VALIDATION OF EXPERT SYSTEMS—WITH APPLICATIONS TO AUDITING AND ACCOUNTING EXPERT SYSTEMS*

Daniel E. O'Leary
Graduate School of Business, University of Southern California, Los Angeles, CA 90089

ABSTRACT

This paper proposes a set of definitions for the concepts "validation" and "assessment" applied to expert systems (ESs). It develops a framework for this validation and demonstrates the framework on existing accounting and auditing ESs to elicit some of the research issues involved in ES validation.

Validation is critical to the design and implementation of decision-making ESs. In a setting where objectivity is sought and variance is avoided, validation ascertains what a system knows, knows incorrectly, or does not know. Validation ascertains the system's level of expertise and investigates the theoretical basis on which the system is based. It evaluates the reliability of decisions made by the system.

The validation framework developed in this paper is research methods. It is designed to reflect the unique aspects of ESs (in contrast to other types of computer programs) and can be used by ES developers as a basis from which to perform validation and by researchers as a framework to elicit research issues in validation.

Subject Areas: Accounting Theory and Auditing.

INTRODUCTION

In addition to the processes of designing, developing, and implementing expert systems (ESs), validation is important to the decision-making success of a system and to the continued use of an ES. An ES that has not been validated sufficiently may make poor decisions. This can lead to a loss of confidence in the particular ES or in other systems, resulting in discontinued use and financial loss.

A number of different approaches to validating *particular* auditing and accounting ESs [16] [17] [18] [38] [39] and medical ESs [52] [8] [7] have been reported. Validation of general ESs [21] [33] and potential bases for the validation of ESs [4] also have been discussed. This paper presents a theory-based framework that is useful not only for guiding the validation of an ES, but also for eliciting other validation research issues.

Validation of ESs

Developing, designing, and implementing systems for making expert decisions requires analyzing the knowledge base and the decision-making capabilities of the system. That process, referred to as validation, requires

1. ascertaining what the system knows, does not know, or knows incorrectly

*The author would like to thank Doug Andrews, Nils Kandelin, Chen-en Ko, and Paul Watkins from the University of Southern California 1986 Conference on Expert Systems; participants at the American Accounting Association 1986 national meeting; Paul Watkins and the anonymous referees

2. ascertaining the level of expertise of the system
3. determining if the system is based on a theory for decision making in the particular domain
4. determining the reliability of the system

Once these concerns have been satisfactorily addressed, the system is updated to reflect the findings and may be revalidated. The process should be performed in an environment designed to provide an objective and cost-effective validation.

Validation *ascertains what the system knows, does not know, or knows incorrectly.* For example, when the expert system R1 (a system for configuring computers used by Digital Equipment [36]) was validated, errors and omissions in the knowledge base were identified. These errors and omissions were corrected before the system was placed in service [21].

Validation *ascertains the level of decision-making expertise of the system.* The types of problems a system can solve and the quality of its solutions define the level of expertise. For example, in education it is common to define an individual's level of accomplishment based on the types of problems that the individual can solve and the quality of the solutions produced.

Validation *determines if the ES is theory based.* Davis [12] [13] argued that basing an expert system on a theory is an efficient approach to developing such a system. Lack of a theory base has resulted in the failure of at least one system [48].

Validation *analyzes the reliability of the ES.* Given similar inputs, the ES should produce similar outputs. In addition, before and after revalidation a system generally should give similar responses to similar sets of test problems.

Validation Process

Previous analyses of ES validation have stressed the importance of periodic informal validation rather than a single, formal validation at the end of a project [44]. This validation will not be the same in each situation but will differ in formality and in the extent to which the validation process is implemented. As suggested by software engineering [44] and ES validation [21], formal acceptance testing may be appropriate. (A formal acceptance test might consist of a test where the user formally signs off on the quality of the decisions made by the system.)

ES Assessment

Although the focus of this paper is on ES validation, other issues addressed in developing, designing, and implementing ESs do not relate to decisions that the system makes; nevertheless, these other issues do contribute to the overall success of a system. The assessment process involves analyzing the user interface and supporting the quality of the development effort. For example, it involves

1. ascertaining the quality of the user interface [31]

of an earlier version of this paper; and, in particular, John Henderson for their comments. Earlier versions of this paper were presented at the University of Southern California Symposium on Expert Systems, February 1986, and the American Accounting Association National Meeting, August 1986.

2. evaluating the documentation of the system (a typical weakness of most systems, with good reason; since ESs are evolutionary, the documentation does not evolve at the same rate as the rest of the system)
3. determining what language or shell should be used to develop the system
4. analyzing the quality of the system programming

Paper Objectives

This paper has four primary objectives: to propose a set of definitions for the concepts "validation" and "assessment" applied to ESs; to develop a framework for the validation of ESs; to demonstrate that framework on some previously published auditing and accounting ESs; and to use that demonstration to elicit some of the research issues involved in ES validation.

The paper proceeds as follows. The objective of developing a framework is to take advantage of the unique aspects of ESs. Therefore the next section discusses some unique aspects of ESs in general and of auditing and accounting ESs in particular. Using a research methods approach, these unique characteristics are incorporated into a framework for ES validation. The framework is demonstrated on accounting and auditing ESs and that demonstration is used to elicit some of the research issues involved in the validation of ESs.

UNIQUE CHARACTERISTICS OF AUDITING AND ACCOUNTING ESS

If ESs are like other computer systems, then validating an ES should be the same as validating any other computer system. However if ESs are different, then their unique characteristics can be used to develop a specific framework for their validation. A number of technical, environmental, design, and domain characteristics distinguish ESs from other computer-based systems.

The *technical* aspects which distinguish ESs include the following. First, ESs process symbolic information (e.g., "If. . .then" rules) rather than just numeric information [1] [43]. This ability to process symbolic information allows ESs to solve nonnumeric-like problems, which generally are less precise than numeric problems. Second, experience with representing knowledge shows that a fraction (less than 10 percent) of this knowledge escapes standard representation schemes and requires special "fixes" [19] to make it accessible. Since other systems do not use knowledge representation, they do not face this problem. Third, ESs often are developed using either artificial intelligence (AI) languages or ES shells [25]. An AI language is a computer language aimed at processing symbolic information (e.g., Prolog [11] or Lisp [51]). An ES shell is software designed to aid the development of an ES by prespecifying the inference engine and making it easier to input knowledge into the system. Some of the first ES shells were EMYCIN [7] and AL/X [15]. More recently developed shells include Texas Instrument's Personal Consultant Plus, Teknowledge's M.1, and Inference Corporation's ART. The characteristics of ES shells and AI languages (e.g., their ease of examination by nonprogrammers) can be made a part of an ES validation framework.

Environmental characteristics which distinguish ESs include the following. First, ESs directly influence or make decisions [1] [25] [49]. Other systems (e.g., decision support systems (DSSs)) simply support decision making or have an indirect impact on decisions. Second, the expertise being modeled by an ES generally is in short supply, is an expensive resource, or is not readily available at a particular geographic location [20]. This is in contrast to other computer systems (e.g., accounting systems) where generally a number of personnel understand what the system is designed to accomplish.

In addition to these characteristics that differentiate ESs from virtually all other types of computer systems, the dominant *design* methodology for both ESs and DSSs differentiates them from traditional computer systems. First, ESs (and DSSs) often are developed using a "middle-out" design rather than the traditional data processing approach of top-down or bottom-up design. A middle-out design philosophy starts with a prototype and gradually expands the system to meet the needs of the decision [27]. Second, like DSSs, ESs evolve over time—the system changes as the decision-making process gradually is understood and modeled [28]. As a consequence, traditional validation models based on other design philosophies are not likely to meet the needs of an ES.

Finally, some *domain* characteristics of auditing and accounting decisions (and other business-based decisions) often distinguish these decisions from decisions made in other domains. First, in contrast to some scientific decisions that have a unique solution (such as those represented by the ES DENDRAL [1]), auditing and accounting decisions generally do not have a single solution. Second, the decision reached often can be evaluated only by how similar it is to decisions other decision makers develop (i.e., by consensus); there may be no way to rank the decisions a priori. Third, different schools of thought may represent alternative knowledge bases. This means experts may not agree on a recommended solution. (This can apply to other disciplines as well.) Fourth, a "good" decision does not necessarily result in "good" consequences. Decisions are based on information available at the time they are made; this does not guarantee desirable consequences at a later time. Fifth, in contrast to some disciplines where a decision has no direct dollar value, the decision modeled by an ES can have substantial monetary value.

VALIDATION FRAMEWORK

Software engineering encompasses the general set of tools and techniques that aid programmers in software development. Since ESs are computer programs, software engineering might appear to be a likely candidate to supply a framework for ES validation. However, the unique characteristics of ESs in general, and of accounting and auditing ESs in particular, indicate that such a framework will not be appropriate.

An examination of one such approach [44] supports this view. First, in software engineering the focus of validation is on finding errors in the program. The validation of ESs is more than just a process of finding errors. Here the validation process meets the development needs of defining the level of expertise of the system, identifying what the system can and cannot do, understanding the quality of the decisions produced by the system, and describing the theory on which the system is based. Because software engineering generally is not concerned directly with human expertise, it is not directly concerned with these issues.

Second, the unique characteristics of ESs indicate they are different from other computer programs. Since ESs process symbolic information, use ES shells and AI languages, and require special fixes in their knowledge bases, they also require different validation approaches than other computer programs. Further, ESs directly affect decisions and they model expertise that is in short supply; the environment in which they operate is different from that of other computer programs.

Third, the domain-based decisions of auditing and accounting ESs often are not well enough understood to use traditional software engineering approaches. While software engineering generally uses structured top-down or bottom-up approach in the design and evaluation of software, ESs are evolutionary and often are developed using a middle-out approach.

One alternative to the software engineering approach is a research methods approach. This approach views the development of ESs as experimental representations of human expertise, that is, as research designs. Kerlinger defined research design as "the plan, structure and strategy of investigation conceived so as to obtain answers to research questions and to control variance" [30, p. 300]. In a similar sense, validation can be defined as the plan, structure, and strategy of investigation conceived so as to obtain answers to questions about the knowledge and decision processes used in ESs and to control variance in that process.

Kerlinger also noted, "research designs are invented to enable the researcher to answer research questions as validly, accurately, objectively, and economically as possible" [30, p. 301]. He also noted that accuracy consists of four concepts: reliability, systematic variance, extraneous systematic variance, and error variance.

We use these characteristics of research design (validity, objectivity, economics, and accuracy) to formulate a framework (Table 1) for the validation of ESs. Since Kerlinger noted the existence of three types of validity in research methods (content validity, criterion-related validity, and construct validity), each will be treated separately.

Content Validity

"Content validity is the representativeness or sampling adequacy of the content—the substance, the matter, the topics" [30, p. 458]. In validating ESs, content validity refers to ascertaining what the system knows, does not know, or knows incorrectly. This can be operationalized in at least two ways: by direct examination of the system components or by testing the system.

Direct Examination of the System Components. An ES is based on the knowledge of experts. Accordingly, it is important that these experts know what is contained in the system. The expert can examine the knowledge base directly in one of two ways. First, the ES could develop, for example, a list of the rules in its knowledge base for periodic review or a summary of the process it uses for

Table 1: Summary of validation framework.

1. Content validity
   a. Direct examination of the system by the expert
   b. System test against human experts (Turing test)
      (1) Intraexpert test
      (2) Interexpert test
   c. System test against other models
2. Criterion validity
   a. Definition of the level of expertise of the system
      (1) Human evaluation criteria
      (2) Test problems to define the level of expertise
      (3) Quality of responses defined
   b. Knowledge-base criteria
   c. Clarification of evaluation criteria
3. Construct validity
4. Objectivity
   a. Programmer validation
   b. Independent administration of validation
   c. Sponsor/end-user validation
   d. Biasing and blinding
   e. Different development and test data
5. Economics (cost-benefit)
6. Reliability
   a. System test against itself (sensitivity analysis)
   b. Test problems for revalidation
7. Systematic variance (experimental variance)
   a. Problems reflecting range of problems encountered
   b. Variation in the test problems
   c. Number of test problems
   d. Type I and Type II errors
8. Extraneous variance
   a. Complexity of the system
   b. ES's location in the system life cycle
   c. Recognition, examination, and testing of special fixes
   d. Location of judges during testing
   e. Learning on part of judges
9. Error variance

the inference engine. Second, the expert could examine the storage of the information by the ES directly.

The first solution might produce reports that could be read easily but, being an intermediate step, it also would require validation. As a result, it would be preferable if the expert could examine the knowledge base directly. In the second case, the primary concern is the format of the knowledge to be reviewed. If the system were built using an ES shell or an AI language such as Prolog, direct review likely is feasible.

A knowledge expert might not be able to investigate the inference engine to see whether it is correct because of the complexity of the computer code. However, if an ES shell is used, the inference engine normally would be prespecified.

The direct examination of the components has some limitations. First, fear of computers [21] may generate hostility toward the system. Second, human information processing is limited. Direct examination may not correctly process the links between parts of the knowledge base; also, the breadth of information contained in the knowledge base could limit the success of a direct investigation. Third, the direct approach may require substantial resources. If the expertise is scarce, purchasing expertise may be costly. Fourth, direct examination may be a tedious job that could lead to errors in the validation. Fifth, if the knowledge base is large or complex, examination could prove very time consuming and complex and lead to information overload. Sixth, current technology is not designed to facilitate direct examination.

Further, any direct analysis of the knowledge base is limited to looking at the pieces of the base and not at how they interact. In addition, translation of the computer code makes any direct analysis of the heuristics in the inference engine difficult. Accordingly, the best solution is to test the system as a whole.

System Test against Human Expert. If the system is designed to perform as an expert, it should be tested against an expert(s). To ensure that the ES has captured the expertise of the expert it was designed around, its decisions should be compared to that expert's.

However, such an intraexpert test procedure has the potential to introduce bias into the validation process through the acquisition of information from memory or the manner in which information is processed [26]. In terms of ESs, this means that the knowledge base or the relationships between sets of rules or the heuristics in the inference engine all may contain bias. This suggests that the ES also must be tested against *other* experts from the same "school" as the original expert but who are not biased or invested in the particular ES.

Since such interexpert tests when conducted using experts from *alternative* schools may yield contradictory decisions, this type of comparison is not recommended. However, alternative views of the world might produce additional knowledge for the system.

System Test against Other Models. System tests against human experts may be preferred to tests against other models. However, tests against human experts can be expensive. Experts also face time constraints [7]. Since the system is a model, one important characteristic should be its relationship to other models. An ES should

1. perform in a similar or better fashion than other models for the same problem
2. be able to solve problems that are not amenable to other solution methodologies

One type of model used to analyze decisions is regression analysis [6] [32] where independent variables represent the variables used by a decision maker. Other types of models may be used, but the model used largely is a function of the problem and previous recommended solutions to the problem. In particular, the preferred model would be the one that has provided the best solution to the problem so far. For example, an ES for production scheduling would be tested against an existing operations research model.

Unfortunately, in some cases comparison of the ES to regression or to other approaches may not be feasible because of the structure of the ES knowledge base, (e.g., if "If...then" statements are present). It may be very difficult to translate such rules into numeric variables. However, simulation generally is a feasible alternative [9] [10].

Criterion Validity

"Criterion-related validity is studied by comparing test or scale scores with one or more external variables, or criteria, known or believed to measure the attribute under study" [30, p. 459]. In ES validation, this refers to the criteria used to validate the system, for example, to ascertain the level of the system's expertise.

The primary criterion for system validation is the relationship between the decisions developed by the system and decisions developed by human experts. In AI this is referred to broadly as a Turing test. However, this sort of relationship is not evaluated easily. Difficulties arise for a number of reasons: differing definitions of the level of expertise, differing knowledge-base criteria, and lack of clarity about what is to be evaluated.

Definition of the Level of Expertise. One way to define and measure the level of expertise the ES has attained is to use the same criteria identified for defining expertise in human experts [7]. Where feasible, this is a good option. However, in many auditing- and accounting-based decision-making situations specific criteria are not available for specific decisions. Instead, a portfolio of decisions is subject to a set of criteria. For example, managers may be evaluated on their monthly profit and loss statements. These statements reflect a portfolio of decisions.

An alternative approach is to evaluate the system's performance on a set of test problems. There are at least two possible ways to do this. First, in analyzing a human's knowledge, the problems that the person can solve determine his/her level of accomplishment. Similarly, the difficulty of test problems that a system can solve defines the system's level of expertise. Second, in analyzing a human's knowledge, the quality of the solutions also helps determine the level of accomplishment. Similarly, the quality of the responses to test problems defines the level of expertise of an ES. Quality, of course, is a difficult concept to measure; its definition often is situation-based. However, with humans the number of correct responses obtained often is used to measure quality. This criterion also has been used to validate ESs.

Knowledge-Base Criteria. There are three generic knowledge-base criteria that suggest what to test: consistency, accuracy, and completeness. Consistency refers to the relationship between the information in the knowledge base and the ability of the inference engine to process the knowledge base in a consistent manner. Accuracy refers to the correctness of the knowledge in the knowledge base. Completeness refers to the amount of knowledge built into the knowledge base.

Clarification of Evaluation Criteria. Although general approaches may be developed to test the above generic criteria, a particular application also may require specific evaluation criteria [21]. For example, when evaluating an automobile a number of criterion (such as luxury, economy, and sportiness) might be developed. Particular measures then must be developed to characterize these concepts. For example, "miles per gallon" can be used to measure economy. Because many measures are possible, researchers suggest that the specific criteria and characterizations to be used be established and agreed on before validation begins.

Construct Validity

Construct validity refers to those constructs or factors that the test is designed to discover. "The significant point about construct validity, that sets it apart from other types of validity is its preoccupation with theory [and] theoretical constructs" [30, p. 461]. In terms of ESs, this type of validity indicates the importance of the existence of a theory on which the system is based. Construct validity suggests that a purely empirical approach may be inappropriate. Davis [12] [13] suggested that an empirical approach is not as efficient as an approach based on a theory (or at least on an understanding of the problem at hand) when developing an ES. He suggested instead the use of systems that reason from first principles, that is, from an understanding of causality in the system being examined. McDermott (see [48]) indicated that a primary reason ES development may fail is the lack of a theory on which the knowledge encoded in the ES is based.

One problem with using construct validity as a criterion is that conflicting or alternative theories or first principles may be available. This can lead to difficulties in establishing interexpert validation tests.

Objectivity

In variance terms, objectivity refers to minimizing observer variance [30, p. 491]. Accordingly, in validating an ES any judgmental variance should be minimized. This includes eliminating expert bias by administering the test independently, and using blinding techniques [7].

Programmer Validation. Because of the programmer's knowledge of the system and the nature of the development process, the system programmer generally will perform periodic informal validation on the system. If the programmer does not have a vested interest in the ES, this can be a way of developing an independent validation. However, programmers typically do not perform all the validation processes required. They may have a vested interest in the system, or they may not understand the problem being addressed by the system. This suggests using a combination of programmer and other validation.

Independent Administration of Validation. The importance of independence is recognized generally in research design. It also is recognized in accounting when a CPA performs an external audit or when internal auditors perform audits for internal purposes. If the model builder also validates the model, conflicts of interest can arise.

Sponsor/End User Validation. Software engineering uses an acceptance test by the sponsor or the end user as the final step in validating the computer program. Gaschnig, Klahr, Pople, Shortliffe, and Terry [21] suggested a similar type of formal test for an ES. That is, the user must "sign-off" on the system. If the end user has expertise in the area, then this acceptance test can be a critical test of user acceptance. However, if the sponsor or end user does not have sufficient expertise to judge the quality of the system, he or she may not be an appropriate judge.

Blinding Techniques to Eliminate Bias. Buchanan and Shortliffe [7] reported that some human expert validators are biased against computers making certain kinds of decisions. In order to minimize this limitation, the validation of MYCIN used a "blinded" study design to remove that source of bias.

Using Different Development and Test Data. Test problems often are used when developing systems to ensure consistent and reliable performance. However, it is important that the problems used to test the system not be limited to those problems used in developing the system; otherwise, these are not true test problems.

Economics (Cost-Benefit)

Cost-benefit decisions permeate research methods. For example, Simon noted "you will want to invest your time and energy in the work that will be the most valuable" [46, p. 100].

A key aspect of any economic activity is cost-benefit analysis. Since auditing- and accounting-based ESs are developed and validated using resources and may make decisions that have economic consequences, cost-benefit analysis is a major concern in their validation.

One important factor used to determine system benefit is how the system ultimately will be used. The system may be used for commercial purposes or it may be a prototype designed to investigate the feasibility of developing a system for a particular purpose. In either case, benefit may be difficult to measure.

Two of the main factors that determine the cost of validating a system are the formality and extent of validation required. Formality indicates the breadth of application of the validation—for example, is there an independent validation of the system? Extent indicates the depth of validation of the system—for example, how many test problems are used? Thus a prototype ES designed to determine whether a particular process can be represented as an ES will not receive a validation that has the same formality or extent as a commercially based system.

Cost-benefit affects not only the scope of validation but also the development of the system because the amount of validation defines the extent to which, for example, it can be determined if the knowledge in the knowledge base is correct or complete.

Reliability

One synonym for accuracy is reliability. "Reliability is the accuracy of precision of a measuring instrument" [30, p. 491]. In ES validation, the stability of the system and the ability of the system to generate identical solutions given identical inputs measures the reliability of the system.

System Test against Itself (Sensitivity Analysis). One possible validation procedure is the analysis of a program's sensitivity to slight changes in the knowledge base or in the weights [7]. That is, the model can be tested against itself for stability. If the system produces several solutions over minor parametric shifts, the system may be unstable. Alternatively, in certain environments a highly sensitive ES may be required. In this case it can be difficult to differentiate instability from an appropriate model response.

Standard Test Problems. Standard test problems may be used in the revalidation process. These problems should be designed to test standard system responses to ensure, for example, that additions to the knowledge base do not result in contradictions. The limitations of standard problems are discussed in [21].

Systematic Variance (Experimental Variance)

"If the independent variable does not vary substantially, there is little chance of separating its effect from the total variance of the dependent variable, so much of which is often due to change" [30, p. 308]. In validating an ES's performance, the test problems used need to allow the validator to distinguish between systematic variance and chance. This can be accomplished in a number of different ways.

Test Problems Reflect the Range of Problems Encountered. Test problems should be representative of problems the expert encounters in practice. Problems should reflect not only common "middle of the road" situations, but also some of the more unusual occurrences [8].

Variation in Test Problems. Validation success is based to a large extent on the test problems used in the validation process. Variation in the problems must exist if the entire set of parameters in the model and the range of the parameters is to be tested. If the problems are not varied enough in terms of the set and range of the parameters, the validation process and the test of any variations in the system's behavior will be too limited.

Number of Test Problems. A sufficient number of test problems must be used to ensure the statistical significance of the model. A test of only a few problems will not adequately test the knowledge base or range and set of parameters nor will it provide statistical significance [8].

Type I and Type II Errors. A Type I error occurs when we incorrectly reject a null hypothesis. A Type II error occurs when we accept a null hypothesis that is false. Both types of errors need to be considered when validating an ES. In some decision problems (e.g., bankruptcy–no bankruptcy) a large majority of the test problems will result in null hypotheses being accepted (and only a few rejected) because of the nature of the process itself. In these cases, there is a large possibility of Type II errors. In such a decision setting, there may be few test problems that actually test the abilities of the system. Accordingly, the test problems must be chosen carefully in light of this concern.

Extraneous Variance

"The control of extraneous variance means that the influences of independent variables extraneous to the purposes of the study are minimized, nullified or isolated" [30, p. 309]. At least five extraneous variables can influence the quality of system output: complexity of the system, location in the system life cycle, special fixes, location of the judges, and learning by the judges.

Complexity of the System. Complexity of the system is one of the most important variables in determining how difficult the validation process will be in software engineering. Software engineering uses a classification schema to measure the complexity of a computer program [44]: a simple program is one with fewer than 1,000 statements, written by one programmer, with no interactions with any other systems; an intermediate program is one with fewer than 10,000 statements, written by one to five programmers, with few interactions with other systems.

These criteria are not reasonable standards to use for evaluating ESs. Traditional software engineering computer programs do not use a knowledge base or an inference engine. In addition, in software engineering problems the process being modeled generally is well defined. Therefore, these software engineering classification schema cannot be used to evaluate ESs.

The criteria, however, can form the basis for other measures of complexity. Unfortunately, a ready replacement for general evaluation is not available. Some suggest the number of rules in the knowledge base as a standard, but this ignores connections between the rules and the inference engine. In spite of a lack of clear criteria, complexity remains an important variable in determining how difficult the validation process will be.

ES's Location in the System Life Cycle. Gaschnig et al. [21] suggested that the ES's current position in the development cycle is critical in determining the extent of validation desired. Validation is less critical in the early stages of development (but also less expensive and less difficult) than it is in later stages. In the early stages, a system is not very knowledgeable. Accordingly, simple validation tests will determine whether or not the system has an adequate set of working knowledge. In the latter stages, however, the ES may be facing adoption by external users and may require extensive validation. As a result, the ES's location in the life cycle is an important variable in determining the extent of the validation process.

Recognition, Examination, and Testing of Special Fixes. As noted in Fox, "experience with representing large varieties of knowledge show that a small fraction (< 10%) escape standard representation schemes, requiring specialized 'fixes'" [19, p. 282]. That is, in order to ensure that the knowledge base is complete, special features (fixes) are added to the system. These special fixes are of concern in ES validation. They need to be summarized so that special validation procedures can be developed to address them. Since these are "special" fixes, it is difficult to generalize any further.

Location of Judges during Testing. A critical distinction is made in research methodology between laboratory and field test settings. In the laboratory, many variables that may be faced by the judges who will be compared to the ES can be controlled. On the other hand, a field setting offers a richer decision-making environment and a more realistic test of the ES.

Learning during the Validation Process. Although most business-based ESs do not learn from processing decision problems, this is not necessarily true of the judges to whom the system is compared. For human judges, the ability to learn from the decision problems may be an extraneous variable. Learning could occur from the order of the test problems or from the type of test problems used. The amount of the learning due to order can be assessed by changing the order of the test problems. Learning that occurs from the process of making decisions on test problems is a more difficult problem and often is ignored in Turing tests. This problem can be addressed by determining whether an expert who has experience with the test problems makes decisions similar to those made by experts with less system experience.

Error Variance

In a discussion of error variance, Kerlinger noted

> There are a number of determinants of error variance, for instance, factors associated with individual differences among subjects. Ordinarily we call this variance due to individual differences "systematic variance." But when such variance cannot be, or is not identified and controlled, we lump it with the error variance. [30, p. 312]

This kind of variance can be caused by variation in the experts' responses from trial to trial, guessing, momentary inattention, slight temporary fatigue, lapses of memory, transient emotional states, etc. [30]. The validation framework for the ES's attempts to minimize error variance by controlling many of the variables in controlling the extraneous variance is the same as in all experimental designs. However, much of this minimization of error variance occurs by preventing errors and ambiguities, that is, by using proven measurement scales and unambiguous questions.

RELATIONSHIP OF PREVIOUS AUDITING AND ACCOUNTING ESS TO THE PROPOSED VALIDATION FRAMEWORK

A limited number of auditing and accounting-based ESs currently exist. The few that are being used (or are planned for use) on a commercial level (e.g., [50] and [45]) are proprietary. Accordingly, the present analysis is limited to *prototype* auditing- and accounting-based ESs.

The review and analysis of ESs in this paper is limited to those accounting and auditing systems generally available in the literature at the time the paper was written (February 1986) and to information published about the validation processes used in developing and designing those systems.[1] Accordingly, [3], [14], [22], and

[1]ESs examined in this paper were found by examining three sources: Peat, Marwick, Mitchell & Co.'s *Research Opportunities in Auditing* (1985) [42], Miller's *1984 Inventory of Expert Systems* [40], and Ph.D. dissertations through calendar year 1984 (listed in University Microfilms). In addition, some systems mentioned in presentations and papers the author was aware of also are included.

[24] are not discussed because they are unpublished papers; [3], [14], [20], [23], [29], and [34] are excluded because they contain little or no information about how the systems were validated.

The validation procedures used in those prototype ESs discussed in previous sections offer us a chance to examine the ability of our framework to capture the validation process. These findings are summarized in Table 2. Some framework items (economics and error variance) from Table 1 are not included in Table 2 because of the prototype nature of the systems. *This analysis is not to be considered as critical of these systems—each system was an innovative prototype effort.* The framework also can be used to analyze some of the potential research issues involved in ES validation.

Content Validity

The validators used examined the system's overall behavior rather than individual components. Future research needs to develop cost-benefit tools that allow individual components (such as the knowledge base) to be validated rather than simply comparing the system's overall performance to that of another expert [41].

The system tests analyzed each used human experts (in both interexpert and intraexpert tests) as the basis for validation. However, probably due to the prototype nature of the systems, only a small number of human experts generally were used. In addition, the use of interexpert tests may have introduced variance because the different experts used may have held contradictory views about the subject area.

In some cases, the small number of human experts available was supplemented by the use of alternative models or standard tests of validation. Unfortunately, developing an alternative model may cost more than having a human expert serve as a standard by responding to a set of test problems. Perhaps as a result, the comparison models analyzed were relatively unsophisticated. Generally, little research compares the performance of ESs to other sophisticated models.

Criterion Validity

The similarity of solutions proposed by the ES and the human expert(s) was used as the single criterion for success of the model. This standard ignored the possibility that an ES might derive better solutions than those of a human expert. It is not unusual for other types of models to outperform their human counterparts [26]. However, no such comparison has been made with ESs. If ESs can in fact consistently outperform humans, then other criteria will have to be developed to ensure appropriate validation.

Because the developers of these prototypes were involved in their validation, there apparently was little difficulty clarifying the evaluation criteria. However, little research has been done on developing effective evaluation criteria when outside validators are used.

In some cases, identifying the level of difficulty of the test problems established the expertise of the system. For example, AUDITOR [16] [17] [18] used test problems drawn from the work papers of audits. However, currently there is no general way of determing the system's level of expertise.

Table 2: Validation characteristics of selected accounting ESs.

	Clarkson [9] [10]	Bouwman [5]	Dungan and Chandler [16] [17] [18]	Michaelsen [38]	Steinbart [47]	Meservy [37]
1. *Content validity*						
Direct examination	No	No	No	No	No	No
Human expert comparison	Yes	Yes	Yes	Yes	Yes	Yes
Intraexpert test	Yes	Yes	No (multiple experts)	No (author was expert)	No	Yes
Interexpert test	No	Yes (18)	Yes (2)	Yes (2)	Yes (6)	Yes (3)
Other model test	Yes (random & naive)	No	Yes	No	No	No
2. *Criterion validity*						
Level of system	Expert	Expert	Expert	Expert	Expert	Expert
3. *Construct validity*	Largely empirical	Largely empirical	Weights generated empirically	Books	Audit manuals & textbooks	Prototypic reasoning
4. *Objectivity*						
Independent	No	No	Yes	Yes	Yes	Yes
5. *Reliability*						
Sensitivity	No	No	No	No	No	No
6. *Systematic variance*						
Actual problems	Apparently	NA (experiment)	Yes	Yes	Yes	Experiment similar to actual
Variation	Small range available; funds of 22k-37.5k	Not at extremes	small range; actual $ amounts	Chosen by subjects	Difficult to ascertain (likely)	Sought problems different than development problems
Sample Size	Two sets of four cases	Four sets of one case	11 cases (one company)	Two cases (one company)	Thirteen actual problems	Three sets of one case
Type I and type II errors	No	No	No	No	No	No
7. *Extraneous variance*						
Location of expert	Few controls	Laboratory	Office of expert	Office of expert	Office of expert	Unclear

The primary knowledge-base criterion used in the studies examined was accuracy of the system. However, little analysis was performed by the validators to ensure completeness or consistency of the system. In part, this may be because few methods for analyzing the completeness or consistency of a system are available. Future research could focus on developing such tools.

Construct Validity

Some of the prototype ESs were tied to a theoretical construct. However, some apparently were designed to discover how decisions are made, without the benefit of a normative decision-making model. The danger in this approach is that the model may be too "expert specific" with no ability to generalize or be compared to other expert systems or humans. On the other hand, of course, one of the primary research benefits of developing an ES is to understand better the nature of the particular decision process being modeled.

Objectivity

Generally, the validation processes used in these studies did not have an independent validator. Since the developers were involved in administering the validation, bias may have been introduced. Commercially developed packages, however, likely will have independent validation or at the least a user acceptance test.

Generally, no blinding techniques were used in administering the validation tests, and this may have affected the results.

Cost-Benefit Analysis

Each of the ESs examined was a prototype system. As a result, validation was minimally cost beneficial. As system development moves away from research-oriented prototypes, however, the cost-benefit relationship favors more validation. Generally, the costs of validation can be specified, but the benefits of validation are not as easily measured. Research could be directed toward measuring the benefits of validation.

Reliability

Since the ESs were prototype systems, there was no need to develop test problems for revalidation. However, for commercial systems, such test problems can be used after revision to ensure that no inconsistent or incorrect information has entered in the knowledge base. Test problems, of course, test only facets of the knowledge base. Research could be directed toward developing other methods to test for reliability.

Further, if revising the ES adds new knowledge to the knowledge base, can tools such as test problems be used effectively? With new knowledge in the system, a test problem may have different correct answers before and after revision.

Systematic Variance

A major recognized problem of auditing- and accounting-based ESs is the small number of test problems used in the sample. In addition, the studies looked at examined test problems from a "middle of the road" viewpoint. While many real-world situations are of this type, such a test does not examine system behavior in extreme situations and may not provide enough variation to test the system's overall behavior. This potential limitation can be overcome by broadening the range of test problem parameters.

Further, validation of the prototypes studied did not take into account the frequency of Type I and Type II errors in developing the test problems, nor did it consider the effect the number of the test problems used would have. Both omissions could produce a false sense of security in the test results for a system.

Extraneous Variance

The researchers in the prototype studies did not account for all extraneous system variables (e.g., complexity) in their validation efforts. Little research elicits or characterizes extraneous variables. Information on the special fixes and the validation of those special fixes also was lacking. This may be because a relatively small number of fixes were required or because the fixes were application specific. However, research into the extent of these special fixes could provide a guide to the amount of effort that should be expended in validating these special fixes.

The effect of learning on the part of the judges apparently was not a concern in validating these prototype ESs. The location of the judges did vary from study to study. The extent to which learning and location affect a judge's decisions would make an interesting research question.

CONCLUSION

This paper has developed a framework for validating ESs. The framework is based on research methods and can be used by ES developers to guide validation efforts. It also can be used to elicit further research topics in validation.

The framework addresses validation in terms of validity, objectivity, cost-benefits, and accuracy. Using previously developed ESs as examples, the framework elicited some directions for future research in validation. Those directions include the development of tools to aid in cost-benefit validation, analysis of intervening variables and their characteristics, and comparison of ESs to other models. In addition, the framework suggests some important validation questions including, "What validation criteria can be used if the ES is expected to outperform a human expert?" and "How can we best identify the system's level of expertise?" [Received: April 1, 1986. Accepted: January 28, 1987.]

REFERENCES

[1] Barr, A., & Feigenbaum, E. A. *The handbook of artificial intelligence.* Stanford, CA: Heuristech Press and Los Altos, CA: William Kaufmann, 1981.

[2] Bennett, J. (Ed.). *Building decision support systems.* Reading, MA: Addison-Wesley, 1983.
[3] Biggs, S. F., & Selfridge, M. *GC-X: A prototype expert system for the auditor's going concern judgment.* Presented at the University of Southern California Symposium on Expert Systems, Los Angeles, CA, February 1986.
[4] Blanning, R. W. Knowledge acquisition and system validation in expert systems for management. *Human Systems Management,* 1984, *4,* 280-285.
[5] Bouwman, M. J. Human diagnostic reasoning by computer. *Management Science,* 1983, *29,* 653-672.
[6] Bowman, E. H. Consistency and optimality in management decision making. *Management Science,* 1963, *9,* 310-321.
[7] Buchanan, B. G., & Shortliffe, E. H. *Rule-based expert systems.* Reading, MA: Addison-Wesley, 1984.
[8] Chandrasekaran, B. On evaluating AI systems for medical diagnosis. *AI Magazine,* 1983, *4*(2), 34-38.
[9] Clarkson, G. P. E. *Portfolio selection: A simulation of trust investment.* Englewood Cliffs, NJ: Prentice-Hall, 1962.
[10] Clarkson, G. P. E. A model of the trust investment process. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought.* New York: McGraw-Hill, 1963.
[11] Clocksin, W. F., & Mellish, C. S. *Programming in Prolog.* New York: Springer-Verlag, 1984.
[12] Davis, R. Reasoning from first principles in electronic troubleshooting. *International Journal of Man-Machine Studies,* 1983, *19,* 403-423.
[13] Davis, R. Diagnostic reasoning based on structure and behavior. *Artificial Intelligence,* 1984, *24,* 347-410.
[14] Dillard, J. F., & Mutchler, J. F. *Knowledge based expert systems for audit opinion decisions.* Presented at the University of Southern California Symposium on Expert Systems, Los Angeles, CA, February 1986.
[15] Duda, R. O., & Gaschnig, J. G. Knowledge-based systems come of age. *Byte,* September 1981, pp. 238-281.
[16] Dungan, C. *A model of audit judgment in the form of an expert system.* Unpublished Ph.D. dissertation, University of Illinois, 1983.
[17] Dungan, C., & Chandler, J. S. *Analysis of audit judgment through an expert system* (Faculty working paper no. 982). University of Illinois, College of Commerce and Business Administration, 1983.
[18] Dungan, C., & Chandler, J. S. Auditor: a microcomputer-based expert system to support auditors in the field. *Expert Systems,* 1985, *2*(4), 210-221.
[19] Fox, M. S. On inheritance in knowledge representation. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence* (Vol. 1). Menlo Park, CA: American Association for Artificial Intelligence, 1979.
[20] Fox, M. *Artificial intelligence in manufacturing.* Presented at the CPMS Seminar on Expert Systems, Pittsburgh, PA, December 1984.
[21] Gaschnig, J., Klahr, P., Pople, H., Shortliffe, E., & Terry, A. Evaluation of expert systems. In F. Hayes-Roth, D. A. Waterman, & D. B. Lenat (Eds.), *Building expert systems.* Reading, MA: Addison-Wesley, 1983.
[22] Grudnitski, G. *A prototype of an internal control expert system for the sales/accounts receivable application.* Presented at the University of Southern California Symposium on Expert Systems, Los Angeles, CA, February 1986.
[23] Hansen, J. V., & Messier, W. F. *A knowledge-based expert system for auditing advanced computer systems* (ARC working paper 83-5). University of Florida, Department of Accounting, 1985.
[24] Hansen, J. V., & Messier, W. F. *A preliminary test of EDP-XPERT.* Presented at the University of Southern California Symposium on Expert Systems, Los Angeles, CA, February 1986.
[25] Hayes-Roth, F., Waterman, D. A., & Lenat, D. B. (Eds.). *Building expert systems.* Reading, MA: Addison-Wesley, 1983.
[26] Hogarth, R. *Judgement and choice.* New York: Wiley, 1980.
[27] Hurst, E., Ness, D., Gambina, T., & Johnson, T. Growing DSS: A flexible, evolutionary approach. In J. Bennett (Ed.), *Building decision support systems.* Reading, MA: Addison-Wesley, 1983.
[28] Keen, P., & Scott Morton, M. S. *Decision support systems.* Reading, MA: Addison-Wesley, 1978.

[29] Kelly, K. P. *Expert problem solving for the audit planning process.* Unpublished Ph.D. dissertation, University of Pittsburgh, 1984.
[30] Kerlinger, F. *Foundations of behavioral research.* New York: Holt, Rinehart & Winston, 1973.
[31] Kidd, A., & Cooper, M. Man-machine interface issues in the construction and use of an expert system. *International Journal of Man-Machine Studies,* 1985, *22,* 91-102.
[32] Libby, R. *Accounting and human information processing: Theory and applications.* Englewood Cliffs, NJ: Prentice-Hall, 1981.
[33] Liebowitz, J. Useful approach for evaluating expert systems. *Expert Systems,* 1986, *3*(2), 86-99.
[34] McCarty, L. T. Reflections on TAXMAN: An experiment in artificial intelligence and legal reasoning. *Harvard Law Review,* 1977, *90,* 827-893.
[35] McDermott, J. *Background, theory and implementation of expert systems, II.* Presented at the CPMS Seminar on Expert Systems, Pittsburgh, PA, December 1984.
[36] McDermott, J. R1 revisted: Four years in the trenches. *AI Magazine,* 1984, *5*(3), 21-35.
[37] Meservy, R. D. *Auditing internal controls: A computational model of the review process.* Unpublished Ph.D. dissertation, University of Minnesota, 1985.
[38] Michaelsen, R. H. *A knowledge-based system for individual income and transfer tax planning.* Unpublished Ph.D. dissertation, University of Illinois, 1982.
[39] Michaelsen, R. H. An expert system for federal tax planning. *Expert Systems,* 1984, *1*(2), 149-167.
[40] Miller, R. K. *The inventory of expert systems.* Madison, GA: SEAI Institute, 1984.
[41] O'Leary, D. *Validation techniques for expert systems.* Presented at the ORSA/TIMS Meeting, Miami, FL, October 1986.
[42] Peat, Marwick, Mitchell & Co. *Research opportunities in auditing.* New York: Peat, Marwick, Mitchell Foundation, 1985.
[43] Rich, E. *Artificial intelligence.* New York: McGraw-Hill, 1983.
[44] Shooman, M. L. *Software engineering.* New York: McGraw-Hill, 1983.
[45] Shpilberg, D., & Graham, L. E. *Developing ExperTAP: An expert system for corporate tax accrual and planning.* Presented at the University of Southern California Symposium on Expert Systems, Los Angeles, CA, February 1986.
[46] Simon, J. *Basic research methods in social science.* New York: Random House, 1978.
[47] Steinbart, P. J. *The construction of an expert system to make materiality judgments.* Unpublished Ph.D. dissertation, Michigan State University, 1984.
[48] Texas Instruments. *The Second Artificial Intelligence Satellite Symposium.* Dallas, TX: Texas Instruments, 1986.
[49] Turban, E., & Watkins, P. Integrating expert systems and decision support systems. *MIS Quarterly,* 1986, *10*(2), 121-136.
[50] Willingham, J., & Wright, W. *Development of a knowledge-based system for auditing the collectability of a commercial loan.* Presented at the ORSA/TIMS Meeting, Boston, MA, 1985.
[51] Winston, P., & Horn, B. *LISP.* Reading, MA: Addison-Wesley, 1981.
[52] Yu, V., Fagan, L., Wraith, S., Clancey, W., Scott, A., Hannigan, J., Blum, R., Buchanan, B., & Cohen, S. Antimicrobial selection by computer: A blinded evaluation by infectious disease experts. *Journal of the American Medical Association,* 1979, *242,* 1279-1282. (See also Buchanan and Shortliffe [7, chapter 31].)

Daniel E. O'Leary is Assistant Professor in the Graduate School of Business, University of Southern California. Dr. O'Leary received his B.S. from Bowling Green State University, his M.B.A. from the University of Michigan, and his Ph.D. from Case Western Reserve University. His research interests include the theory of and methods for validating and assessing expert systems, and natural language interfaces. Dr. O'Leary has published or had accepted for publication papers on artificial intelligence and expert systems in various journals and books. He also has presented papers on artificial intelligence and expert systems at national and international symposia. Dr. O'Leary is the chair of the Measurement in Management Section of TIMS and the editor of *Expert Systems in Business and Accounting,* a newsletter published by the University of Southern California.