the validity of the model by comparing model predictions of performance with observations of system performance. If the validity is satisfactory, the model predictions in the experimental frame will be accepted. If the validity is poor, the model may have to be recalibrated with the new observations of performance.

Calibration and validation for any model can be improved simultaneously up to a point; beyond that point, they may have to be traded off. Thus, calibration using data from a larger number of observations than the order of complexity of the model may cause poor overall validity in the region. On the other hand, the same model may yield an acceptable degree of validity for more local sub-regions.

(4) Prediction Using the Model. Once a model has been calibrated and validated in an experimental frame, it can be used to predict system performance in that frame. However, if the experimental frame should ever change, the process of calibration and validation will have to be repeated for the new frame.

2.6 Hierarchy of Performance Evaluation Models

Performance evaluation of computer systems is not a new concept. For example, Ballance, et. al. describe a simulation model that was used in the design of the look-ahead unit of the IBM Stretch System [Ballance, et. al., 62]. Boland, et. al. discuss a simulation model used in designing the memory unit of the IBM System 360/Model 91 [Boland, et. al., 67].

Hierarchical approaches to modelling have also been examined in the past [Browne, et. al., 72; Bhandarkar, 76]. These models are concerned with the reduction in complexity of analytical models, by structural decomposition of the system model to form sub-system models that are analyzed independently. Thus, all the models in the hierarchy use the same modelling tools. However, Kumar's model [Kumar, 78] brings together a variety of state-of-the-art modelling tools, whose range of cost and complexity make it very suitable for use in a hierarchy.

The reasons for undertaking performance studies are (typically) the following:

(1) To design a computer system which is the optimum system for some objective function that includes system performance as a component.

(2) To optimize an existing computer system for an objective function as in (1).

In either case, the systems analyst is interested in obtaining an optimum system configuration.

Since optimization procedures usually use an iterative scheme to converge to the optimum, a number of evaluations of the objective function is required. This requires that the performance component of the function be evaluated with minimum cost, so as to keep the cost of the optimization procedure within reasonable bounds. On the other hand, the performance evaluation must be sufficiently accurate to meet the accuracy demanded of the optimization procedure. A performance model hierarchy provides a cost-effective trade-off between accuracy and computational cost, in much the same way that a memory hierarchy is a cost-effective tradeoff between memory access time and cost [Kumar, 78].

2.7 Characteristics of the Hierarchy

The hierarchy of performance models have the following characteristics:

(1) Low end. The low end of the hierarchy contains models of high structural, and thus high predictive, validity. These models have a broad range of validity in the system parameter space and they are capable of yielding detailed performance information of great accuracy. The price to be paid for these qualities is in the high computational demands of these models.

(2) *High end*. The high end of the hierarchy will contain models of only predictive validity and their range of validity in the system parameter space is very much limited. The performance information that they yield generally has less accuracy than the low level models and are summary type, i.e., less detailed. However, they have the advantage of being very much less demanding in their computational requirements.

Intermediate levels of the hierarchy have intermediate values of these characteristics. Thus, travelling up the hierarchy, one sees models that have [Kumar, 78]:

(1) Less structural validity.

(2) More limited range of validity.

(3) Less detailed performance information.

(4) Less accurate information.

(5) Lower computational requirements.

In terms of the types of models described in Section 2.3, there will be structural models at low levels, functional models at intermediate levels and analytical models at very high levels of the hierarchy.

This hierarchical approach will be used to evaluate the performance of expert systems: developing a simulation model of (or of a part of) an ES and collecting performance data; this data will be used to develop an analytical model of the entire ES. The analytical model thus developed could be used to predict the performance of the ES.

Details of this approach along with other issues will be addressed in a future report.

3. THE EXPERT SYSTEM AS A SOFTWARE SYSTEM (OR PROGRAM)

Expert systems can also be evaluated by using many metrics (or criteria) that have been used to evaluate software systems in general. Some of the criteria are listed below:

(1) Reliability.

(2) Correctness.

(3) Learnability.

(4) Usability.

(5) Flexibility.

(7) Applicability.

(8) Security/Protection.

(9) Cost-effectiveness.

(10) Maintainability.

This list will be refined and/or extended when used to evaluate ESs and will be addressed in much more detail in a future report.

4. PRESENT CONSIDERATIONS

4.1 Expert System Considered

DIPMETER ADVISOR is tentatively "selected" as the ES that will be used in this hierarchical performance evaluation study.

DIPMETER ADVISOR (DA) is an expert system to interpret oil-well logs and was developed by Schlumberger (Ridgefield, Connecticut). An overview of the system is provided below (based on [Davis, et. al., 81; Gershman, 82]).

The principal business of Schlumberger is gathering and interpreting data related to oil wells. Sensors are lowered into the bore hole, and measurements are made as sensors are raised. Some measurements are taken every tenth of an inch and some are taken only every six inches. There are as many as twenty different types of sensors, and two or three can be used at once.

Data from the sensors dropped into bore holes are plotted on logs. These data indicate how different kinds of energy (sonic, electrical, and nuclear) interact with the formation.

Some of the questions that the expert who looks at the sensor data need to answer are whether there is hydrocarbon under the ground; if so is it oil or gas? how much is there? can it be removed? Depth is also important because in completing a well, making it a producer, the expert needs to know the exact location of the hydrocarbon.

Like experts in any domain, oil-well log interpretation experts are very few. Schlumberger's solution is to embody the skill of its valuable people in computer-based expert systems.

The DA attempts to emulate a special type of expert in this interpretation, starting with measurements from the dipmeter tool. The dipmeter measures the tilt of the underground formations. In one place, the layers may be basically flat. In another, the layers may start to incline at a substantial degree.

A sensor on each of the dipmeter tool's four arms measures the conductivity of the formation as the tool is pulled out of the hole. This results in four curves that look approximately the same. As the sensor comes out of a hole bored into a tilted formation, not all of the arms will measure the movement into the formation at the same time. By observing the difference in the measurements, one can determine the magnitude of the inclination, as well as the direction.

It is possible to associate certain types of patterns with these data. One of the patterns may indicate roughly constant magnitude with increasing depth. Another pattern may indicate magnitude with increasing depth. These patterns can be detected by part of the DA system. From relationships among these patterns, combined with other data, one can deduce the geology.

An example of a rule in DA is:

```
If    There exists a normal fault, and
      There exists a red pattern
         with bottom above the top of the fault,
         with length greater than 200 ft.,
         with azimuth perpendicular to the strike
              of the fault

Then   The fault is a growth fault
         with direction to downthrown block
         opposite to the azimuth of the red pattern
```

This type of rule deals with a large number of ideas: a normal fault, a red pattern, and some of the geometry. The goal of this kind of rule is to identify, to the greatest degree possible, what type of fault is involved. Currently there are 90 such rules in the DA system.

The DA goes through many steps before reaching conclusions, as shown in the Table 2.

Validity Check

- Washout zones
- Blank zones
- Mirror image zones

Structural Dip Analysis

- Structural dip zone determination
- Structural dip removal

Structural Feature Analysis

- Structural interruption detection
- Structural pattern detection
- Structural feature description

Stratigraphic Feature Analysis

- Lithology determination
- Depositional environment analysis
- Stratigraphic pattern detection
- Stratigraphic feature description

TABLE 2 DIPMETER ADVISOR System: Interpretation Steps

[Baker, 84 p. 59]

(1) First, the system verifies that the data is correct. The validity check is needed because several things can go wrong. For example, if the bore hole collapses, the sensors will be unable to measure anything.

(2) After the DA verifies that the data are correct, it begins the structured dip analysis. Structured dip refers to large tilts in the formation that have occurred after deposition. These tilts are important for two reasons: they are likely indicators that there is a fault in the area, and the tilts must be removed (i.e., the structure must be retilted by the system in order for the analysis to continue).

(3) In the third step, the DA tries to identify the geometry and the characteristics of the faults that are present.

(4) The last step is a stratigraphy analysis, which relates to determination of the geological structures involved.

4.2 Expert System Tool(s) Considered

From the preliminary comparison of LISP, PROLOG, and OPS5, OPS5 seems to be more suitable for expert system implementation. PROLOG's search strategies are very restrictive and it is difficult to encode uncertain information (PROLOG is based on predicate logic).

OPS5 is a production system oriented (IF-THEN type) language developed at CMU and is based on LISP. One of the highly successful expert system R1 (or XCON at DEC) is implemented by using OPS5.

Thorough evaluation of OPS5 is planned at USL (in the near future) and the results of the evaluation will be provided in a future report.

4.3 Reconstruction Versus Simulation

"Reconstruction" means building a small portion of an existing ES (e.g., DIPMETER ADVISOR) using some ES building tool. The main motivation is the following:

For those who like to develop a working ES without going through a lengthy apprenticeship of research and theoretical study, the approach of "rebuild and improve" will be very effective means of gaining considerable practical knowledge and experience in a relatively short time, i.e., to become a knowledge engineer quickly! (Japanese are well-known for this, they are doing the same in their Fifth Generation Project.)

By reconstructing DIPMETER ADVISOR (or some other ES), the author would like to analyze and provide a critical review of that system, addressing choice of the language, the methodology, the certainty factors, line of reasoning, etc. and be able to "fine tune" the system.

"Simulation" means simulating (using SIMSCRIPT or GPSS that are available at USL) the performance of an ES much like developing simulation models for computer systems in general.

REFERENCES

[Baker, 84]. J.D. Baker, "DIPMETER ADVISOR: An Expert Log Analysis System at Schlumberger," in The AI Business, P.H. Winston and K.A. Prendergast (Eds.), MIT Press, Cambridge, Mass, 1984.

[Ballance, et. al, 62]. R.S. Ballance, J.A. Cocke, and H.G. Kolsky, "The Lookahead Unit," in Planning a Computer System, McGraw-Hill, New York, NY, 1962.

[Bhandarkar, 76]. D.P. Bhandarkar, "A Hierarchy of Analytic Models for Complex Computer Systems," The European Computing Conference on Computer System Evaluation, September 1976.

[Boland, et. al, 67]. L.T. Boland, G.D. Granito, A.V. Marcotte, B.V. Messina, and J.W. Smith, "The IBM System 360/Model 91: Storage System," IBM J. of Res. and Dev., vol. 11, January 1967.

[Browne, et. al, 72]. J.C. Browne, K.M. Chandy, R.M. Brown, T.W. Keller, D.F. Towsley, and C.W. Dissly, "Hierarchical Techniques for the Development of Realistic Models of Complex Computer Systems," Proc. of the IEEE, vol. 63, June 1975.

[Davis, et. al, 81]. R. Davis, H. Austin, I. Carlbom, B. Frawley, P. Pruchnik, R. Sneiderman, and A. Gilreath, "The DIPMETER ADVISOR: Interpretation of Geological Signals," Seventh Int'l. Joint Conf. on AI (IJCAI-7), Vancouver, British Columbia, Canada, 1981.

[Gashing, et. al, 83]. J. Gashing, P. Klahr, H. Pople, E. Shortliffe, and A. Terry, "Evaluation of Expert Systems: Issues and Case Studies," in Building Expert Systems, F. Hayes-Roth, D.A. Waterman, and D.B. Lenat (Eds.), Addison-Wesley, Reading, Mass, 1983.

[Gershman, 82]. A. Gershman, "Building a Geological Expert System for Dipmeter Interpretation," Proc. of the European Conf. on AI, July, 1982.

[Hayes-Roth, et. al, 83]. F. Hayes-Roth, D.A. Waterman, and D.B. Waterman, Building Expert Systems, Addison-Wesley, Reading, MA, 1983.

[Kavi, 84]. S. Kavi, "Knowledge-Based Systems: A Critical Survey of Major Concepts, Issues, and Techniques," Master's Thesis, Department of Computer Science, University of Southwestern Louisiana, Lafayette, LA, December 1984.

[Kumar, 78]. B. Kumar, "Performance Evaluation of a Highly Concurrent Computer by Deterministic Simulation," Coordinated Science Laboratory Report R-717, University of Illinois, February 1978.

[Svobodova, 76]. L. Svobodova, Computer Performance Measurement and Evaluation Methods: Analysis and Applications, Elsevier, 1976.

[Zeigler, 76]. B.P. Zeigler, Theory of Modelling and Simulation, Wiley-Interscience, 1976.

4.20

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
IN-82	183565 ~~117375~~	

4. Title and Subtitle	5. Report Date
USL/NGT-19-010-900: AN HIERARCHICAL APPROACH TO PERFORMANCE EVALUATION OF EXPERT SYSTEMS 42 P.	May 6, 1985
	6. Performing Organization Code
7. Author(s) SRINU KAVI	8. Performing Organization Report No.
	10. Work Unit No.
9. Performing Organization Name and Address University of Southwestern Louisiana The Center for Advanced Computer Studies P.O. Box 44330 Lafayette, LA 70504-4330	11. Contract or Grant No. NGT-19-010-900
	13. Type of Report and Period Covered FINAL; 07/01/85 - 12/31/87
12. Sponsoring Agency Name and Address	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The number and size of expert systems is growing rapidly. Formal evaluation of these systems - which is not performed for many systems - increases the acceptability by the user community and hence their success. Hierarchical evaluation that had been conducted for computer systems is applied for expert system performance evaluation. Expert systems are also evaluated by treating them as software systems (or programs). This paper reports many of the basic concepts and ideas in the Performance Evaluation of Expert Systems Study that is being conducted at the University of Southwestern Louisiana.

This report represents one of the 72 attachment reports to the University of Southwestern Louisiana's Final Report on NASA Grant NGT-19-010-900. Accordingly, appropriate care should be taken in using this report out of the context of the full Final Report.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Performance Evaluation, Expert Systems, Information Storage and Retrieval Systems	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	40	

*For sale by the National Technical Information Service, Springfield, Virginia 22161