Expert System Shells: Tools to Aid Human Performance

Jack R. Welsh  
Advanced Systems, Inc.  
155 E. Algonquin Road  
Arlington Heights, IL 60005  
and  
Brent G. Wilson  
Programming Instructional Technology  
GA 219-LEPS  
Northern Illinois University  
DeKalb, IL 60115

Abstract. The purpose of this article is to examine the role an expert system can play as an intelligent job aid. The use of expert system shells for microcomputers is discussed. We argue that instructional designers, with their competencies in consulting with experts and representing content, are an ideal constituency for learning how to develop expert systems within organizations. The paper concludes by presenting some issues that are important for instructional designers to keep in mind as they begin using expert systems to support job performance.

Expert systems technology is receiving a lot of attention in computing publications. Expert systems technology has grown out of research in artificial intelligence (AI), and is beginning to enter the mainstream product line for microcomputers. An expert system is a program containing a body of knowledge and a reasoning algorithm (often called an “inference engine”) for using that knowledge to make decisions and solve problems. The program is made accessible to a user through a computer system; in addition, the system can be linked to outside databases and programs.

Figure 1 briefly illustrates an expert system’s main components, and the IF-THEN rules that comprise the knowledge base. A small expert system might have anywhere from 50 to 500 rules like the one shown in the figure.

During a user consultation, the inference engine typically starts with a goal, then systematically searches through the knowledge base to determine an answer for the goal. The logic, known as backward chaining, is briefly demonstrated below.

Suppose the goal is: FIND the value of George. The system begins looking through rules, and finds this one related to its goal:

IF Stocks=Up THEN George=Happy.

Knowing the value of Stocks can now provide the value of George; henceforth, the system includes Stocks in its search for information about George. If it encounters a rule such as:

IF Day=Sunny THEN Stocks=Up

then Day also is included in the search. This process of backward chaining, working from a goal backwards until confirming or disconfirming evidence is found, continues until the rules in the knowledge base are exhausted. Eventually, the knowledge base (or the user during the consultation) may provide a value for a premise in the “chain.” Once a value is provided, the chain can progress through until the goal is obtained. Although this example is simplified, it is an accurate description of backward chaining. For a more thorough introduction into an expert system’s reasoning processes, see Harmon and King (1985). Such reasoning techniques are part of an expert system; other features are described later in the paper.

Being an “expert” in a subject is often (though not always) knowing when to perform which actions to solve a given problem. A computer-based expert system usually does not actually perform all the actions it recommends; deciding how to act is its main purpose. Thus, for content areas where action-taking and performance are important for expertise (e.g., playing a piano), expert systems are not much help. In content areas where decision-making is the primary concern, expert systems technology can be applied successfully.

We have used the term “expert system” in this article because it has taken on a generic meaning referring to any system that includes a knowledge base and an inference engine for using that knowledge base. We do not believe that expert systems can necessarily perform

An expert system is a program containing a body of knowledge and a reasoning algorithm for using that knowledge to make decisions and solve problems.
Figure 1: Main components of an expert system.

all the functions of a human "expert" in a given domain.

Within the last two years, expert system "shells" have become available for personal computers (Wilson & Welsh, 1986). An expert system shell is a program that provides a generic structure—a built-in inference engine and a system for editing a knowledge base—for use in developing customized expert systems. A useful parallel would be that of an authoring system for computer-based training development; a shell allows non-programmers to develop products efficiently and conveniently. LISP computer programmers and AI researchers would argue that the convenience comes at a price. Compared to a programming language, shells are inflexible and restricting. The "rest of us," however, feel fortunate that expert system development now lies within our power. What is lost in flexibility and power is gained in ease of use and efficiency. And just as authoring systems may differ in features and performance, there are differences among expert system shells in ease of use, power, and flexibility. (Information on a sampling of products using expert systems is provided in the Appendix; a more complete listing of shells is available in Wilson & Welsh, 1986.)

In effect, the marketing of expert system shells means that tools that were previously available in only a few AI research laboratories are now commercially available and will run on personal computers. These new tools allow computer-using educators to address many problems that were previously too difficult to tackle.

Microcomputer-based expert systems can be used in a number of ways. A product called Guru is available for the IBM PC that includes an expert system shell as part of an integrated word-processing/database/spreadsheet system. Another product, IMSATT, allows instructional designers to develop intelligent interactive video lessons. The authoring system includes an expert system shell to direct individualized prescriptions for instruction. Software tools will soon become available that include expert advisors built into the product just as help features are now built into products.

This last role of expert systems—as an advisor to product use—is an example of a job aid (Harmon, 1986a). Job aids can also stand alone, independent of other products. Stand-alone job aids are the most obvious immediate application of the expert system shells to training (Harmon, 1987). The use of expert system shells for the development of job aids is something instructional designers can learn to do without waiting for advanced technology.

The use of expert system shells for the development of job aids is something instructional designers can learn to do, without waiting for advanced technology.
all the functions of a human “expert” in a given domain.

Within the last two years, expert system “shells” have become available for personal computers (Wilson & Welsh, 1986). An expert system shell is a program that provides a generic structure—a built-in inference engine and a system for editing a knowledge base—for use in developing customized expert systems. A useful parallel would be that of an authoring system for computer-based training development; a shell allows non-programmers to develop products efficiently and conveniently. LISP computer programmers and AI researchers would argue that the convenience comes at a price. Compared to a programming language, shells are inflexible and restricting. The “rest of us,” however, feel fortunate that expert system development now lies within our power. What is lost in flexibility and power is gained in ease of use and efficiency. And just as authoring systems may differ in features and performance, there are differences among expert system shells in ease of use, power, and flexibility. (Information on a sampling of products using expert systems is provided in the Appendix; a more complete listing of shells is available in Wilson & Welsh, 1986.)

In effect, the marketing of expert system shells means that tools that were previously available in only a few AI research laboratories are now commercially available and will run on personal computers. These new tools allow computer-using educators to address many problems that were previously too difficult to tackle.

Microcomputer-based expert systems can be used in a number of ways. A product called Guru is available for the IBM PC that includes an expert system shell as part of an integrated word-processing/database/spreadsheet system. Another product, IMSATT, allows instructional designers to develop intelligent interactive video lessons. The authoring system includes an expert system shell to direct individualized prescriptions for instruction. Software tools will soon become available that include expert advisors built into the product just as help features are now built into products.

This last role of expert systems—as an advisor to product use—is an example of a job aid (Harmon, 1986a). Job aids can also stand alone, independent of other products. Stand-alone job aids are the most obvious immediate application of the expert system shells to training (Harmon, 1987). The use of expert system shells for the development of job aids is something instructional designers can learn to do without waiting for advanced technology. As more sophisticated expert systems tools become available for use on more powerful microcomputers, instructional designers will have learned basic expert systems principles on using currently available programs.

Traditional and Automated Job Aids Techniques

Techniques for building job aids are well established in the field of performance technology (Lineberry & Bullock, 1980; Harless, 1986). It is an axiom that often the best solution for a performance problem does not involve training. Performance can often be improved with the use of a job aid. One benefit of job aids is that they are less costly than the development and administration of a training program. Job aids can take many different forms. Some of the most

The use of expert system shells for the development of job aids is something instructional designers can learn to do, without waiting for advanced technology.
common are decision tables, information tables, flowcharts, and decision trees. What is essential is that the job aid must help the person perform a task correctly in the job environment.

Table 1 presents a decision table that is part of a job aid that helps computer programmers decide which Job Control Language (JCL) commands to include when defining a data set. The programmer approaches the task with information about the data set. The decision table is used to identify which JCL commands are needed. The availability of the job aid frees the novice JCL-user from having to memorize every combination of commands used for data sets. As procedures are repeated on the job, the programmer can gradually internalize the rules for solving the problem. Meanwhile, the job aid supports job performance.

Table 2 presents the same content through a sample dialogue with an expert system. The dialogue is based on a small knowledge base created using Insight II, but the substance of the dialogue is typical of many expert system shells. In this dialogue, a user tells the computer she is going to enter information on a case. The computer then asks a series of questions pertaining to the case. It processes the information and advises the user which JCL commands to code. In effect, the decision table is inside the computer rather than on paper. The computer guides the user through the decision making process.

Can expert systems do anything that can’t be done using decision tables or flowcharts? Actually, there are several advantages that expert systems provide over traditional devices for presenting job aids. One advantage is that more complex problems can be represented. Complex tasks simply mean more questions to answer. The user does not have to make sense of a complex figure, nor even know how to read a decision table or flowchart.

Another example, shown in Table 3, illustrates some additional features of an expert system. This dialogue recommends an appropriate statistical test for a given set of data. Because the system’s reasoning processes are quite flexible, the system does not necessarily fail when an item of information is missing. In this example, the system adapts to not knowing if the data were continuous or discrete. The system actually obtained that information in an indirect way. No flowchart or decision table could be so

<table>
<thead>
<tr>
<th>Is it an existing data set?</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the data set cataloged?</td>
<td>YES</td>
</tr>
<tr>
<td>Should this data set be put on tape?</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code the DDNAME</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code the DD operand</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Code the DSN parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Code the DISP parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Code the UNIT parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Code the VOL parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Code the LABEL parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code DSN parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code DISP parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code UNIT parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code LABEL parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code DCB parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Create and code SPACE parameter</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Table 3
Dialogue of Statistics Job Aid to Illustrate Confidence
StatAID
An Advisor for Choosing Statistical Tests

Computer: Which statistical test should apply to a data set? This program helps answer that question. If your problem lies within its expertise, it can help you choose an appropriate statistical test for data analysis.

User: I don’t know

Computer: Is your intended test a one-way or two-way classification?

User: One-way

Computer: Is your comparison – one mean with the population mean – two sample means – one pi – two pis

User: One mean with the population mean

Computer: Is the population variation known or unknown?

User: Unknown

Computer: What is the number of cases? Specify confidence level from 00–100

User: 27 cases. Confidence level is 80

Computer: Based on the information you provided, we recommend: A test for one mean using the t distribution

flexible to user needs. Note also that the user is free to ask for confidence levels on any data provided. For this problem, confidence levels may not be crucial, but many problems make good use of this feature.

Characteristics of Intelligent Job Aids

Like all job aids, an expert system advisor is meant to solve a particular performance problem. The people on the job are responsible for getting the job done—diagnosing a problem, deciding on a correct course of action, etc. To assist these people, an intelligent job aid is developed that serves as a guide and support to their own performance. Without the help of the job aid, performance suffers and costs are incurred in the organization.

What characteristics would a small expert system used as an intelligent job aid have? How would it differ from a traditional job aid? Several characteristics of intelligent job aids that go beyond traditional job aids are listed below.

1. Engage in a dialogue with the user. More of the burden for arriving at a solution can be placed on the job aid. For example, with a flowchart or a decision table, the user has to know how to read flowcharts or decision tables. There is a practical consideration of how much information a developer can put on a flowchart or a decision table and still have it usable. A typical, small expert system will have 80-100 rules for even a simple problem. With an intelligent job aid, more of the mechanical, repetitious burden can be placed on the machine. The user sees only the questions being asked, not the complex paths that are designed into the software.

2. Handle ambiguous situations. Many job aids place heavy emphasis on algorithmic rather than heuristic procedures. Decision tables and flowcharts, for example, require algorithmic approaches. However, many problems are best solved with heuristic rules. An intelligent job aid can capture competent performance that often involves ambiguous situations. Using confidence and threshold levels (for uncertain or missing data), and allowing information to be combined in various ways, small expert systems are suitable as intelligent job aids where other job aids would fail.

3. Make expert knowledge more generally available. Society has a variety of ways of trying to capture and preserve human expertise. It also has a long history of trying to find ways to make better use of that expertise. Books, for example, are one very successful attempt to capture and then distribute human expertise. One striking characteristic of computer programs is that they are easily duplicated and distributed. New information and rules can be added to an expert system easily. Thus an expert system’s performance can continue to improve. While significant effort is initially required to create an expert system, there can also be significant rewards for the effort invested. This has implications for society in general as well as firms competing in the marketplace.

4. Explain choices, decisions, or procedures involved in doing a task. Most expert systems can provide a path for reviewing the rules used in arriving at a decision. This feature allows developers to debug a system when they are building it. However, it is also a beneficial feature for users. It allows them to evaluate the plausibility of the program’s recommendations by examining the premises and connecting logic of the decision-making procedures used.

Conclusion

Taylor (1980) differentiated the roles of the computer in education as that of tutor (CAI), tutee (programming), and tool (e.g., word processing or database management). Until now, tools have been extremely useful but relatively "dumb." That is, they performed the repetitious processing of a task, leaving the decision-making and reasoning

JOURNAL OF INSTRUCTIONAL DEVELOPMENT
In effect, the marketing of expert system shells means that tools that were previously available in only a few AI research laboratories are now commercially available and will run on personal computers.

almost entirely to a human operator. Expert systems technology has the potential to merge Taylor’s three roles into a single, intelligent product. Computer tools can break new ground in helping people make hard decisions and perform complex tasks. It is entirely possible that expert AI applications will have a larger impact on education and training than our traditional models of computer-based training (Kearsley, 1987).

References

Appendix

Listed below is a sampling of products using expert systems technology that have some relevance for education.

Insight II Plus
Level Five Research
503 Fifth Ave.
Indialantic, FL 32903
(305)729-9046

Allows forward and backward chaining; interfaces with PASCAL, dBase, and Lotus files. $485.

VP-Expert
Paperback Software
2830 Ninth St.
Berkeley, CA 94710
(415)644-2116

Highly regarded shell that includes an inductive front end found in more expensive systems (similar to M1). Attractive user interface. $100.

ESP Advisor
Expert Systems International
1700 Walnut St.
Philadelphia PA 19103
(215)735-8510

A flexible but not too easy expert system shell. Interfaces to and looks a lot like Prolog. Good manual. $625.

More than 30 additional expert system shells are currently available (see Wilson & Welsh, 1986 for a more complete listing).

Author Note. We wish to thank Brad Waggoner and Boyd Nielson for their assistance in completing this article, as well as the JID reviewers who commented on an earlier draft of the manuscript.