INFORMATION TECHNOLOGY DIVISION

Audit and Security Issues With Expert Systems

RESEARCH REPORT

AICPA
American Institute of Certified Public Accountants

Notice to Readers

This research report is the first in a series of in-depth reports that focus on the impact of particular technologies on the CPA's job and his or her organization. These reports are issued by the AICPA Information Technology Division for the benefit of Information Technology Section members. This report does not establish standards or preferred practice; it represents the opinion of the author and does not necessarily reflect the policies of the AICPA or the Information Technology Division.

Various members of the 1990–1991 AICPA Information Technology Research Subcommittee were involved in the preparation of this research report. The members of the subcommittee are listed below.

Raymond W. Elliott, *Chairman*
Terry Campbell
Rand B. Hammond
Everett C. Johnson, Jr.
George E. Love
J. Louis Matherne, Jr.
Daniel O'Leary
John E. Parker
J. Christopher Reimel, Jr.
Robert A. Sellers
Ira R. Weiss

The author gratefully acknowledges the contributions made to the development of this research report by subcommittee members Terry Campbell, Raymond Elliott, Everett Johnson, and Chris Reimel. In addition, the author would like to acknowledge Michael Harnish, Chairman, AICPA Information Technology Executive Committee; Jane Mancino, Technical Manager, AICPA Auditing Standards Division; and Robert Moeller, Chairman, AICPA Computer Auditing Subcommittee.

Richard D. Walker, *Director*
Information Technology Division

Nancy A. Cohen, *Technical Manager*
Information Technology Membership Section

1 2 3 4 5 6 7 8 9 0 IT 9 9 8 7 6 5 4 3 2

Library of Congress Cataloging-in-Publication Data
O'Leary, Daniel Edmund.
Audit and security issues with expert systems/Daniel O'Leary.
p. cm. — (Information Technology Division research report)
Includes bibliographical references and index.
ISBN 0-87051-124-6
1. Auditing, Internal — Data processing. 2. Accounting — Data processing. 3. Expert systems (Computer science) 4. Computer security I. American Institute of Certified Public Accountants.
II. Title. III. Series.
HF5667.12.043 1992
006.3'3—dc20 92-28385
CIP

Table of Contents

1. Introduction 5

What Are Expert Systems? 5
The Scope of This Report 5
The Structure of This Report 6

2. Unique Aspects of Expert Systems 7

Development Environment 7
Delivery Environment 8
Maintenance Environment 8
Software Limitations 9
Downward Delegation — Data-Base Access 9
Sources of Data 9
Symbolic and Numeric Information 9
Explanation of Knowledge to the User 10
User Interfaces 10
Well-Structured Vs. Not-Yet-Structured Problems 10
Users' Assumptions About Expert Systems' Emulation of Human Behavior 10

3. Auditing Expert Systems 11

Previous Research on Auditing Expert Systems 11
What Types of Systems Need to Be Audited? 12
Systems That Impact the Financial Statements 13
Systems That Can Impact the Going-Concern Status of the Firm 13
Systems That Impact the Audit 14
System Efficiency and Effectiveness 14
Systems That Impact the Security of Other Systems 14
Additional Reasons to Audit an Expert System 14
Techniques of Verification and Validation 15
Verification 15
Validation 16
Auditor Requirements 16

4. Security of Expert Systems 17

Previous Research on the Security of Expert Systems 17
Security Controls for Expert Systems 17
Development Environment 17
Delivery Environment 17
Maintenance Methodology 18

Software Limitations (Preventing and Detecting the Use of Knowledge to Camouflage Other Activities) 18
Downward Delegation — Access to Data Files 19
Source of the Data (Preventing and Detecting Use of the Wrong Knowledge) 19
Symbolic and Numeric Information (Preventing and Detecting the Changing of Knowledge) 19
Explanation of Knowledge (Preventing Leakage of Knowledge) 20

5. Additional Concerns Regarding Both the Auditing and Security of Expert Systems **21**

Integration of Expert Systems 21
Integration of Systems and Work Processes 21
Impact on Audit 22
Impact on Security 22
Nature of the Auditing Team 22
Need for Domain Knowledge 23
Need for a Knowledge of Expert Systems 23
Source of System Development 23
Finding the Systems 23
Use of Expert Systems 24
Attitudes Towards Expert Systems 24

6. Summary **25**

References **26**

Index **28**

Chapter 1. Introduction

More and more expert systems (ESs) are being developed for business and accounting applications. ES applications range from those for accounting and auditing, to those for tax and finance, to those for operations management and production. A survey of some of the auditing applications of expert systems is provided in O'Leary and Watkins (1989).

For example, TRW now uses an expert system to verify that individuals attempting to access a credit reporting system are valid users. Recent newspaper articles have focused on the potential litigation associated with the unauthorized use of credit reporting systems; accordingly, the protection of such systems from intruders is critical.

Security Pacific has developed an expert system to assist in identifying the fraudulent use of debit cards with automated teller machines (ATMs). If it determines that fraud is likely, the expert system can block the use of the ATM system.

The first two systems are intrusion-detection systems, which are designed to determine if a user is an intruder. Another type of expert system is used by Northwest Airlines to verify the pricing of the tickets that it issues. Using scanning, the system verifies between 50,000 and 60,000 tickets each night.

As these examples suggest, expert systems can serve as sophisticated tools for protecting and effectively managing valuable corporate assets. But expert systems are not just tools; they are also corporate assets in themselves and, as such, are vulnerable to tampering and increasingly require the consideration of the auditor. Accordingly, the purpose of this report is to provide some guidance to those concerned with the internal audit and security of ES applications, both in accounting and non-accounting applications.

What Are Expert Systems?

Typically, expert systems are computer programs that incorporate certain amounts of expertise or knowledge derived from human sources. These programs are used by a decision maker to assist in the decision-making process. In some cases, the systems function independently of a human user; however, such systems are rare.

Many systems are developed using expert system software referred to as an *expert system shell.* This software facilitates the development of representations of the knowledge gathered from experts. For example, an ES shell could help a developer design rules according to which knowledge would be stored (for example, "If condition *N,* then consequence *Q*"). These rules sometimes include a weight representing the "strength of association" or probability of a statement. Shells also contain an inference engine that sorts through the knowledge in order to find the answers to user inquiries.

ESs sometimes also interface with data bases. When this is the case, the systems can derive data from data bases to assist in the development of answers to user inquiries. Alternatively, users may be responsible for providing the data.

The Scope of This Report

Accounting researchers and system developers have been working with artificial intelligence and expert systems for almost ten years. During that time, accounting and auditing expert systems have gone from the lab to the field. Now, expert systems are available to auditors and tax accountants alike. In addition, auditors

are not only currently using expert systems in their audits, but are also facing expert systems that their company may be using in its operations. As a result, auditing such systems is an increasingly important consideration.

Unfortunately, both research on and experience with the auditing of expert systems are limited, which could have severe consequences. As noted by Moeller (1988, p. 8), "One would hope that we will not have to wait for an 'Equity Funding' type of event covering an expert system in order to have the impetus for sufficient audit guidance materials." (Although Moeller's article focuses on external audit considerations, his comments are of value to the internal auditor as well.) Thus, one purpose of this report is to investigate some of the primary concerns in the auditing of expert systems.

Another issue critical to the initial and ongoing success of expert systems is security. If an ES is not secure, it will be vulnerable to a loss of assets (possibly a loss in the particular application domain, for example, accounting, or a loss of knowledge to competitors). Further, it will also be vulnerable to a loss of system credibility: The system could be changed, for example, and might not function appropriately. There are other reasons for ensuring the security of an expert system as well, some of which are discussed later in this report. Thus, the second purpose of this report is to investigate some of the issues relating to the security of expert systems.

The Structure of This Report

This report proceeds as follows. Chapter 1 has provided an introduction. Chapter 2 analyzes the unique aspects of expert systems and some of the implications for the auditing and security of expert systems.

Chapter 3 examines the auditing of expert systems. A summary of some of the previous research on auditing expert systems appears, followed by an extensive analysis of the types of systems that may need to be audited after the risk assessment of each system has been taken into consideration. The chapter concludes with a brief summary of some verification and validation approaches developed by computer scientists that can be useful in the audit of expert systems.

Chapter 4 analyzes the security of expert systems. The primary focus in this chapter is on addressing the unique aspects of expert systems described in chapter 2 in the development of a set of control considerations for expert systems.

Chapter 5 investigates the implications of additional issues on both the audit and security of expert systems. Chapter 6 provides a brief summary of this report.

Chapter 2. Unique Aspects of Expert Systems

Expert systems are computer programs and should be treated as such. However, a number of their characteristics differentiate expert systems from other computer programs, creating new audit and security risks.

These unique characteristics include the development environment (the policies, procedures, and approaches used to develop an expert system), the delivery environment, the maintenance environment, the limitations of the software, the technique of downward delegation, the source of data used by an expert system, the simultaneous incorporation of symbolic and numeric information within an expert system, the explanation of the knowledge, the user interface of an expert system, the nature of the problem structured in an expert system, and the notion that an expert system emulates human behavior.

Much of the uniqueness deriving from these characteristics relates to the knowledge in an expert system. Knowledge in an expert system is gathered from a variety of sources, including human experts, questionnaires, and books. In any case, that knowledge is likely to be a valuable asset of the firm for which the expert system has been built. Therefore, it is necessary to establish appropriate controls to ensure that the knowledge is secure.

Some of these controls include those over the entry of knowledge into the system, the ability to access that knowledge for purposes of either changing it or controlling its "leakage," the solicitation of information that leads the system to use the wrong knowledge, and the use of knowledge to camouflage other knowledge (for example, Trojan horse programs that have names and functions that conceal their actual purpose).

In addition, it is necessary to audit an expert system's knowledge to ensure that it is correct and complete. Auditors have not been in a position to audit the quality of knowledge. As a result, new approaches need to be considered. These unique differences point to new audit concerns.

Since the auditing and security of traditional systems have been the subjects of many books and papers, this report will not address these topics. However, this report is concerned with those factors that impact all computer systems and have relatively unique implications for the security and auditing of an expert system. Four aspects of expert systems are identified in this regard:

1. The integration of expert systems with other systems.
2. The unique aspects of the audit and security teams.
3. The system developers (an outside vendor, an internal user, and so forth).
4. The users' attitudes towards an expert system.

These aspects are discussed further in Chapter 5.

Development Environment

In many cases, expert systems are developed by domain experts or users, not by a software engineer in the systems development department. For example, DuPont's well-known approach is to support the development of a large number of expert systems by individual users. Since the systems development department is not included in the process, it is not likely that an expert system will be developed with the same formality and structure of more traditional computer systems, nor is it likely that the development process or a system itself will be documented.

In addition, there is no generally accepted model of a life cycle or development methodology for expert systems. As a result, even if the development process was documented, there would be no one model with which the process

could be compared in order to determine its quality. Thus, it is often impossible for the auditor to satisfy himself or herself regarding the quality of a system based on the development process. Accordingly, there is little basis on which to audit the input or development process. Due to the limitations of the development process, more emphasis is placed on the testing of the system to ensure its quality. Tests of system quality generally are referred to as *validation and verification tests*, and are discussed later in this report.

ESs are normally developed using a "middle out" approach, also referred to as a *prototyping approach*. Successive versions of a system are developed iteratively as the problem at hand becomes better understood. Indeed, the prototype assists in developing a better understanding so that another version of the system can be developed. This process allows the developer to gradually determine the knowledge that is necessary for the system to function properly. This approach contrasts with more traditional software engineering approaches such as the "top down" or "bottom up" approaches.

Although prototyping has been found to be an excellent method of eliciting knowledge and gradually structuring a decision problem, it does introduce some security concerns. Researchers have found that prototyping makes managing and controlling the system development process more difficult. If managing the process is more difficult, it may also be more difficult to maintain security and easier for someone to sabotage the system.

Delivery Environment

When the user is the developer, the personal computer (PC) typically serves as the development and delivery environment. Further, since PCs can be taken almost everywhere and are generally easy to use, many ESs that interface directly with users are developed for a PC environment.

Given the portability and relative ease of use of PC-based systems, it is difficult to control the extent of a system's use. The ease with which PC software can be replicated and a PC can be accessed by other users is a primary concern. Systems can be developed and play an integral part in decision making without the internal auditors ever becoming aware of their existence. Even when the internal auditor is familiar with the development of a system, there is no guarantee that the extent to which it may be duplicated or used by others can be controlled. Moreover, ESs created for PC environments are generally developed for PC operating systems, which have few security devices. In addition, access controls in PCs and workstations generally are severely lacking.

Maintenance Environment

The maintenance of the knowledge base of an expert system raises a number of audit-related questions, including the following:

- Who can update the knowledge in the system?
- How is the updating process accomplished?
- Does the updating process jeopardize the quality of the knowledge?

The security of a system can be jeopardized if naive users are responsible for updating it. Although it is often suggested that knowledge can be freely entered and removed without disturbing the system, knowledge bases are fragile. The technical nature of the relationship between pieces of knowledge can be delicate and, without appropriate knowledge maintenance, is easily disrupted by naive users.

Software Limitations

A portion of knowledge in expert systems (typically less than 10 percent) escapes standard representation schemes and requires special fixes. These special fixes can take any of a number of designs. For example, in order to accommodate non-rule-based knowledge representation in a rule-based expert system shell, an external program or module may be used. In addition, a linear programming module or statistical module may be used to supplement or interface with the system.

Special fixes pose an audit risk, since they may require additional audit tests that are not required for the rest of the system being audited. Further, the possible existence of special fixes is an audit concern, since auditors will not perform additional tests if they have no knowledge of them.

Special fixes also pose a security risk because they offer the opportunity to hide knowledge that can result in unusual or dysfunctional behavior. Special fixes that involve additional modules, for example, offer the potential for intruders to hide Trojan horse programs.

Downward Delegation — Data-Base Access

By capturing expert knowledge in a computer system, expert systems allow expertise to be leveraged. Thus, ESs can be used to "delegate" decision making downward. However, instances of downward delegation can lead to situations in which the ES needs access to data for which the user may not have appropriate clearance. That is, the ES may have a higher priority than the user. As a result, either authorized or unauthorized access to the ES can yield unauthorized access to the data base and result in security problems.

Sources of Data

ESs solicit information directly from the user, a data base, or both. Most traditional computer programs function independently of the user, soliciting substantial amounts of information from data bases. Data are massaged and numerous steps are taken to ensure that the information is correct, depending on whether or not it is in a PC environment.

However, when data are solicited directly from the user, there is a substantial potential for error. Humans make errors of omission, misinterpretation, inconsistency, and other types of errors. If there is an error in the data input to the system, the ES may use the wrong knowledge, and it may develop inappropriate recommendations.

Symbolic and Numeric Information

In general, expert systems are designed to process both symbolic and numeric information. Knowledge may be represented, for example, in a rule-based format ("if condition *a*, then consequence *b*"), and the rules may include strengths-of-evidence or probability assessments.

Traditional computer programs typically process numeric data contained in data bases. In the case of numeric data bases, software is available that allows the auditor to investigate relationships in the data. However, in the case of expert systems, there are no similar tools, other than the expert system shell, to assist in the examination of the knowledge. In addition, numeric data may generally be easier to audit than nonnumeric data because they are not necessarily affected by the nuances of language and the nature of symbolic meanings. Thus, it can be more difficult for the auditor to get to the information in an expert system that needs to be audited. Since other types of computer programs do not contain knowledge used as data by an expert system program, there has been no previous investigation of ways to secure that knowledge, the costs of not securing it, and a variety of other concerns.

Explanation of Knowledge to the User

One important characteristic of many expert systems that differentiates them from other computer systems is their ability to explain their solutions and recommendations. Such explanations range from a trace of the sequence of rules that were used to come to the conclusions to specially developed explanation systems. These explanations reveal the insights of the experts from whom the knowledge was gathered.

Although insights into how a program makes a decision may be important to a decision maker, such insights may be used against the system. In the case of a system designed to assist in stock purchases, buy and sell rules could be discovered and used against the system. In the case of an audit system, insight into what transactions the system chooses to audit can be used against the system. Thus, the security of this area is another critical point in an ES.

User Interfaces

Typically, ESs have user-friendly interfaces. These interfaces make ESs easy to use with both natural-language and menu-driver interfaces. This feature can facilitate the auditor's ability to gather from the system information about its behavior. Unfortunately, that ease of use is not limited to authorized users, but also extends to unauthorized users. As a result, unauthorized users with only limited knowledge may gain access to the system. This can be a major security problem.

Well-Structured Vs. Not-Yet-Structured Problems

Most traditional computer programs are designed to solve well-structured problems that previously have often been structured as computer programs or as manual processes. For example, accounting programs, such as those for the accounts payable and accounts receivable functions, are well-established computer and manual applications.

On the other hand, most expert systems are designed for decision problems that are not yet structured or difficult to structure. Previous approaches to a given problem may include checklists or some other form of documentation or they may have no written structure. The lack of previous models can make the audit process a complicated one, since there may be no basis on which to assess system quality.

Users' Assumptions About Expert Systems' Emulation of Human Behavior

Since expert systems emulate human problem-solving behavior, the reaction of users to the embodiment of the expertise in a computer program may be a critical variable. For example, in the case of one expert system users began to assume that it would do all of the things that the humans it replaced would do. Unfortunately, the system was only designed to perform certain functions, and the remaining humans were expected to perform other functions for which the system had not been designed. As a result, some functions were not done.

Chapter 3. Auditing Expert Systems

The audit of expert systems has received only limited attention to date. This chapter summarizes some of that research. In addition, it provides a detailed analysis of the types of systems that may need to be audited. Finally, a brief discussion of the verification and validation of expert systems is provided.

The verification and validation of expert systems represent an active research area in computer science and artificial intelligence. Its focus is on providing methodologies to assist in determining the correctness, completeness, and decision quality of expert systems. As such, it provides a major audit tool. Chapter 4 summarizes another audit activity: analyzing the security of expert systems. Because the discussion of security is so extensive, it has been placed in its own chapter.

Previous Research on Auditing Expert Systems

There have been few direct references to auditing expert systems. However, at least four papers and one discussion from a book on electronic data processing (EDP) auditing address issues on auditing expert systems.

Moeller (1988) relates the audit of expert systems with existing audit literature, and discusses the relationship of Statement on Auditing Standards (SAS) No. 3, *The Effects of EDP on the Auditor's Study and Evaluation of Internal Control* (superseded by SAS No. 48 in July of 1984); SAS No. 48, *The Effects of Computer Processing on the Examination of Financial Statements*; and SAS No. 55, *Consideration of the Internal Control Structure in a Financial Statement Audit* to the internal control structure of expert systems. As part of his discussion, Moeller (1988, p. 7) notes that "while there is a growing body of other literature covering the auditor's use of expert systems, there is very little published material on audit techniques for reviewing expert systems." Moeller's paper summarizes many techniques and points to a body of work on those techniques that has developed recently: the literature on verification and validation. Moeller's perspective is that audits of expert systems should be aimed primarily at those systems in "financially significant applications." Moeller took this approach in recognition of the external auditor's role of attesting that financial statements are fairly stated.

Moeller's report suggests that in order to meet the unique requirements of expert systems, the audit of such systems could be accomplished using "conventional application control procedures." In particular, he provides a control analysis using input controls, processing controls, and output controls.

Kick (1989) discusses some of the risk exposures associated with expert systems resulting from a loss of strategic or competitive position, an inability to sustain growth, and a loss of strategic knowledge. The primary emphasis in his report is on ensuring that the auditor examine expert system applications to determine whether they are properly applied; are deployed to gain strategic advantage; are cost-effective; are well designed and operationally efficient; minimize exposure to fraud, poor decision making, and other consequences; are used by individuals who are properly trained; are easy to maintain; and are continually updated. These issues have been referred to as *effectiveness issues* in the expert system literature.

In order to accommodate these concerns, Kick suggests audit procedures that consist of the following steps:

- Examine selection priorities.
- Review development standards.

- Define roles and identify risks.
- Review the knowledge engineering and validation process.
- Evaluate efficiency and effectiveness.
- Evaluate the maintenance history.

Jamieson (1990) presents an analysis of the audit of expert systems in which he identifies a number of objectives:

- Identify the personnel relevant to an audit investigation.
- Identify the developer of the system.
- Present an expert system development life cycle.
- Review the evaluation of expert systems.
- Understand the security, control, and auditability requirements appropriate for an expert system environment.
- Review those mechanisms.
- Understand where auditors should be involved in the development process.
- Review documentation and legal concerns associated with expert system development.

Watne and Turney (1990) briefly analyze expert systems as a target of an audit. They suggest that systems that directly impact the balances in the financial statements or that provide information to the auditor individually be the potential targets of audits. They also analyze some of the controls in expert systems using a structure based on general and application controls. Watne and Turney also note that the computer science area of validation, discussed below, is the source of tests for the reliability and quality of the expert systems.

McKee (1991) investigates a theory of the demand for audits of expert systems. His investigation provides a market-based analysis of the factors leading up to voluntary audits. These factors include:

- The perception of conflict of interest.
- The importance of the consequences of the use of the system.
- The complexity of the system.
- The remoteness of those who might perform audits.

McKee's report also suggests that AICPA Statements on Standards for Attestation Engagements (SSAE), *Attestation Standards* might play a critical role in the audit of expert systems, and although it applies to independent CPAs, it may also provide useful guidance to internal auditors. *Attestation Standards* indicates that an audit should be done by someone who has adequate technical knowledge and proficiency, in this case in expert systems verification and validation, and in the specific domain. In addition, *Attestation Standards* indicates that the assertion is capable of being evaluated against reasonable criteria that either have been established by a recognized body or are stated in the presentation of the assertion in a sufficiently clear and comprehensive manner for a knowledgeable reader to be able to understand them.

What Types of Systems Need to Be Audited?

A critical issue in the internal audit of expert systems is the type of systems that may need to be audited. As is the case with traditional financial statement audits, those systems that impact the financial statements in a material manner require an audit. However, since expert systems also emulate human decision making, they can have a far-reaching impact on the internal auditor and on the firm that

uses them. This report identifies five basic types of systems that may need to be audited. (Of course, there may be other systems with which the internal auditor might also be concerned after having evaluated risk assessment.) The five basic systems are:

1. Systems that impact the financial statements
2. Systems that may impact the going-concern status of the firm
3. Systems that provide the auditor with information that is relied on during an audit
4. Systems for which efficiency and effectiveness are of concern to the auditor
5. Systems that impact the security of other systems

Systems That Impact the Financial Statements

In some cases, the auditor will find that an expert system impacts the financial statements directly. This impact can be observed in at least two ways. First, system activity may directly impact a particular account, such as loans. (For example, an expert system may be designed to assist in the loan-granting process.) Second, the system may perform the tasks of a human accountant, such as allocating costs and revenues to different accounts.

In the first case, a system designed to assist in the choice of those to whom banks lend can directly impact the quality of the loans. There are many other situations in which a system can impact the financial statements. For example, authorizing credit card transactions or insurance reimbursements (two well-established applications) can also impact the financial statements.

One aspect of these applications is the expert system's ability to provide an authorization for a set of transactions. This is typical of those transactions in which an activity has been delegated downward. When this is the case, expertise is captured and used by lower level personnel. It is critical that the expert system be audited when downward delegation exists, because the lower level personnel would have neither the knowledge to recognize unusual activities programmed into the system nor the authority to act on that knowledge to alter the system.

In the second case, the expert system may provide or manipulate accounting numbers in the same way that a human would. When this is the case, the program could include knowledge that leads it to make some inappropriate allocations. For example, revenues could be allocated to different periods in order to ensure a smoothness of income, or expenses could be allocated to the wrong accounts.

In either of these cases, the materiality of the activity would be a concern. If the levels were immaterial and the potential for fraud minimal, further auditing would not necessarily be cost-beneficial.

Systems That Can Impact the Going-Concern Status of the Firm

Many expert systems that are developed will not directly impact the financial statements, but they should be audited nevertheless. These systems include those whose activity is critical to the particular firm to the extent that their failure could force a change in its going-concern status. Such a change in status is a definite concern for the auditor.

For example, in the case of Van de Kamps (Los Angeles *Times*, September 12, 1990), a new computer system so disrupted deliveries that the firm was reportedly forced into bankruptcy. Although Van de Kamps was a privately held firm and the system that was implemented was not necessarily an expert system, the case does demonstrate the far-reaching impact that a computer system can have on going-concern status.

It is important to note that SAS No. 59, *The Auditor's Consideration of an Entity's Ability to Continue as a Going Concern,* does not contain any requirement that such systems be audited for financial statement audits. However, for internal audit purposes, as noted in the example, the audit of such systems may be critical to the continued existence of the company.

Systems That Impact the Audit

If expert systems are used in the audit process, they should be audited themselves. Otherwise, errors and biases may be introduced into the audit process through the use of the audit software.

System Efficiency and Effectiveness

Once an expert system has been developed and found to provide correct decisions, attention may be focused on alternative concerns, such as its efficiency and effectiveness. System efficiency relates to measures such as how well a system runs in its specific hardware and software environment. For example, when expert systems were first developed, many were so slow that the users did not want to operate them. The software with which these systems were built was very cumbersome. Accordingly, some systems were thought to be relatively inefficient.

System effectiveness concerns a system's ability not just to provide correct answers, but also to identify better answers. Effectiveness addresses the questions "Can the system be improved?" and "Does the system help users make better decisions?"

Systems That Impact the Security of Other Systems

Increasingly, expert systems are being used to provide security to other systems. These systems are used to prevent and detect intrusions into the systems that they protect. The extent of the control of security systems varies from advisory to independent operation. In one situation, a human uses output from an expert system to establish whether or not there was (or is) an intrusion. The system may not work fast enough, so that an intruder enters and exits the system being protected before any action can be taken. In another situation, the system operates on its own devices to determine whether or not someone is a potential or actual intruder. The system may lock out legitimate users. In either situation, the long-run viability of the system being protected may be questioned.

In some situations, security expert systems may fall under the set of systems, the failure of which could affect an entity's ability to continue as a going concern. If the security system is ineffective at keeping out intruders, it could be exploited or shut down, possibly leading to a going-concern issue.

Additional Reasons to Audit an Expert System

There are a number of reasons to audit an expert system. These reasons include economic conditions, specific market determinants, and other factors. From an economic perspective, audited information is worth more than unaudited information. Thus, one can expect to see an audit as a means of creating value in some circumstances.

Specific market determinants include conflicts of interest, the importance of a system's function, the complexity of a system, and insurance demands. A conflict of interest might occur when, for example, a system is to be sold for use in another firm. The purchasing firm might not believe all the claims of the selling firm because it perceives the other firm's vested interest in selling the system. In critically important situations, the life of an individual may depend on the

proper functioning of an expert system. In such situations, greater confidence would be given to an audited system. In other situations, an expert system may be extremely complicated. In these situations, an expert audit of the system may be able to turn up problems that normally would not have been found. Finally, an audit can provide a type of insurance, since the audit provides a basis for determining the accuracy of an expert system.

Techniques of Verification and Validation

The SSAE *Attestation Standards* states that an engagement can be performed only if (*a*) the assertion is capable of evaluation against reasonable criteria that either have been established by a recognized body or are stated in the presentation of the assertion in a sufficiently clear and comprehensive manner for a knowledgeable reader to be able to understand them, and (*b*) the assertion is capable of reasonably consistent estimation or measurement using such criteria. Again, while *Attestation Standards* does not apply to internal auditors, it may provide useful guidance.

In addition, *Attestation Standards* does not state the specific procedures that the practitioner should use. Instead, it focuses on the objective, which is to accumulate sufficient evidence to reduce attestation risk to a level that is, in the judgment of the auditor, appropriately low for the high level of assurance that may be imparted by the auditor's report. However, procedures should be chosen in order to assess internal control risk and restrict detection risk, in combination, so as to limit attestation risk to an appropriately low level (see paragraph 39 of *Attestation Standards*). Generally, these procedures will be derived from tests of verification and validation. Verification has been defined in much of the computer science literature as an examination of the completeness, correctness, redundancy, and consistency of software. This process as a whole has been called "building the system right." Validation has been defined as a determination of the quality of the decisions made by a system, as compared to specifications, human experts, or alternative criteria. The process of validation has been called "building the right system." Accordingly, it is evident that verification and validation are also concerned with the criteria and the consistency of estimation. An extensive list of basic concerns and tests is developed in O'Leary (1987).

This report briefly summarizes many of these techniques. However, the reader seeking more detailed information is referred to the references and to the extensive references in the papers given therein.

This discussion draws on the substantial research on verification and validation in the computer science literature. However, not all of the issues relating to verification and validation have been resolved. As a result, although some solutions are presented, the reader will have to look to future research for alternative and additional solution procedures.

Verification[1]

The purpose of verification is to determine whether or not the knowledge representation in an expert system is correct. Probably the most frequently used form of knowledge representation is the rule "if *a* then *b*, with certainty factor *y*." If one assumes that the knowledge in an expert system is represented in the form of rules, some of the verification tests can be specified.

Tests have been developed that can be used to determine when rules are incorrect, incomplete, redundant, or inconsistent. The rules are incorrect if they employ circular reasoning, as in the following example: "if *a*, then *b*; if *b*, then *c*; if *c*, then *a*." The rules are incomplete if there is a rule with no *a* or *b*. There is

[1] Nazareth (1989), Nyguyen et al. (1987), O'Leary and Kandelin (1988), and O'Leary (1990a and 1990b).

redundancy if there are multiple versions of the same rule in the same knowledge base. The rules are inconsistent if there are two rules such as the following: "if *a*, then *b*" and "if *a*, then *c*." In the last case, the occurrence of *a* leads to *b* and *c*, yet it is unclear which should be used.

There are also potential problems with the representation of uncertainty factors in expert systems. It has been found that developers of expert systems have difficulty using some of the schemes that have been developed to weigh the importance of the systems' rules. As a result, the weights often do not meet the appropriate underlying assumptions of probability theory.

Validation[2]

The function of validation is concerned with the quality of the decisions of the expert system. There are a number of approaches to assist the auditor in analyzing the validity of an expert system.

One of the most frequently used approaches is for the expert to directly inspect the knowledge. This approach could benefit from the development of a system that facilitates examination of the knowledge in the same way that audit software assists auditors in the examination of a data base. Such a system could allow the user to get a listing of the rules, or to get a pictorial network representation of the rules to assist in understanding how different rules are connected to each other.

Another approach is to treat the expert system as a black box and to test it against human experts. In such a test, the only concern is to determine the similarity of the judgments generated by the system and by the human expert. Typically, another human is used for purposes of comparison to determine which judgment is preferred. An alternative to this approach is to investigate the expert system to understand why it made certain judgments. When this is done, the explanation process plays a critical part in the process, in that it allows the reasoning used by the system to be analyzed.

Unfortunately, all of these methods require substantial human involvement. As a result, there has been a movement to develop alternative methods that require less direct human involvement. For example, O'Leary and Kandelin (1988) present statistical methods based on the weights of the rules in the expert system.

Auditor Requirements

The potential audit team associated with the audit of an expert system must have a broad base of knowledge. The team must possess knowledge of the application domain; otherwise it may be unable to determine when knowledge is incorrect. In one situation discussed in the *Wall Street Journal*, an expert on dams was thought to have withheld knowledge. However, at the time of the construction of the expert system, the developers had only a textbook knowledge of the domain. Thus, they were unable to assess the contribution of the expert.

The team must include someone knowledgeable about the tools of verification and validation in order to test the system. There is a broad literature of approaches deriving from artificial intelligence, computer science, and operations research.

Finally, the team must include someone with an understanding of auditing approaches and the requirements of auditing, and not just of expert systems. In addition, as discussed in the next chapter, an understanding of security needs is also critical.

[2] O'Leary (1987), O'Leary and Kandelin (1988), and O'Leary (1991).

Chapter 4. Security of Expert Systems

Although the analysis of the security of expert systems is an audit issue, it is treated separately, since security involves such a large set of issues. As noted below there is little, if any, specific research in the area of expert system security. The discussion presented here draws on the unique aspects of expert systems, as compared to other types of computer programs.

Previous Research on the Security of Expert Systems

There has been very limited research on the security of expert systems. In general, the closest discussion has been aimed at general EDP systems. Since expert systems essentially are computer programs, they require the same security measures as other computer programs. Many of the security concerns relating to traditional computer programs have been addressed in other sources (for example, Halper et al. [1985] and Weber [1988]) and, thus, are beyond the scope of this report. However, expert systems differ in numerous ways from other more traditional computer programs. These differences require the investigation of additional security concerns. Accordingly, the approach here will be to elicit some of these unique features and then discuss some of the controls that could mitigate the risks.

Security Controls for Expert Systems

The unique features of expert systems have a direct impact on the security of those systems. Some controls can be used with all types of expert systems and all types of expert system software shells.

Development Environment

The controls devised for the development methodology are those security measures designed to prevent and detect the entry of inappropriate or wrong knowledge into initial versions of an expert system. Although prototyping provides insights into the knowledge required for the problem-solving process, as noted earlier there may be problems managing the prototyping efforts. These management problems may result in security problems.

In addition, as expert systems move out of the lab and into production other problems may occur. As the number and size of expert systems increase, controls such as the maintenance of production schedules and quality assurance become critical to project management. Further, as an increasingly larger number of people become involved in the development of expert systems, the possibility that security problems will arise increases. The larger the number of people involved, the more likely it will be that one of them will attack the system.

In response to these concerns, some developers have recently suggested that a more traditional software engineering approach be used to develop expert systems. These methods are more structured and so may more effectively employ controls to mitigate some of these problems.

Delivery Environment

The personal computer environment, which impacts any PC application, has a number of security threats associated with it. Since so many ES applications run on PCs, controls for this environment are examined here. First, PC operating systems such as DOS have few security devices built into them. For example,

there is no general capability such as that provided by a system using passwords. Thus, unless such controls are a part of the ES shell, PC systems may be exploited by unauthorized users. In some cases, ES shells allow developers to embed passwords. For example, Guru, which is designed for use in a DOS environment, provides the designer with the ability to use passwords.

Second, most PCs are out in the open and easily accessed. Although locks are available for PCs, they are seldom used. Further, in some cases the only way to control access to a PC is to lock it in an office. The accessibility of PCs, combined with their easy-to-use interfaces, makes it easier to access knowledge.

Third, PCs are often brought out of the office to remote locations. The security of the expert system (and other systems) on a PC can be enhanced if the hard disk or diskettes are brought with the user whenever he or she leaves the PC.

Maintenance Methodology

The security of the maintenance of an expert system is accomplished, in part, using organizational controls. The primary organizational response that has been developed for expert systems is to designate an expert systems manager (ESM). The ESM is responsible for the overall operation and maintenance of a given ES. Conceptually, the ESM is similar to a data-base administrator. Designating an ESM allows for the assignment of responsibility, which is not the case with team-based approaches.

An important maintenance approach built into some ES shells and specific systems is to use verification tests. Verification tests are controls on the quality of the knowledge entered into particular ESs. For example, verification tests may be designed to ensure that there is no circular reasoning ("if *a*, then *b*; if *b*, then *c*; if *c*, then *a*"). As of this writing, the verification tests vary from shell to shell and from ES to ES.

In some large ESs it may be cost-beneficial to develop a specific system to assist in the updating and maintenance process. For example, in the case of EXPERTAX, a special maintenance system has been developed to assist in the process of updating the knowledge in that system.[1]

Software Limitations (Preventing and Detecting the Use of Knowledge to Camouflage Other Activities)

Programming software might not meet the needs of an application, requiring special fixes and separate modules. If the software does not meet the needs, one form of control is a statement to that effect in the documentation. Another general form of control is a requirement that such special fixes and the expected behavior of the systems be documented.

One approach to the prevention and detection of the inappropriate use of special fixes is to use so-called intrusion-detection systems.[2] These systems are called intrusion-detection systems because they are designed to either detect or prevent intrusions. They employ behavior patterns to establish expectations, which are then compared to actual behavior. Such systems consist of additional computer programs designed to monitor the use of expert systems for unusual activity, which may include unusual computer program activity (such as that which would occur with a Trojan horse program) or unusual user activity (such as the use of a system at an unusual time of day or for an unusual purpose).

[1] Shatz et al. (1987).

[2] See, for example, Denning (1987) and Tenor (1988).

Downward Delegation — Access to Data Files

Downward delegation may occur when an expert system has been developed using expertise that is gathered at a higher level in the organizational hierarchy than the one in which it is to be used. In such situations, the expert may have a higher priority or clearance level than the user for whom the system is designed. Thus, the ES and the user have different levels of data-base security.

One approach to this problem is to employ a data base for each ES, ensuring that the ES data base does not include any information to which its set of users should not have access. Controls can be established to ensure that users cannot derive data to which they do not have access.

However, doing so still does not eliminate the basic problems presented by having different priority levels. There appear to be few simple solutions to this problem. A more complicated approach is to develop a system that allows the ES to access data but does not allow the user to see that data. This approach fully accounts for the intermediary nature of the user.

Source of the Data (Preventing and Detecting Use of the Wrong Knowledge)

In some systems, the ES solicits data from the user. If the data are incorrect, the wrong portion of the knowledge base will be searched. Thus, it is critical that an ES provide controls on the data it solicits. These controls can include traditional data-edit controls (for instance, numeric field tests). In addition, they can include some specific application-based reasonableness tests. Such tests could include analyses of relationships between submitted data items (for example, those of the type "pay rate times hours worked = total pay") or analytical tests of reasonableness.

Symbolic and Numeric Information (Preventing and Detecting the Changing of Knowledge)

There are specific tests that can be used to prevent and detect changing either the rules themselves or the weights on the rules.

One preventive approach is to provide the user with only a run-time version of the system (for example, a compiled version). When this is done, the user can operate the system but not change it. However, in some cases, the user may need or receive a version of the system that is not a run-time version, or he or she may have a copy of the software with which the system was developed. When this is the case, providing a run-time version may not be feasible, so alternative approaches must be used. For example, if there are weights on the rules in an expert system, a scheme could be devised to capture information on the change of the weights. Each weight could be multiplied by the number of the rule, a prime number, or some other number. An unauthorized change of one or more weights would result in a change to the product, which could be used to detect any changes. Further, some controls could be numeric in nature — for example, the number of rules, the product of the number of words in a rule and the rule number, or some other number could provide a basis for detecting and controlling changes to the system.

Other traditional approaches such as base-case testing could be used to detect the possibility of a change. Unfortunately, unless the test data tested the portion of the knowledge that was changed, this approach may not work.

Explanation of Knowledge (Preventing Leakage of Knowledge)

The elimination of unnecessary leakage of knowledge can be critical to the continued success of an expert system, and perhaps to the continued success of the company that uses it. Two different controls are found in some types of ES shells that prevent the unwanted leakage of knowledge.

First, some shells offer the user the opportunity to provide an explanation that does not reveal the actual rules used by the expert system. These shells provide control over what is shown to the user of the ES, thereby controlling what is known about the system by the user. This approach may also be used to make a system easier to understand, since explanations (rather than system knowledge) are presented to the user. Second, in those cases where the explanations do not reveal the actual system knowledge, a run-time version can be used to control access to the knowledge.

Chapter 5. Additional Concerns Regarding Both the Auditing and Security of Expert Systems

So far, this report has focused primarily on the unique aspects of expert systems that affect auditing and security. However, expert systems are also computer programs and, thus, raise some of the same concerns that other computer programs do. This chapter examines some of these concerns that, although not unique to expert systems, have important implications for the audit and security of expert systems. In particular, this chapter is concerned with —

- The integration of expert systems.
- The nature of the auditor team.
- The attitude towards expert systems.
- The system developer.

Integration of Expert Systems

Since there are a number of reasons to expect that expert systems are different from other computer systems, the integration of expert systems with traditional computer systems deserves additional consideration. Previous research has exhibited a diversity of views on the impact that integration has on both what is audited and when it requires auditing. One author has suggested that "a system that is embedded in an accounts receivable application should be audited as a separate entity and not merely as a component of the accounts receivable system," while another gives the following example:

> ... [the] American Express expert system works under or is part of a much larger overall credit authorization system, a conventionally programmed application. While audit attention has almost certainly been given to that overall authorization system, it would not necessarily be given to the Authoriser Assistant subsystem. The auditor would give consideration to that subsystem only if it controlled a material amount of the receivable balances.

Integration of Systems and Work Processes

The classic expert system is a stand-alone system that is designed to solve a single problem or part of a problem. However, as they have evolved, expert systems have begun to be integrated into other more traditional computer systems. Thus, an important concern in the audit of an expert system is the extent to which the system is integrated with other systems or work processes. The integration of systems and processes is illustrated in the figure.

	Stand-Alone Problem	Integrated Problem
Stand-Alone System	A	B
Integrated System	C	D

Integration of Systems and Processes

Impact on Audit

For systems of type A, an audit of the system can be decomposed from much of the rest of the firm for independent assessment. Systems in this category might include a loan-approval system, in which many of each system's activities and users would be relatively independent of other activities and users.

However, for systems in categories B, C, and D, the expert system is part of a system, a process, or both. It interacts with other components of the system. Accordingly, the auditor would be concerned with the expert system as a component that interacts with the rest of the system components or work processes and with the overall system.

Systems in category B are distinguished from those in category A by the extent to which they complete work on a given problem. If the system only provides a solution for a part of the problem, as is the case in category B, there are a number of other persons or processes with which it must interface. In addition, there are a number of other inputs and outputs required for the overall process.

Systems in categories C and D are more complicated, since they are integrated with another system and possibly represent only a portion of the work process. The integration with another system requires that input, output, or both be exchanged between the systems in categories C and D and another system. The resulting interaction of multiple systems can complicate the audit of the expert system component, since the systems would likely employ different technologies. The impact of integration can be complicated further by the nature and audit requirements of the other systems. For example, if the other system is an accounting data base from which the accounting financial statements are generated, performing an audit becomes even more critical.

In any case, the primary concern should be the audit of the system. This audit would include an audit of any expert system component and an audit of the system as a whole. By saying that a component does not need an audit if it does not include a material level of activity, one suggests that no system will ever have sufficient materiality to require an audit if the components are made small enough.

Impact on Security

The degree to which an expert system is integrated with other systems is also critical to its security. If an expert system is embedded in another system with substantial security controls, the expert system can also benefit from those controls. However, if an expert system is embedded in a system with a lesser degree of security than would normally be accorded to the expert system, the expert system faces exposure, which would not normally be a concern.

However, even if an expert system is embedded in a system that is in a very secure environment, the threats faced by the expert system may be different. For example, a primary threat to an expert system could be posed by a lack of control over access to its knowledge base. In such a situation, as it is in auditing, it is important to consider both the individual expert system and the overall system in which it is embedded.

Nature of the Auditing Team

Currently, there is a relatively small number of EDP auditors (auditors who specialize in the audit and security analysis of computer-based systems). This specialty has developed over time in response to the need for auditors who understand the complicated audit and system environments of EDP operations.

In a similar sense, there is a need for expertise in the audit and security of expert systems. The audit and security of expert systems require knowledge both of the domain in which the system is built and of expert systems technology.

Need for Domain Knowledge

Since an expert system captures domain knowledge, it is critical for an auditor to be able to understand that knowledge. If the auditor does not possess any knowledge of the domain on which a system is based, it would be very difficult for him or her to assess the quality or correctness of the knowledge used by the expert system.

The importance of gaining an understanding of an expert system's domain became evident in the course of the development of such systems. When an expert system is developed, the developers typically become "near-experts." Substantial case evidence indicates that even when the developers are not near-experts at the beginning of the development process, they are by the end of the process. For example, to build a value-added-tax (VAT) expert system, knowledge engineers found it necessary to travel with the VAT auditors for over a year. It took that long for the developers to obtain a sufficient understanding of the domain to develop a system. (Today of course, educating developers to such a degree would probably be too costly.)

Need for a Knowledge of Expert Systems

While substantial computer software has been developed to assist auditors in their analysis of data-base systems, no such software has been developed to assist in the analysis of knowledge bases. If an auditor does not know about the process of knowledge representation or about expert systems in general, it would be very difficult for him or her to assess the quality or correctness of the knowledge representation in an expert system. In addition, it could be very difficult for an expert to investigate aspects of an expert system such as efficiency and effectiveness if the auditor is unfamiliar with the technology. A lack of familiarity with the technology can also limit the auditor's understanding of potential security difficulties.

Source of System Development

As is the case with other computer systems, expert systems can be developed internally or they can be purchased from a vendor. The problems associated with user-developed expert systems include inadequate documentation, a lack of concern for security, control or audit matters, poor programming, and insufficient testing and evaluation of knowledge bases. (Similar problems are associated with systems developed in other environments, including research labs.) Typically, problems like those listed above indicate to the auditor that further testing of the system at hand is required. However, there are additional concerns associated with user-developed systems, including determining the existence and extent of use of such systems.

Finding the Systems

Systems developed by research labs or systems development departments (as well as systems purchased from outside vendors) often leave a clear trace of their existence and use. However, user-developed systems do not always leave such a trace. Although user-developed expert systems may be embedded in decision processes in a PC environment, their presence in a PC environment does not automatically make their applications immaterial. As a result, the auditor should take steps to identify the existence of these systems. Obviously, an expert system can be audited (and its security needs can be assessed) only after it has been identified.

Use of Expert Systems

Given the ease with which PC programs can be copied and implemented, auditors should not be surprised to find that multiple copies of an expert system may be dispersed throughout the firm. When this is the case, the auditor's consideration of materiality should not be limited to the single copy of the software that he or she has identified. The possible existence of any additional copies should be investigated.

Attitudes Towards Expert Systems

As noted earlier, expert systems emulate human decision making. The very words *expert system* draw an analogy between a human and a computer program. Accordingly, users of an expert system or those who interface with it may assume that the system is more than just a computer program.

For instance, in the case of one expert system embedded in a work process of other computer programs and users, it wasn't clear to the users what the computer program could and could not be expected to do. People who interfaced with the system assumed that since it was an expert system, it would perform a set of activities as thoroughly and completely as a human would. However, the system was specialized to accomplish a subset of a human's activities. The humans who interfaced with it were expected to do some of the activities that they assumed the system would do.

The attitude towards an expert system can impact its audit and security. Thus, compliance audit activity may be emphasized in the audit of an expert system. The observation that a system can perform its set of activities according to expectations does not guarantee that the overall set of expert system and human activities will be done according to plan.

The security of an expert system could also be jeopardized when users make assumptions about its security. For example, if a system has some intelligence, users might assume that this capability extends to the system's security. Such an assumption could easily be unfounded.

Chapter 6. Summary

This report has examined some of the issues in the audit and security of expert systems. Since expert systems are computer programs, they face some of the same audit and security concerns as other computer programs. However, expert systems have a number of unique characteristics, many of which were summarized in this report. These unique characteristics require that special consideration be given to the auditing and security of expert systems. In addition, they also form the basis for the generation of some audit and control procedures.

One of the primary issues addressed in this report is identifying the systems that should be audited. At least five different types of systems have been identified: systems that impact the financial statements, systems that impact the going-concern status of the firm, systems that provide information that the auditor will rely on in an audit, systems for which efficiency and effectiveness are of concern to the auditor, and systems that impact the security of other systems.

Tools that assist in the audit process include those developed by computer scientists and artificial intelligence researchers to verify and validate computer programs. Verification and validation are designed to ensure that a system is "built right" and that the "right system is built." Verification and validation provide the basis for a sequence of tests of the quality of the system.

Some features of expert systems that are also common to other computer systems can assist in determining critical audit and security activities. In particular, the degree of an expert system's integration with traditional systems, the nature of the audit team, the source of the expert system development effort, and the users' attitudes towards an expert system also impact important audit and security steps.

References

Bull, M., et al. "Applying Software Engineering Principles to Knowledge Base Development." In *Proceedings of the First Annual Conference on Expert Systems in Business*, 27–38. New York: Learned Information, Nov. 1987.

Davis, D. "Artificial Intelligence Goes to Work." *High Technology*, Apr. 1987.

Denning, D. "An Intrusion Detection Model." In *IEEE Transactions on Software Engineering*, vol. SE 14, no. 3, 252–261. New York: Institute of Electrical and Electronics Engineers, Mar. 1987.

Denning, D., et al. "Views for Multilevel Databases." In *IEEE Transactions on Software Engineering*, vol. SE 13, no. 2, 129–139. New York: Institute of Electrical and Electronics Engineers, Feb. 1987.

Fox, M. "Artificial Intelligence in Knowledge Representation." In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, vol. 1, 282–284. Palo Alto, Calif.: Morgan Kaufmann, 1979.

Halper, S., G. Davis, P.J. O'Neil-Dunne, and P. Pfau. *Handbook of EDP Auditing*. Boston, Mass.: Warren, Gorham & Lamont, 1985.

Holsapple, C. and Whinston, A. *Business Expert Systems*. Homewood, Ill.: Irwin, 1987.

Jamieson, R. "Auditing Knowledge Based Systems." Monograph presented at the EDP Auditors Foundation, University of New South Wales, Sydney, Australia, Jan. 1990.

Kick, R. "Auditing an Expert System." *Expert Systems*, Summer 1989, 33–38.

Lethan, H. and H. Jacobsen. "ESKORT — An Expert System for Auditing VAT Accounts." In *Proceedings of Expert Systems and Their Applications*. Avignon, France 1987.

McKee, T. "An Audit Framework for Expert Systems." *Expert Systems Review* 2 (no. 4, 1991).

Michaelsen, R. *A Knowledge-based System for Individual Income and Transfer Tax Planning*. Champaign, Ill.: University of Illinois, 1991.

Moeller, R. "Expert Systems: Auditability Issues." Paper presented at the 1st International Symposium for Expert Systems in Business, Finance and Accounting, Oct. 1988. Publication forthcoming in *Expert Systems in Business and Finance*. New York: John Wiley.

Nazareth, D. "Issues in the Verification of Knowledge in Rule-based Systems." *International Journal of Man-Machine Studies* 30 (1989), 255–271.

Nyguyen, T., W. Perkins, T. Lafferty, and D. Pecora. "Knowledge-based Verification." *AI Magazine* 8 (no. 2, 1987).

O'Leary, D., 1987. "Validation of Expert Systems: With Applications to Accounting and Auditing." *Decision Sciences* 17 (no. 3), 468–486.

O'Leary, D., 1988a. "Expert Systems Prototyping as a Research Tool." In E. Turban and P. Watkins, *Applied Expert Systems*. Amsterdam: North-Holland, 1988.

O'Leary, D., 1988b. "Methods of Validating Expert Systems." *Interfaces* 18 (no. 6), 72–79.

O'Leary, D., 1988c. "Software Engineering and Research Issues in Accounting Information Systems." *Journal of Information Systems* 2 (no. 2, Spring 1988).

O'Leary, D., 1990a. "Soliciting Weights or Probabilities from Experts for Rule-Based Expert Systems." *International Journal of Man-Machine Studies* 32 (1990), 293–301.

O'Leary, D., 1990b. "Verification of Frames and Semantic Network Knowledge Bases." In *Preproceedings of the 5th Knowledge Acquisition for Knowledge-based Systems Workshop,* Banff, Alta., Nov. 1990.

O'Leary, D., 1991a. "An Exploratory Approach to Assessing the Efficiency and Effectiveness of Expert Systems." *International Journal of Intelligent Systems in Accounting, Finance and Management.* Publication forthcoming in 1992.

O'Leary, D., 1991b. "Design, Development and Validation of Expert Systems: A Survey of Developers." In *Verification, Validation and Testing of Expert Systems.* New York: John Wiley, 1991.

O'Leary, D., 1991c. "Knowledge Discovery as a Threat to Database Security." In *Knowledge Discovery in Databases.* Cambridge, Mass.: MIT Press, 1991.

O'Leary, D., 1991d. "Measuring the Quality of an Expert System's Performance." *European Journal of Operational Research,* Feb. 1991.

O'Leary, D. "Measuring the Quality of Computer Model Performance." Publication forthcoming in *European Journal of Operational Research.*

O'Leary, D. and N. Kandelin, "Validating the Weights in Rule-based Expert Systems: A Statistical Approach." *International Journal of Expert Systems* 1 (no. 3, 1988).

O'Leary, D. and P. Watkins. "Review of Expert Systems in Auditing." *Expert Systems Review* 2 (nos. 1 and 2, 1989).

Ribar, G. "Development of an Expert System." *Expert Systems Review* 1 (no. 3, 1988).

Shatz, H., R. Strahs, and L. Campbell. "Expertax: The Issue of Long-Term Maintenance." In *Proceedings of the 3d International Conference on Expert Systems,* 291–300. Oxford, England: Learned Information, June 1987.

Socha, W. "Problems in Auditing Expert Systems." *The EDP Audit, Control and Security Newsletter,* Mar. 1988.

Tenor, W. "Expert Systems for Computer Security." *Expert Systems Review in Business and Accounting* 1 (no. 2, 1988).

Watne, D. and P. Turney. *Auditing EDP Systems.* Englewood Cliffs, N.J.: Prentice-Hall, 1990, 555–590.

Weber, R. *EDP Auditing.* New York: McGraw-Hill, 1988.

Index

A

AICPA Statements on Standards for Attestation Engagements (SSAEs), 12, 15

Auditing ESs
- additional reasons for, 14–15
- auditor requirements, 16
- financial statements, impact on, 13
- going-concern status of firm, impact on, 13
- impact on audit, 22
- nature of auditing team, 22
- previous research, 11–12
- security of other systems, impact on, 14
- systems impacting audit, 14
- types of systems, 12–13
- verification and validation
  - validation, 16
  - verification, 15–16

Auditing team, 22
- EDP auditors, 22

Auditor requirements, 16

C

Copying of programs, investigation, 24

D

Data, sources of, 9

Delivery environment, 8
- security of ESs, 17–18

Development environment, 7
- prototyping approach, 8
- security of ESs, 17
- "top down" and "bottom up" approaches, 8

Domain experts, 7

Domain knowledge, 23

Downward delegation of decision making, 9
- security of ESs, 19

E

Emulation of human problem solving, 10

Expert system shell, 5

Expert systems (ESs)
- attitudes towards, 24
- auditing (*See:* Auditing of ESs)
- defined, 5
- integration of, 21
- security (*See:* Security of ESs)
- unique aspects (*See:* Unique aspects of ESs)
- use of, 24

Expert systems manager (ESM), 18

F

Financial statements, impact on, 13

G

Going-concern status of firm, impact on, 13

I

Intrusion-detection systems, 5, 18
- ATMs, fraudulent use, 5

K

Knowledge
- changing of knowledge, detecting, 19
- domain knowledge, 23
- explanation to user, 10
- leakage of, 20
- software limitations, 9
- use to camouflage other activities, 18
- wrong knowledge, detecting, 19

Knowledge of expert systems, need for, 23

M

Maintenance environment, 8
- and audit-related questions, 8
- security of ESs, 18

Maintenance methodology, 18

P

Personal computers (PCs) delivery environment,
- use as, 8
- security problems, 17–18

Problems
- well-structured and not-yet-structured, 10

R

Research
- and auditing ESs, 11–12
- security of ESs, 17

S

Security of ESs
- controls, 17
- data source, 19

Security of ESs (cont.)
delivery environment, 17–18
development environment, 17
downward delegation, 19
explanation of knowledge, 20
integration with other systems, 22
maintenance methodology, 18
previous research, 17
security of other systems, impact on, 14
software limitations, 18
symbolic and numeric information, 19

Software limitations
security of ESs, 18
special fixes, use of, 9

Special fixes, 18
audit risk, 9

Stand-alone system, 21
integration with problem, 21

Symbolic and numeric information, 9, 19

System development
finding user-developed, 23
sources, 23

T

Trojan horse program, 18

U

Unique aspects of ESs
as computer program, 7
data, sources of, 9
delivery environment, 8
development environment, 7–8
downward delegation of knowledge, 9
emulation of human problem-solving, 10
explanation of knowledge to user, 10
maintenance environment, 8
not-yet-structured problem solving, 10
software limitations, 9
symbolic and numeric information, 9
user-friendly interfaces, 10

User-friendly interfaces, 10

V

Validation and verification tests, 8, 15–16

Value-added-tax expert system, 23

About the Author

Daniel E. O'Leary received his Ph.D. from Case Western Reserve University and his MBA from the University of Michigan. Dr. O'Leary is on the faculty of the School of Accounting at the University of Southern California.

Dr. O'Leary is the author of a forthcoming book, *Expert Systems and Internal Auditing*. He has edited other books, including *Expert Systems in Finance*, published by North-Holland. Dr. O'Leary is the editor of the John Wiley journal *International Journal of Intelligent Systems in Accounting, Finance and Management*.

He has been on the program committee of a number of meetings on the use of artificial intelligence. He is the co-chair of the upcoming Fifth International Symposium on Expert Systems in Accounting, Finance and Management, the oldest symposium on expert systems in business.

Dr. O'Leary has served on the AICPA Information Technology Research Subcommittee and on a United Nations subcommittee formed to study the impact of information technology on member nations.