Expert Systems as Decision Aids: Issues and Strategies

Ting-peng Liang

ABSTRACT

Although expert systems technology that takes advantage of artificial intelligence techniques is very powerful, its application in business domain is not without problems. This article examines issues involved in integrating expert systems and decision support systems and discusses strategies for using this technology. Five general guidelines for developing EDSS are presented. They are (1) selected applications, (2) realistic objectives, (3) validated knowledge, (4) evolutionary design, and (5) risk control.

INTRODUCTION

EXPERT systems (ES) designed to mimic and replace human experts have drawn considerable attention in the past several years. Although most of the early applications were developed in medical or engineering domains, business applications have become more and more popular [Blanning, 1984; Ernst & Ojha, 1986; Lin, 1986; Michaelsen & Michie, 1983]. Articles presenting existing prototypes have increased dramatically. Many potential benefits have been reported [Fried, 1987]. They include:

- Improved decision making,
- More consistent decision making,
- Reduced design or decision making time,
- Improved training,
- Operational cost saving,
- Better use of expert time,
- Improved products or service levels, and
- Rare or dispersed knowledge captured.

These potential benefits, coupled with research conducted in the decision support systems (DSS) area, have strongly encouraged an integration of ES and DSS technologies. For example, Scott Morton (1984) stated that "DSS as we know them

Ting-peng Liang is Assistant Professor, Department of Accountancy, University of Illinois at Urbana-Champaign, Champaign, Illinois 61820.

This research was partially supported by a grant from the Minnesota Mining and Manufacturing Company to the Department of Accountancy, University of Illinois at Urbana-Champaign.

may become obsolete in the foreseeable future. They are being supplanted by expert decision support systems—EDSS. The next generation of DSS will combine existing DSS technology with the capabilities of AI." Luconi, et al. (1986) argued that "for many of the problems of practical importance in business, we should focus our attention on designing systems that support expert users rather than on replacing them." Turban and Watkins (1986) discussed how to integrate ES programs into a DSS in order to create even more powerful and useful computer-based systems.

Developing EDSS that take advantage of both ES and DSS technologies is certainly promising. Its implementation, unfortunately, is not without problems. ES and DSS have different objectives, different design philosophies, and different architectures [Ford, 1985; Turban & Watkins, 1986]. These differences make this integration difficult. Furthermore, unlike engineering domains, behavioral considerations usually play an important role in the business arena. For a system that focuses on importing outside expertise, the risk of failure would be high. Therefore, before joining the bandwagon of using ES as decision aids, we need to carefully examine potential applications of this technology and to develop a framework that provides guidelines for employing various types of computer-based decision aids. In the remainder of this article, we shall discuss the issues involved in using ES as decision aids and develop strategies for using this technology.

ISSUES IN INTEGRATING ES AND DSS

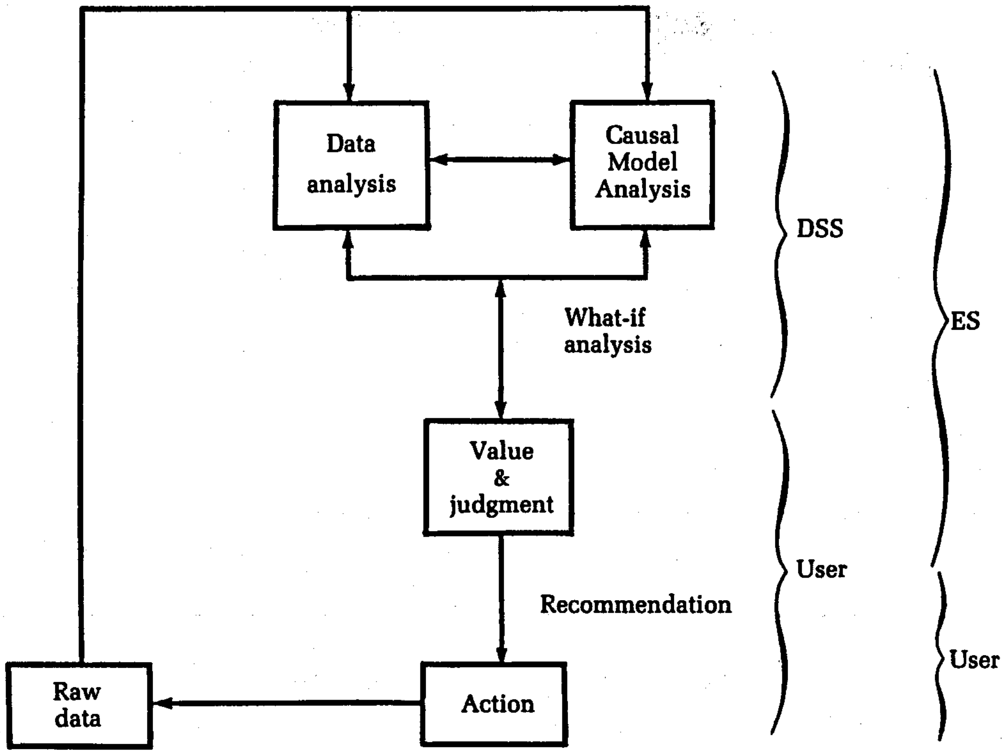
The basic premise of ES is that in some areas a small group of people (called experts) can perform a particular job significantly better than most of the rest. Since the knowledge (called expertise) of these people is rare and expensive, developing ES that capture and disseminate this expertise will be able to improve the decision performance of non-experts [Waterman, 1986]. The basic premise of DSS, however, is that for some semi-structured problems the decision maker can improve performance by conducting "what-if" type of analysis that takes advantage of the power of computers to speed up data analysis and mathematical calculation. Therefore, the integration of these two technologies have the following problems.

First, ES and DSS have different objectives. DSS focus on supporting decision makers in semi-structured or unstructured problems, whereas ES concentrate on replacing human decision makers in structured and narrow problem domains. This difference has resulted in two completely different design philosophies. In designing a DSS, the designer must always have the user in mind and adapt the system to meet user requirements [Keen & Scott Morton, 1978; Sprague & Carlson, 1982]. In designing an ES, however, the designer (or called knowledge engineer) must focus on acquiring knowledge from domain experts who are usually not the user of the system. In other words, the quality of knowledge is the primary concern, users are second. The designer of an integrated system must compromise these two philosophies.

Second, it is not clear whether the focus of integration should be the rule-based approach adopted by ES or the concept of including expert judgment in a system. ES and DSS have different functional capabilities. A typical DSS performs data analysis (called a data-oriented DSS) or model execution assistance (called a model-oriented DSS) for the user. The user is responsible for determining the data to be analyzed and the model to be used. A typical ES, however, further makes judgment based on its built-in knowledge and value systems. Figure 1 illustrates this difference. If an integrated EDSS only takes advantage of the rule-based techniques and still leaves the judgment to the user, then, just like rewrite a COBOL program in PASCAL, there will be no functional difference between EDSS and DSS. The resulting system will not have the anticipated power because it does not have the desired knowledge.

If an EDSS is designed to provide not only data analysis and model execution assistance but also its expert judgment, then the next issue is whose value and judgment functions should be coded into the system? From the DSS perspective, the

FIGURE 1



ES and DSS have different functional capabilities. DSS do not have knowledge to make judgments.

user's judgment function should be used. Since the user may not be an expert, this approach could result in a useless rule-based system. Even if the user is an expert, duplicating the expertise may provide little assistance. From the ES perspective, judgment functions elicited from a small group of selected experts are more appropriate. The problem with this approach is that it may generate high resistance—one of the major reasons for DSS to adopt user-oriented design.

Finally, even the designer successfully implement an EDSS that provides expert judgment, there are chances that in a given situation the EDSS and the user may draw conflicting conclusions. In this case, whose judgment should be adopted? How can we determine whose judgment is correct? Should we bring in another human expert or expert system to make recommendations? If the user's expertise has been proven better than the system's, then why should the user be bothered by the EDSS? If the system is proven better, then how can we allow the user to overwrite the system's judgment?

All these issues suggest that using ES as decision aids is not as simple or as excited as it seems to be. We need to know where it can be applied and how it can be used appropriately.

FIGURE 2

	TPS	DSS	ES	HE
• System-user interaction	Rare	User-directed	System-directed	B-directional
• Reasoning model	Quantitative & causal	Quantitative & causal	Qualitative & judgmental	Qualitative & judgmental
• System guidance in the decision process	Low	Medium	High	High
• System restriction	High	Medium	High	Low
• System customization	Low	High	Low	High
• Performance consistency	High	High	High	Medium
• Common sense reasoning	No	No	No	Yes
• Providing judgment	No	No	Yes	Yes

Transaction processing systems (TPS), decision support systems (DSS), expert systems (ES), and human experts (HE) are four types of decision aids. They are different in many aspects.

SELECTION OF DECISION AIDS

From a broad perspective, all systems, including human expert consultants, are decision aids, because nothing can replace the role of a decision maker who takes full responsibility for the outcome. Different types of decision aids have different characteristics. For example, a human expert has both common sense and professional knowledge in a particular area but is usually less consistent in performance. An ES provides a strong guidance in the decision process but has high restriction because it lacks common sense. A DSS provides customerized support to decision makers but cannot make its own judgment. Figure 2 shows a comparison of four types of decision aids: transaction processing systems (TPS), DSS, ES, and human experts (HE).

With these differences in mind, we must consider at least four factors to select and use a decision aid properly: the task, the nature of knowledge, the system, and the user. The first two factors determine what kind of decision aids is appropriate and the latter two factors determine the strategy for using a selected decision aid.

Selecting a Decision Aid

The first factor that affects decision aid selection is the nature of task. There are many ways to differentiate decision problems. Three of them are particularly important:

(1) availability of expertise,
(2) structuredness of the problem, and
(3) decision frequency.

If the expertise required for solving the problem is not available, then developing a good decision aid is impossible. If the required expertise exists, then we consider whether the problem is structured or unstructured and whether the decision occurs repetitively or only once. The problem structuredness affects the division of labor between the system and the user. In a semi-structured or unstructured decision making, only the structured portion can be automated because a computer system cannot process a job which human beings do not know how to do. The decision frequency is important in determining

FIGURE 3

Task / Knowledge	Structured		Unstructured	
	Repetitive	*Ad hoc*	Repetitive	*Ad hoc*
Qualitative reasoning	Expert systems	Human experts	Human experts	Human experts
Quantitative reasoning	Transaction processing systems	End-user computing	Decision support systems (institutional)	Decision support systems (*ad hoc*)

Selecting decision aids must consider the problem structuredness, decision frequency, and reasoning method. When qualitative reason is required, expert systems are appropriate for structured and repetitive decision and human experts must be hired for the rest. When quantitative reasoning is used, transaction processing systems are appropriate for structured and repetitive decisions, end-user computing is appropriate for structured and *ad hoc* decisions and decision support systems are appropriate for unstructured decisions.

whether a particular decision aid is cost-effective. For a decision that occurs only once, developing a sophisticated expert system may not be justifiable in terms of development time and costs.

The second factor to be considered is the nature of knowledge processed by the decision aid. It could be qualitative or quantitative. A qualitative reasoning process usually involves judgmental models, whereas a quantitative computation process uses causal models. Transaction processing systems (TPS) and traditional DSS focus on quantitative computation, whereas ES and human experts solve problems by qualitative reasoning.

Taking all these factors into consideration, we find that there is no decision aid that fits all cases. Figure 3 shows the situations where the following decision aids are applicable.

1. Expert systems

   In a structured domain where qualitative reasoning is crucial to problem solving and expertise is available, developing an ES (or EDSS) to support a repetitive decision in the domain may be appropriate. For example, loan evaluation is a repetitive decision for most banks. Except some special cases, the loan evaluation process and evaluation criteria are clearly defined. Therefore, an ES can reduce the workload of a loan officer and allow the officer to focus on special cases.

2. Human experts

   If the decision is structured but *ad hoc* or unstructured by nature, then the assistance an ES can provide is very limited. In this case, human experts must be hired if a support is desired.

3. Transaction processing systems

   If the desired support is quantitative by nature, and the decision is structured and repetitive, then a traditional transaction processing system that focuses on standard procedures and large amount of data will be sufficient. For example, providing monthly inventory report is a repetitive, structured and quantitative task, a good TPS will make this process much easier.

4. End-user computing

   When the decision is structured,

ad hoc and quantitative, one technology called end-user computing that encourages decision makers to develop their own *ad hoc* applications by taking advantage of user friendly fourth generation languages (4GLs) is very useful. The key in this case is to provide the user with a powerful 4GL with which an *ad hoc* application system can be built.

5. Decision support systems
   For an unstructured domain that needs quantitative support, DSS technology is appropriate. The system performs data analysis or executes proper models and the user makes judgments. If the decision is repetitive, then an institutional DSS may be developed. Otherwise, the user may develop an *ad hoc* DSS with a DSS generator and discard the system after successfully making the decision.

From this discussion, we find that ES can support only a small set of decisions. Furthermore, proper use of a particular technology may also be affected by characteristics of the system and the user. This is particularly true when ES are used. As discussed in the previous section, from the same set of facts, ES and the user may draw conflicting conclusions. Therefore, strategies for resolving the conflict are required.

Developing these strategies, we must consider the expertise of the user and the quality of the system. Users who use ES may have different levels of expertise varying from beginner to expert. The quality of ES may also vary from a rule-based toy to a real expert. There are many ES that do not demonstrate the desired expertise; but there are also systems that outperform human experts. For example, MYCIN, one of the earliest ES designed to diagnose infections and to recommend appropriate treatment, has been reported better than human physicians [Yu, et al., 1979]. In the experiment, MYCIN had a 65 percent success rate in prescribing correct medication, while physicians had an average success rate of 55.5 percent (ranging from 62.5 percent to 42.5 percent).

By comparing the quality of the system and the expertise of the user, four strategies for using ES technology can be developed: ignore, revise, follow and synthesize (Figure 4).

1. Ignore
   If only a toy ES is available and the user is also not an expert, then the contribution of the system is virtually none and it should not be used.

2. Revise
   If the system is a toy but the user is an expert, then the user may want to improve the system by revising its knowledge base. This strategy is appropriate only when the user has an intention to disseminate expertise. In other words, the enhanced system can be a good decision aid to other non-expert users. The resulting system may also work as a checklist for the user to avoid mistakes caused by ignorance in the decision process.

3. Follow
   The follow strategy applies when the user is not an expert but the system has a real expertise. In this case, the user must trust the system and take actions based on the expert system's recommendation. For example, when consulting with MYCIN, a patient should not overlook the system's prescription.

4. Synthesize
   When both the user and the ES are at the expert level, the best strategy is to find synergy. The ES must be treated as an independent consultant. The decision process will be similar to a group decision making process. Potential benefits in this case include: reducing obvious mistakes and expanding the scope of consideration by complementing with each other.

In summary, we have presented various strategies for selecting and using ES as decision aids in this section. To avoid misapplication of this powerful technology and to alleviate the problems addressed in the previous section, the following general guidelines must be followed: (1) focus on

FIGURE 4

		Quality of User	
		Non-expert	Expert
Quality of System	Toy	Ignore	Revise
	Expert	Follow	Synthesize

Quality of user and quality of system determine the strategy for using EDSS. If neither the user nor the system has adequate expertise, then the system must be ignored. IF the user is an expert but the system is not, then the user can revise the system to improve its knowledge base. If the system has expertise but the user is a beginner, then the user should follow the system's recommendation. If both are experts, then the best strategy is to synthesize two judgments to find synergy.

appropriate applications, (2) set up realistic objectives, (3) validate expert knowledge, (4) implement evolutionary design, and (5) control system risk.

GUIDELINES FOR DEVELOPING EDSS

1. Selected application

One of the obvious dangers involved in using EDSS is called the law of the hammer—give a child a hammer and he will use it on everything encountered [Hopple, 1986]. Therefore, to use ES technology constructively, we must carefully evaluate every application. We have known that an ES is appropriate only when the problem domain is structured, the decision is repetitive and the knowledge involves qualitative reasoning. In addition, there are several functional categories appropriate for this technology. These include interpretation, prediction, diagnosis, design, planning, monitoring, debugging, repair, instruction, and control [Hayes-Roth, et al., 1983]. As long as an application falls into one of these categories, ES may be considered.

To further evaluate an application, the following questions must be asked:

(1) Does the application have a clear boundary? Current ES technology does not allow the system to have much creativity. Therefore, unless the application needs only a finite set of known knowledge, the support an ES can provide will be limited. For example, tax advising is a bounded domain, but new product development is not.

(2) Does the application have standard cases from which knowledge can be derived and validated? If these cases do not exist, then knowledge acquisition will be very difficult and the resulting system may not be reliable.

(3) Is there any expert who can provide knowledge in the domain? The expert must have expertise and also have the willingness and time to cooperate with knowledge engineers in the knowledge acquisition process. If such an expert is not available, developing an ES for the application will not be possible.

(4) Is the size of the knowledge base reasonable? The complexity of the system is an exponential function of the size of the knowledge base. Therefore, developing a system that needs a huge amount of knowledge may be too costly and error-prone.

(5) Is a conventional system adequate for this application? Because ES technology is still in its infancy, using a conventional approach may solve the problem quickly and at a lower cost.

2. Realistic objective

If ES technology is found appropriate for an application, then a realistic objective for system development must be established. This can help us avoid the danger of omniscience that expects an ES to do something we don't know how to do. There are many unsolved (or unsolvable) problems in developing and using DSS. Unfortunately, using ES as a substitute is not the solution. ES are not super-DSS or super-humans. They are just other types of systems focusing on other types of problems. An ES cannot do anything that no one else knows how to do. In most domains, ES cannot perform even close to a real expert. Therefore, attention should be focused on strong economic benefits or knowledge dissemination, rather than unrealistic expectations.

3. Validated knowledge

Another important fact about ES is that the power of an ES is derived from the knowledge it possesses, not from the particular formalisms and inference schemes it employs. Therefore, thorough validation of the knowledge base is essential to the reliability of the system. The validation should start from the selection of experts and continue throughout the system development and utilization process.

(1) Before developing the system, qualified experts must be located. Those experts must have the expertise and also have time to work with knowledge engineers. They may not be the user of the system.

(2) Knowledge acquired from the experts must be validated before coding into the system. Standard cases may be used at this stage to find inconsistency, and indicate incomplete knowledge.

(3) A complete validation must be connected before applying the system to any real world problem.

(4) During system utilization, the knowledge base must be continuously revised to meet the changing environment.

If the system is purchased from a third party vendor rather than developed in house, then the system must be evaluated by a group of experts. In addition, it is important to make sure that the knowledge contained in the system can be either revised by the organization or updated by the vendor.

4. Evolutionary design

Since the user usually does not trust a decision aid until it shows reliable performance, an evolutionary approach that requires the designer first to develop a simple system and then to revise the system under the guidance of the user, has been a major approach for DSS design. In order to support the user with an ES, a similar approach must be adopted. This process will include three major steps.

First, when a system is developed or is purchased from a software vendor, the knowledge base already contains a set of basic knowledge. However, it may not have the specific knowledge that is useful only in that particular organization. Therefore, the system must be considered as a rule-based checklist, the user's judgment still plays a major role in the decision process. The user evaluates the reliability of the system and asks experts to revise the knowledge base if appropriate. The system at this stage may be called a rule-based DSS.

After the first stage, the user has found the strengths and offset the limitations of the system. The reliability of the system increases and the user starts trusting the system. In this case, the system makes judgments, but the user still keeps an eye on the system and overwrites the system's judgment. This system is called a human-aided ES.

Finally, the system becomes very reliable after a certain time period. At this time, the system makes most of the judgment and the user only focuses on special cases that cannot be handled by the system. If the system and the user draw conflicting conclusions for a particular problem, a careful examination of the conflict may be required. Unless there is a good reason, the user should avoid changing the system's recommendation.

This process allows a system to evolve from a rule-based DSS, human-aided ES to a valuable ES. It can reduce the possible resistance from the user and also gradually improve the reliability of the system.

5. Risk control

In addition to the technical issues, another important consideration is to control risks. Both financial and technological risks may occur if EDSS are used.

1. Financial risks

   Developing ES is very expensive and time consuming. A recent survey indicated that the average cost for developing a system was $700.00 per rule—excluding the costs of hardware, software tools, and the time experts contributed to the knowledge base [Fried, 1987]. Therefore, an ES project could be a financial disaster unless the management is fully aware of this fact.

2. Technological risks

   Because current ES technology is pretty young, it is very likely that a system developed today will be obsolete in a few years. In addition, it is sometimes difficult to know who is the real expert in a domain. Knowledge acquired from a non-expert may mislead the user. For example, some lawyers also provide tax advising service usually provided by accountants. It would be difficult to determine whether they are qualified experts. Finally, no reliable tool for knowledge acquisition is currently available. The development of ES is still more an art than a science. This may significantly restrict the reliability of the system.

CONCLUDING REMARKS

The term "expert system" has been controversial. On the one hand, it creates high expectation and has been used as a buzzword for funding and a flag to wave for all sort of projects [Bobrow, et al., 1986]. On the other hand, many people have criticized its feasibility. For example, Dreyfus and Dreyfus (1986) stated that "we believe that trying to capture more sophisticated skills within the realm of logic—skills involving not only calculation but also judgment—is a dangerously misguided effort and is ultimately doomed to failure."

In fact, ES are neither the solution to all problems, nor the solution to none. We need to understand where it can be applied and how to use it appropriately. This has been the main focus of this article. In summary, we have first examined the problems involved in using ES as decision aids. Then, strategies for using various types of decision aids have been addressed. Finally, five general guidelines for developing EDSS have been presented.

REFERENCES

Blanning, R. W. (1984), "Management Applications of Expert Systems," *Information & Management*, 7, pp. 311-316.

Bobrow, D. G., S. Mittal and M. J. Stefik (1986), "Expert Systems: Perils and Promise," *Communications of the ACM*, 29:9, pp. 880-894.

Dreyfus, H. L. and S. E. Dreyfus (1986), "Why Expert Systems Do Not Exhibit Expertise," *IEEE Expert*, 1:2, pp. 86-90.

Ernst, M. L. and H. Ojha (1986), "Business Applications of Artificial Intelligence Knowledge Based Expert Systems," *Future Generations Computer System*, 2, pp. 173-185.

Ford, F. N. (1985), "Decision Support Systems and Expert Systems: A Comparison," *Information & Management*, 8, pp. 21-26.

Fried, L. (1987), "The Dangers of Dabbling in Expert System," *Computerworld*, June 29, pp. 65-72.

Hayes-Roth, F., D. A. Waterman and D. B. Lenat (1983), *Building Expert Systems* (Reading, MA: Addison-Wesley).

Hopple, G. W. (1986), "Decision Aiding Dangers: The Law of the Hammer and Other Maxims," *IEEE Transactions on Systems, Man, and Cybernetics*, 16:6, pp. 834-843.

Keen, P. G. W. and M. S. Scott Morton (1978), *Decision Support Systems: An Organizational Perspective* (Reading, MA: Addison-Wesley).

Lin, E. (1986), "Expert Systems for Business Applications: Potential and Limitations," *Journal of Systems Management*, July, pp. 18-21.

Luconi, F. L., T. W. Malone and M. S. Scott Morton (1986), "Expert Systems: The Next Challenge for Managers," *Sloan Management Review*, Summer, pp. 3-14.

Michaelsen, R. H. and D. Michie (1983), "Expert Systems in Business," *Datamation*, November, pp. 240-246.

Scott Morton, M. S. (1984), "Expert Decision Support Systems," Paper Presented at the Special DSS Conference, Planning Executive Institute and Information Technology Institute, New York, NY, May 21-22, pp. 12.

Sprague, R. H., Jr. and E. D. Carlson (1982), *Building Effective Decision Support Systems* (Englewood Cliffs, NJ: Prentice-Hall).

Turban, E. and P. R. Watkins (1986), "Integrating Expert Systems and Decision Support Systems," *MIS Quarterly*, 10:2, pp. 121-136.

Waterman, D. A. (1986), *A Guide to Expert Systems* (Reading, MA: Addison-Wesley), pp. 24-31.

Yu, V. L., et al. (1979), "Antimicrobial Selection by Computers: A Blinded Evaluation of Infectious Disease Experts," *Journal of the American Medical Association*, 242:21, pp. 1279-1282.