

Expert Systems and AI-based Decision Support in Auditing: Progress and Perspectives

William E. McCarthy
Michigan State University, USA

Eric Denna
Brigham Young University, USA

Graham Gal
University of Massachusetts, USA

Stephen R. Rockwell
University of Florida, USA

'When all the AI rhetoric is boiled away, expert systems are simply computer programs much like general ledger packages or even like video games. Writing a new payroll program in COBOL is not research, and neither is building another auditing expert system.'

Introduction

Since the development of AUDITOR at Illinois there have been a number of auditing expert systems designed and built by both academics and accounting professionals. For surveys of this work, see Messier and Hansen (1987), Gal and Steinbart (1987) and Bailey *et al.* (1987, 1988). However, as encapsulated by the statement above, a continuing criticism of this work (indeed, a criticism of any knowledge-based work in accounting) is that it constitutes more development than research. In this paper we contend that such blanket criticisms are unfounded and are in fact more attributable to

a critic's lack of schooling in computer science than to any conceptual shortcomings in the actual systems research methods. More specifically, we will look at several auditing expert systems and evaluate them in terms of some informally developed differentiation heuristics, heuristics whose rationale depends heavily on the work of March (1988) and Cohen and Howe (1988). We will also try to chart new directions for research in knowledge-based auditing systems. Our central purpose throughout this paper is to attempt to develop a framework of analysis so that when someone proposes a new audit expert system or enhancements to an existing audit expert system we can type its contribution as either primarily research or primarily development or both.

The organization of the paper is as follows. The next section will explore the cognitive modeling rationale for AI-based research in auditing. This reasoning is critical to our analysis framework, but it has been explicated in detail elsewhere. It will only be reviewed

and summarized here. The third section will explore the software engineering legitimacy of knowledge-based audit systems, i.e. a rationale that is quite different from the cognitive modeling approach of most accounting researchers. This section will explore that rationale as adapted from a framework developed by March (1988) and as augmented by other considerations gleaned from the work of researchers such as Cohen and Howe (1988). The three subsections of this software engineering segment will address in order: (1) the March framework, (2) a set of arguments concerning domain specificity and maturity of the research field and (3) some considerations involved in deciding whether to build an entire system or to prototype just part of it. The fourth section will explore time-lined development of four academic audit expert systems and contrast their research content with that of three bogus projects. The rationale developed previously in the second and third sections will be used in the comparison of these four real systems and three straw men. The fifth section will explore the perspective of the audit practitioners in AI tool development, and it will examine briefly areas where academics and practitioners can work together. The final section will contain a summary of our arguments.

Cognitive Modeling Rationale

A central theme which underlies the discipline of accounting is the belief that accounting information influences decision-making processes. This orientation has led both academicians and practitioners to be concerned with improving decisions that fall within the accounting domain. There are basically three different approaches that can be used to improve a decision. The first is to provide better information. A second is to train the decision maker to use the current information set more effectively. Finally, the decision maker can be replaced with a device that produces a consistent decision according to some prescribed model (Libby, 1981). An initial issue that must be resolved prior to taking any of these actions is to understand the current approach used to make the decision in question so that deficiencies, if they exist, can be evaluated. As a group, the decisions made by

auditors have been used as the primary focus of a number of projects as accounting researchers seek to understand the auditing decision process. In recent years, the information-processing paradigm has been used in an increasing number of these projects as researchers seek to uncover different aspects of the auditing decision process.

When auditors make a decision concerning the state of internal controls or the importance of a particular account balance to the completed financial statements they must collect information, combine it using some process and then finally produce a decision. The information-processing paradigm offers a number of different approaches to investigate these activities. A researcher can ask auditors to verbalize what they are doing as they make decisions. These verbal reports (Ericsson and Simon, 1980) provide a trace of the steps that the auditor goes through, and thus give insights into the information used, the combination processes employed and the decisions produced. This verbal trace of problem-solving activities becomes a model of the underlying cognitive process.

A difficulty with this approach is that it is hard to verify the model. This deficiency has led certain accounting researchers to use tools and methods borrowed from computer science in an attempt to implement the model of the auditor on a computer in the form of a program that simulates the auditor's decision process. The rationale for building these systems is that the researcher now has a program which contains a cognitive model of the decision maker and can proceed with an assessment of which of the three approaches mentioned above would be appropriate to improve the decision. That is, should we change the information or should we train auditors to use a different process or, finally, should we use the expert system to replace the auditor?

As noted by Bailey *et al.* (1987), cognitive modeling has certainly provided the dominant justification for most expert systems work in auditing, and it is the rationale most easily accepted by mainstream accounting researchers. We turn now to a less well-known (in accounting) justification for construction and use of AI tools in this area: the software engineering rationale.

Software Engineering Rationale

In describing the scope of *empirical* AI (as opposed to *applied* AI) endeavors and in contrasting its methodological differences with those of traditional behavioral science, Cohen and Howe observe that: 'Whereas . . . much research in the behavioral sciences is concerned with teasing apart the components of behavior and their causal interrelationships, empirical AI is concerned with putting those components together in one box to produce behavior' (1988, p. 18). These researchers go on to say that the task of empirical AI researchers 'is not to find out [by statistical induction] how the average human organism (or organization) works; but rather to build artificial systems that work in particular ways' (p. 19). By building such carefully delineated systems they contend that we can produce useful generalizations deducible from explanations of AI theory. Cohen and Howe's thoughts in this regard echo sentiments expressed a number of years earlier by Newell and Simon (1976) who, in their famous Turing Award Lecture, contended that the purpose of AI research was to enrich our collective store of concrete experiences with specific classes of symbolic processing systems, and to use that collective store to reason across domains about the general characteristics of intelligence and its method of implementation (p. 126). In this same address, Newell and Simon also spoke of the confusion surrounding the scope of basic research in computer science:

Computer science is an empirical discipline. . . . Each new program that is built is an experiment. It poses a question to nature, and its behavior offers clues to an answer. Neither machines nor programs are black boxes: they are artifacts that have been designed, both hardware and software, and we can open them up and look inside. We can relate their structure to their behavior and draw many lessons from a single experiment . . . We build computers and programs for many reasons. We build them to serve society and as tools for carrying out the economic tasks of society. But as basic scientists, we build machines and programs as a way of discovering new phenomena

and analyzing phenomena we already know about. Society often becomes confused about this, believing that computers and programs are to be constructed only for the economic use that can be made of them (or as intermediate items in a developmental sequence leading to such use). It needs to understand that the phenomena surrounding computers are deep and obscure, requiring much experimentation to assess their nature. It needs to understand that, as in any science, the gains that accrue from such experimentation and understanding pay off in the permanent acquisition of new techniques; and that it is these techniques that will create the instruments to help society in achieving its goals (p. 114).

Justifications such as these form the basis for what we call the software engineering rationale for AI research in accounting and auditing. Stated differently, we believe that efforts aimed at building knowledge-based systems in new and innovative ways in previously unexplored task areas can be viewed as research, even if the results of such efforts do not strongly mimic the behavior of a human expert in that particular domain. Computer software in general, and AI programming in particular, can legitimately be defended as an ending goal of accounting research, not just a means to some other end such as the test of a certain behavioral or economic theory. In the subsections that follow we explore different heuristic frameworks that can be used to classify endeavors in this vein as either research or development.

The March Framework

In a speech given at ICIS-88, Sal March (the present editor of *Computing Surveys*) outlined his framework for identifying information technology issues for information systems researchers. That framework is reproduced in Figure 1, and his explanation is given as follows.

My general framework for research in information technology is two dimensional. The first dimension is an engineering paradigm: *build* an artifact to perform a particular task, *evaluate* the performance of that artifact

	BUILD	EVALUATE	PROVE
REPRESENTATION			
METHOD			
TOOL			

Figure 1 The March framework (March, 1988).

(develop performance measures and collect data to evaluate those measures), and *prove* the performance of the artifact (superior to another tool or optimal in some sense). The second dimension is a problem solving (methodological) paradigm: *representation* of the problem within its domain, development of *methods* utilizing the representation to solve the problem, and *tools* to instantiate the method.

In order to build a tool to solve a problem, a representation of the problem must be developed along with a solution method to instantiate. The building of tools based on given problem representations and methods typically does not qualify as research unless it is the *first* tool to be developed, in which case the research question is feasibility: can the representation and method be instantiated into a viable tool?

Similarly, for building representations and methods, the research issues involve building *new* or *substantially* different representations and methods. Simply being 'different' or 'novel' may classify work as 'research' (depending on how novel it is), however, the burden is normally on the researcher to demonstrate that the new representation or method is "better" than existing ones. The evaluate and prove columns of the framework addresses this issue.

To adequately evaluate representations, methods, or tools, the researcher must develop measures of performance. These must address the key issues of the problem domain and the solution approach. The researcher then evaluates these measures for

various representations, methods, and tools to provide a performance comparison. This type of work is typically empirical. It develops case-by-case comparisons until the discipline has decided upon a standard set of measures.

Given a standard set of evaluation criteria, research can then proceed to prove the quality of representations, methods, and tools. Proofs may be in the form of 'optimality' of the solution, or superiority of the representation, method, or tool (where the evaluation measures define the optimization or comparison criteria).

If we apply the March framework and explanation to proposed new work in knowledge-based audit systems, they give us strong guidelines for differentiating research from development or empirical AI from applied AI. As he infers, building a new tool for a task is not really research unless the methods or representations change substantially or unless the researcher can demonstrate performance on well-developed evaluation metrics. For an audit researcher today, novel representations might include new structures (such as advanced forms of semantic networks (Winston, 1984)) and new problem-solving architectures (such as heuristic classification (Clancey, 1985)). Novel methods might encompass the use of new learning algorithms or the discovery of innovative knowledge-acquisition techniques. Moving across to the evaluate and prove columns would mean building new systems that are demonstrably better on tasks such as causal explanations or default reasoning. We might summarize this perspective on research project viability with a simple (somewhat frivolous) heuristic that refers to the columns of Figure 1: *The way to maximize your research might is to assess on all three and move to the right.* We refer to this as *R&D Heuristic #1*.

Domain Specificity and Maturity of the Research Field

According to the March framework, building a new tool with established representations and methods teeters on the research-development fence unless one is clearly the first person to do something in the area. We believe that judgments of novelty in this arena can be clarified by considering both the domain speci-

ficiency of the new effort and the maturity of the particular research field (or subfield) in which that effort's exposition is to take place. These considerations are discussed below.

Specificity and maturity considerations are illustrated with hierarchies in Figure 2, whose roots are very general and whose leaves are audit procedures specialized down to the task and firm level. As with all research, the more general one's conclusions are, the better; so staying up in the tree is desirable. In the three-dimensional plane of this figure, we have illustrated the age of the research subfield with the task classification of audit procedures at the start of the audit expert systems era (roughly 1983) being illustrated on the left and the hypothetical task classifications of future years being illustrated as going into the page on the right. Our point in accounting for time variability is that we believe that the proper set of research activities changes as a field matures. What is acceptable in an emerging area as exploratory research will often be deemed far less noteworthy as cumulative results dictate new directions.

For the proposer of a new audit expert system today, these time-sequenced hierarchies carry some important considerations. For instance, just finding an unexplored node and building a tool which uses established representations and methods is clearly not innovative enough unless the task is at a sufficiently high level of generality to warrant reassessment of the lessons learned from building entire classes of previous systems. In a like manner, exploratory programming of a new niche or subtree becomes less innovative as time goes on because March's research question of feasibility has been resolved. In both cases, the systems efforts being proposed would fall under the headings of development or applied AI. In the same spirit as heuristic #1, we can summarize this advice in *R&D Heuristic #2* (which refers to the tree-like structure of Figure 2): *As you climb down an older tree, you move away from 'R' and toward 'D.'*

Research and Development Delineation in Prototype Systems

A fully functional expert system involves considerably more development effort than

research effort, and designers will find that the new knowledge gained from building system components will decrease considerably as the project progresses. Actually, prototyping to a proof of feasibility is the essence of research in AI tools, a fact illustrated by McCarthy *et al.* (1989) in their task complexity hierarchy of Figure 3.

When a new AI system is proposed, assessing its ultimate feasibility involves the following:

- (1) Breaking the operation of the entire new system into its component procedures and arranging those components into a structured hierarchy like Figure 3;
- (2) Assessing the relative implementation difficulty of the top-level components and choosing the most complex module for further investigation;
- (3) Implementing a prototype of that chosen module down to its full depth of complexity; and
- (4) Assessing overall feasibility by combining estimates of both width and depth of effort from the preliminary structuring of the overall task and from the results of the prototyping efforts.

Empirical AI (research) would stop at this proof of feasibility unless there was clear evidence that further overall complexity (unrelated to individual module complexity/simplicity) might be introduced by full implementation factors such as scaling problems. Applied AI (development), on the other hand, would continue with implementation of the other components. Little new knowledge would be revealed by the development efforts, but the entire project would move closer to actual practical use in a cost-beneficial way. This delineation gives rise to *R&D Heuristic #3* (which again uses our tree metaphor): *Climb out on anything but your stoutest limb, and your prospects for research start to dim.*

Summary

In the introduction to this section we made the claim that empirical AI experimentation in accounting can rightly be viewed as legitimate research. In the subsections that followed, we provided several heuristics with which researchers can evaluate such endeavors. R&D

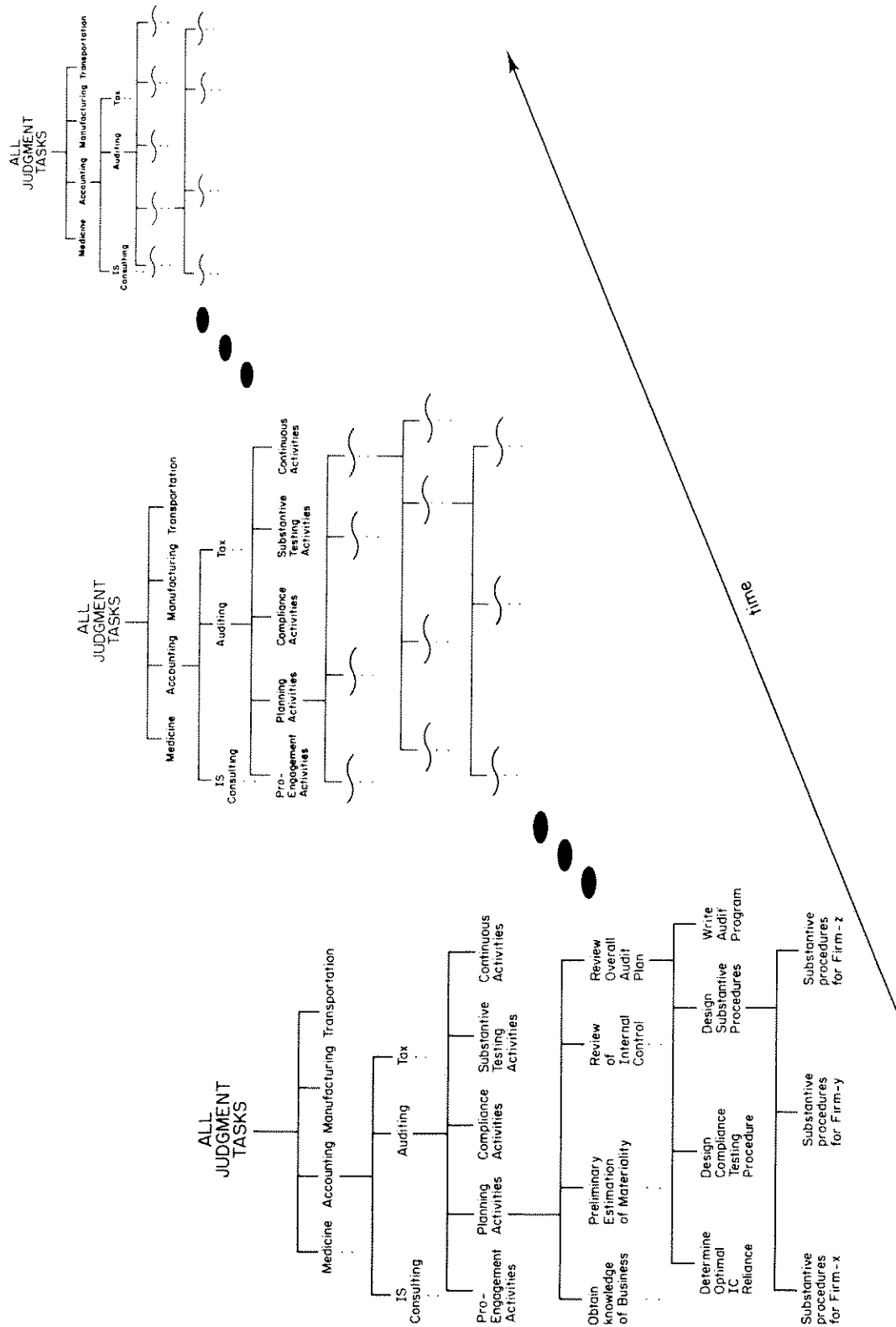


Figure 2 Domain specificity and maturity (adapted from Akresh et al., 1988).

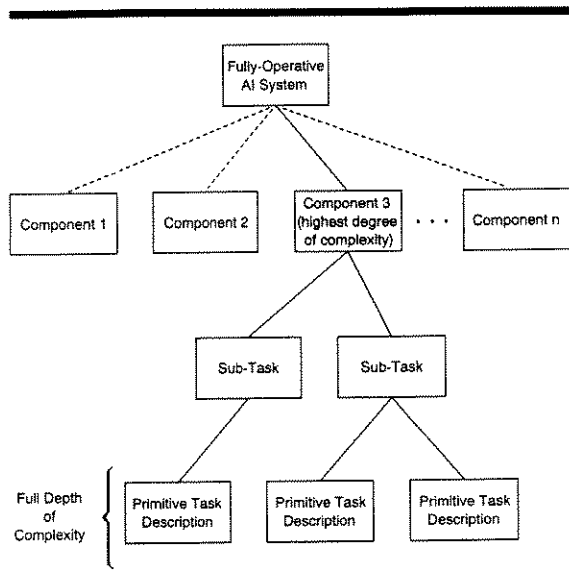


Figure 3 Prototype module structure (adapted from McCarthy *et al.*, 1989).

Heuristic #1 dealt with the engineering and methodological dimensions of information systems projects, R&D Heuristic #2 addressed the domain specificity and maturity of the research field, and R&D Heuristic #3 provided criteria for delineating the boundary between research and development in the use of prototype systems. The heuristics we supply are meant as easy-to-remember guidelines which can help determine the research contribution of software engineering projects.

We do not make the claim that software development is, of its own nature, always research—far from it. The principles contained in those heuristics set strict standards for both the selection of research projects and the manner in which these projects are implemented and evaluated. Simple heuristics rarely capture all the subtleties of the underlying knowledge they represent, but those we present here contain many of the essential principles to be followed if one hopes to produce quality empirical AI research. This is particularly true in the area of accounting research, where numerous knowledge-based software engineering projects have been undertaken, but where there have arisen no widely accepted standards for evaluation. The frameworks (and corresponding heuristics) we propose provide one

such set of standards—standards based upon the cumulative wisdom of several decades of work in this area.

Some Research/development Examples

In the previous two sections of the paper we have outlined in preliminary fashion some heuristic frameworks which can be used to assess the research content of a proposed AI-based audit tool. In this section, we will demonstrate use of those frameworks in exploring the time-lined development of four academic audit expert systems. We intend also to highlight their evaluation by contrasting their research content with three bogus expert systems. We have tentatively designated these bogus systems as YAK-BATs (Yet Another Knowledge-Based Auditing Tool), and they serve as prime straw men for our research/development differentiation arguments.

Our example audit systems are displayed in the Figure 4 box which portrays *empirical* AI systems as bubbling up and above the dotted line separating research and development and *applied* AI systems as gravitating down. The four real systems are AUDITOR (Dungan and Chandler, 1985), AUDIT-PLANNER (Steinbart, 1987), GC-X (Selfridge and Biggs, 1988) and IRE (Peters, 1989). The three bogus systems are YAK-BAT-1, YAK-BAT-2 and YAK-BAT-3; and we have positioned these straw men at particular time intervals purposely to highlight the types of proposed work properly classified as development. General features of each system are given in order below.

AUDITOR: This was a simple rule-based system that used a linear weighing method to assess the adequacy of a client's allowance for bad debts. It was the first publicized application of knowledge-based methods and representations to the domain of auditing, and it was certainly a pioneering research effort. The system was developed and validated with a set of working auditors.

YAKBAT-1: At the nascent stage of the audit expert system field in 1984-5, it would be hard to think of a proposed project which would not have shed some new light of knowledge on the area. However, if someone had proposed to

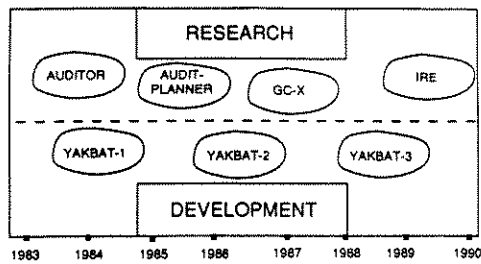


Figure 4 Academic systems: research and development.

use a known development shell on a fairly low-level task using well-understood methods of knowledge acquisition, we would consider that as sinking below the R&D surface. This would be especially true if there was no attempt made at emulation of an acknowledged expert and/or validation. In those cases, the developer would simply have been using the technology for automation of *ad hoc* decision-making heuristics.

AUDIT-PLANNER: This was a rule-based system with a much more complex control structure than AUDITOR. AUDIT-PLANNER was truly a cognitive model of one individual's expertise in the area of materiality judgments, so its research contribution is unquestioned. It was validated with subordinate auditors of the same firm. The representations and methods used in building the system were well known, but the task was fairly high on the domain hierarchy. **YAKBAT-2:** Steinbart's system circumscribed the entire materiality decision very well, and it was essentially self-contained in the sense that a consultation with AUDIT-PLANNER elicited a set of environmental cues from a user and used those cues in its goal of producing a materiality judgment. A tool developed later that would have concentrated heavily on the less complex development branches (such as tuning the user interface) or that would have used the same rule-oriented representations to emulate a lower-level audit task would fall into the development or applied AI compartment. **GC-X:** The Selfridge and Biggs going-concern expert system introduced the complex representations of semantic networks. They also demonstrated the complicated interactions between

audit task knowledge and client domain knowledge that had long been thought to be an important ingredient of audit expertise.

YAK-BAT-3: This might be a frame- or rule-based expert system which would lack the domain richness of GC-X. Certainly at this point in time, simple implementations of somewhat specific judgment tasks would lend little new insight to the field, unless the tool could be moved over to the evaluation or prove columns of the March framework.

IRE: The Inherent Risk Evaluator used complex representations of both firm-specific and general business knowledge along with specific predictions derived from analytical review rules to assess risk for audit planning. The system was validated carefully on three sets of case data, and its cognitive modeling intent is quite clear.

The research viability of each real system discussed above is widely (but not universally) acknowledged in the auditing community. Their developers undoubtedly would cast them first as cognitive models, but they all display innovation in a software development sense also. Certainly, researchers new to this field would be wise to concentrate on the more widely accepted behavioral science rationale in their development of proposed new projects. We remain convinced, however, that the technology-oriented rationale of the prior section constitutes an additional basis on which to plan new work, and perhaps more experienced researchers should venture out on this new turf. In the case of some research projects, such as REACH (McCarthy and Rockwell, 1989), we believe that sole reliance on the cognitive modeling rationale would lead to premature abandonment of demonstrably worthwhile ideas.

The Accounting Firm Perspective

As has been explained, research efforts concentrate on pursuing more accurate representations or models of cognitive processes while improving the methods for evaluating the representation methods themselves. Therefore, academic efforts focusing on the particular use of previously explored frameworks are best charac-

terized as development rather than research, given our discussion to this point.

Unlike academic researchers, professional firms tend to not be concerned about whether a particular project is characterized as research or development. Rather, firms focus primarily on enhancing the efficiency and effectiveness of audit practice instead of understanding low-level cognitive processes, exploring complex instances of judgment or developing formal methods of evaluating concepts. This interest typically results in accounting firms applying artificial intelligence technology along two fronts: (1) automating clerical or low-level audit judgment tasks and (2) leveraging firm or individual expertise.

Notwithstanding the profession's disinterest in distinguishing between research and development, these projects often result in significant contributions to academic research efforts. Graphically, the results of AI work among the firms might be characterized as shown in Figure 5. Although the major portion of a particular firm project will likely be characterized as development, a portion of the effort could legitimately be considered a research contribution.

The Contribution of Practice to AI Research

In a nutshell, we see practice efforts providing two contributions to AI research. First, the firms may propose and utilize novel methods and representations as well as providing well-developed metrics for evaluation of those methods and representations; and second, they expose weaknesses of scalability of academic theories and ideas.

During the past few years, a number of the firms have released various expert systems which are in use today. To a limited extent, some of these systems have provided a contribution to AI research by providing improved representation and evaluation methods. For illustrative purposes, we will briefly review the contribution of Coopers & Lybrand's new audit tool Risk AdvisorSM.

As Graham *et al.* (1990) explain:

Risk AdvisorSM is an expert system based on the knowledge and experience of senior audit and consulting professionals. It is used by

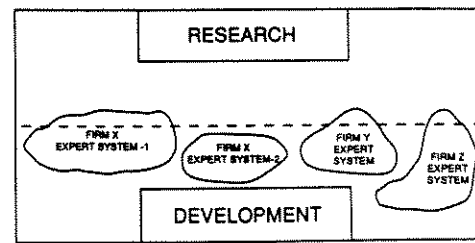


Figure 5 Accounting firm systems: research and development.

auditors to enhance the risk assessment process through the systematic capture and analysis of a wide range of financial information and other data to allow the timely identification of audit and business issues.

The system captures, analyzes and reports information ranging from standard client, industry, and economy-wide financial information to qualitative information captured through dialogue with the system. The system is utilized during audit planning to identify and document potential audit risks and management issues which are important to the audit. Additionally, it assists in analyzing whether appropriate action is taken in response to the issues raised by the system during the planning process.

Risk AdvisorSM certainly provides useful contributions to the issues of knowledge acquisition from more than one expert, knowledge representation and human/computer interaction. However, we believe the larger contribution of projects such as these lies more in their ability to address the 'toy world' problems which have plagued academic efforts for years.

Although execution of a computational model serves as a 'proof of concept or feasibility' which academics have used as their primary evaluation tool, the proof is still susceptible to weaknesses of scalability such as those often revealed by the overly simplistic application of exhaustive search methods. Upon being tested in realistic decision-support scenarios, the solutions offered by auditing academics sometimes prove insufficient for addressing problems in the real world. As Waterman (1986) states: 'When gross simplifying assumptions

are made about a complex problem, and its data, the resulting solution may not scale up to the point where it's applicable to the real problem' (p. 27). Projects such as those by Coopers & Lybrand certainly provide a test of the scalability of academic theories and thereby result in feedback to the academic community as to the adequacy of academic research.

Practice and Academics Working Together— The Optimal Solution

The primary contribution of academic researchers in any field is the low-cost application of analytical skills to problem solving. However, when academic efforts are isolated from the real-world problems faced by practitioners, the usefulness of the research wanes. Conversely, practitioners face real, complex and important problems daily which can prove costly if not carefully studied in a timely fashion.

The logical conclusion to an analysis of academic and practice efforts in the use of AI is that the two should work together. Such a consortium could likely result in significant enhancements to audit practice by providing sound solutions to real problems which have been carefully scrutinized without the pressure of the practice environment. To the extent that academics and practitioners can enhance audit practice while also increasing our understanding of audit judgment, significant contributions can be expected.

The reality of the situation, however, highlights significant challenges to developing working relationships between practitioners and academics. The strategic nature of AI projects tends to encourage confidentiality of project results at least in the short run while the firm realizes the rewards of being the 'first-mover' with a new idea. Such a practice is diametrically opposed to the nature of the academic environment, which attempts to distribute project results in a much more timely fashion in order to encourage additional research.

Although differences are obvious, they are no greater than those faced in many of the physical and engineering sciences in which universities and organizations work together on more sensitive issues of national security as

opposed to simpler marketing or operational advantages. We believe that any challenges can be overcome once practice and academe recognize the mutual benefit of working together.

SUMMARY

This paper has reviewed the progress of knowledge-based research projects in auditing, primarily in the academic section of the field. We outlined some heuristic rules and frameworks against which a proposed new audit tool could be evaluated and typed as either fundamental research or practical development. We readily admit that certain types of expert systems are like COBOL payroll programs in the sense that they are simple computational exercises that add little to fundamental knowledge. Building software systems which make *marginal* improvements with *known* approaches in *established* domains is definitely development activity, while building software systems which make *significant* improvements with *novel* approaches in *unexplored* domains is most certainly research. The difficulty lies in the middle, where we have concentrated our discussion. Academic researchers can follow our guidelines in trying to stay above the research/development surface.

We remain very optimistic that work in this particular field will continue to grow, along with knowledge-based research in other areas of accounting. Expertise in professional judgment will always be a scarce commodity on both public and corporate accounting staffs, and AI research methods continue to offer promising avenues for both explanation and leveraging of that expertise. The problems are interesting, the cognate field (AI) set of solutions and research methods continues to grow and the auditing practice imperative for efficiency and effectiveness remains high.

Acknowledgements

An earlier version of this paper was presented to the 1990 Deloitte & Touche/University of Kansas Auditing Symposium on 17 and 18 May 1990. Support in the development and

preparation of this paper was provided by the Department of Accounting at Michigan State University and Arthur Andersen & Co.

References

- Akresh, A.D., Loebbecke J.K. and Scott, W.R., (1988) 'Audit approaches and techniques', in Abdel-Khalik, A.R. and Solomon, I. (eds), *Research Opportunities in Auditing: The Second Decade*, American Accounting Association.
- Bailey, A.D., Hackenbrack, K., De, P. and Dillard, J. (1987) 'Artificial intelligence, cognitive science and computational modeling in auditing research: a research approach'. *Journal of Information Systems*, Spring, 20-40.
- Bailey, A.D., Jr, Graham, L.E. and Hansen, J.V. (1988) 'Technological development and EDP', in Abdel-Khalik, A.R. and Solomon, I. (eds), *Research Opportunities in Auditing: The Second Decade*, American Accounting Association.
- Clancey, W.J. (1985) 'Heuristic classification', *Artificial Intelligence*, 289-350.
- Cohen, P.R. and Howe, A.E. (1988) 'Towards AI research methodology: three case studies in evaluation', COINS Technical Report 88-31, Department of Computer and Information Science, University of Massachusetts, Amherst, MA.
- Dungan, C.W. and Chandler, J.S. (1985) 'Auditor: a microcomputer-based expert system to support auditors in the field', *Expert Systems*, October, 210-21.
- Ericsson, K. and Simon, H. (1980) 'Verbal reports as data', *Psychological Review*, 87, 215-51.
- Gal, G. and Steinbart, P. (1987) 'Artificial intelligence and research in accounting information systems: opportunities and issues', *The Journal of Information Systems*, Fall, 54-62.
- Graham, L.E., Damens, J. and Van Ness, G. (1990) 'Developing Risk AdvisorSM. An expert system for risk identification', presented at the USC Audit Symposium, Newport Beach, CA., February.
- Libby, R. (1981) *Accounting and Human Information Processing: Theory and Applications*, Englewood Cliffs, NJ: Prentice-Hall.
- March, S. (1988) 'Computer science research methods in information systems', presented at the International Conference on Information Systems, Minneapolis, MINN, December.
- McCarthy, W.E. and Rockwell, S.R. (1989) 'The integrated use of first order theories, reconstructive expertise, and implementation heuristics in an accounting information systems design tool', in *Proceedings of the Ninth International Workshop on Expert Systems and their Applications*, 537-48. Avignon, France: EC2.
- McCarthy, W.E., Rockwell, S.R. and Wallingford, E. (1989) 'Design, development, and deployment of expert systems within an operational accounting environment', *Proceedings of the Workshop on Innovative Applications of Computers in Accounting Education*, University of Lethbridge, Lethbridge, Alberta.
- Messier, W.F., and Hansen, J.V. (1987) 'Expert systems in auditing the state of the art', *Auditing: A Journal of Practice and Theory*, Fall, 94-105.
- Newell, A. and Simon, H.A. (1976) 'Computer science as empirical inquiry: symbols and search', *Communications of the ACM*, March, 113-26.
- Peters, J. (1989) *A knowledge based model of inherent audit risk assessment*, unpublished PhD dissertation, University of Pittsburgh.
- Selfridge, M. and Biggs, S.F. (1988) 'GCX, a computational model of the auditor's going-concern judgment', *Proceedings of the Audit Judgment Symposium*, University of Southern California, Los Angeles, CA.
- Steinbart, P.J. (1987) 'Materiality: a case study using expert systems', *The Accounting Review*, January, 97-116.
- Waterman, D.A. (1986) *A Guide to Expert Systems*, Reading, MA: Addison-Wesley.
- Winston, P.H. (1986) *Artificial Intelligence*, Reading, MA: Addison-Wesley.