Expert systems are moving out of research laboratories and into the real world. Forecasts of their rapid spread throughout business and industry abound. It is no surprise, therefore, that recommendations are being made that instructional developers use expert systems as intelligent job aids to replace or augment instruction. It is proposed that knowledge should be made available to performers on the job through computer-based advisors instead of off the job through training.

Nevertheless, it is easy to sense that the excitement about expert systems comes more from expectations about what they can do than from what they have already done. The technology is still new and changing rapidly. Except for a few large systems, the real “bottom-line” benefit of the technology, especially on microcomputers, has yet to be widely proven. Evidence even exists that many systems may not realize their full potential until additional advances in hardware and software design are achieved.

Is it really time, therefore, for instructional designers to become experts in expert systems? Should something so new and so untested really be the answer to their next performance problem? Of course, these questions require more than a simple yes or no answer. Like many training solutions, the appropriateness of expert systems depends upon numerous considerations, including the characteristics of the performance environment and the developers’ willingness to take risks, learn new skills, and face uncertainty.
The purpose of this article is to provide a framework for making choices about expert systems. It begins with a more elaborate description of the current status of expert system applications, looking briefly at both positive and negative points of view. A simple matrix is then presented which lists important issues instructional designers should consider regarding expert system solutions to performance problems. The article ends with several observations about the present and future value of expert systems to the instructional development community.

EXPERT SYSTEMS ARE HERE

Popular and professional journals alike contain very positive reports of the increased use of expert systems. For example, *Time* magazine (1988) concludes: "Eighteen months ago scarcely a handful of these systems existed in business and government. Now there are an estimated 1,000 to 3,000 in daily use and the number is increasing by 50% annually" (p. 60). To support this claim, *Time* identifies several large companies which have implemented custom-built systems in their organization. For example, IBM has 50 expert systems and plans to implement double that number each year. DuPont says it has 200 expert systems and plans to have over 1,500 before 1991. In the *Harvard Business Review*, Leonard-Barton and Sviokla (1988) conclude that expert systems are creating competitive advantage. They support their conclusion with numerous examples of large, successful expert systems built to solve a variety of internal performance problems, such as a system at Digital Equipment Corporation which helps solve problems in configuring newly purchased computers and a system at American Express which helps creditors make quick credit decisions by phone. Expert systems for mass marketing and distribution are also expected to increase, with sales forecasts growing from $325 million in 1988 to $2500 million in 1996 (Andren, 1987). Much of this commercial market will consist of expert system shells.

Shells are like authoring systems for creating computer-based instruction. They provide a structure for storing and delivering knowledge. Companies that buy expert system shells must build and insert their own knowledge bases, just as developers buying authoring systems must design their own instructional programs. Developers advocating expert systems have emphasized the importance of shells. Without shells, they suggest, the technology would be too complex. They contend that shells, especially those available on microcomputers, are a major breakthrough which makes expert systems accessible to persons not having artificial intelligence backgrounds or years of experience with traditional AI languages like PROLOG or LISP (Grabinger, 1988).

As use of expert systems increases, pressures for developer involvement will grow. Numerous authors already have recommended that expert systems be used as intelligent job aids (Welsh & Wilson, 1987; Harmon, 1987; Grabinger, 1988). Job aids have been described and advocated for years, primarily for situations which occur infrequently, where errors are costly, and where mistakes can be prevented with immediate access to important information. Welch and Wilson argue that expert systems go beyond traditional job aids because of the complex information they can handle. While decision tables become impossible to use when they contain more than a handful of rules, expert systems can easily manage hundreds of them while hiding their complexity (Welsh & Wilson, 1987).

An expert system derives its power from a cleverly devised relationship of its two major components: a knowledge base and inference engine. The knowledge base is an organized set of rules usually represented as IF-THEN propositions (IF patient has finding X and Y but not Z, THEN it is 75% probable the patient has disease A). The inference engine acts on the knowledge base and user-specified data to make decisions. The output of a system