Theory-Driven Expert Systems: The Next Step in Computer Software

Joseph M. Scandura

As the field matures, increasingly sophisticated uses of instructional software are inevitable. This software necessarily will draw on information gained from a variety of fields, one of which will be artificial intelligence (AI).

Application of AI to education, however, is not apt to be direct. One major limitation of artificial intelligence has been the lack of an adequate cognitive and/or instructional (i.e., interactional) base.

In view of our own research in the area, I would like to illustrate the above in the context of current work on expert systems and expert system generators. Put succinctly, expert (knowledge) systems are computer software systems that can perform tasks normally assigned to human experts or their supporters. Expert system generators are computer systems and/or programming aids which facilitate the generation of expert systems.

Artificial Intelligence and Programming Techniques

To date, programming techniques derived from artificial intelligence (AI) have been the primary driving force behind most contemporary developments in expert systems. Neither artificial intelligence per se, however, nor the closely related activity of "knowledge engineering," is likely to provide a sufficient basis for solving many of the questions raised in this area. Most contemporary expert systems lack a comprehensive and generalizable basis in modern cognitive theory. It is one thing, for example, to say that various types of knowledge representation may be useful for various purposes. It is quite another to know exactly when, how, and why. It would be even better to have a unified form of representation that would have the desirable qualities of each one individual—-and make it possible to represent both "deep" as well as "surface-knowledge."

Similar comments apply to goal- and data-driven inferencing or control. In this regard, references have been made in the expert systems literature (Kinnucan, 1984) to the purportedly "recent" development of inference engines, which selectively choose appropriate means of solving problems. While lacking full generality, the mechanisms proposed have more than a passing similarity to the human control mechanism postulated in structural learning theory and empirically verified years ago (e.g., Scandura, 1971, 1974, 1977, 1980, 1981).

An important problem with current expert systems, deriving from inadequate theoretical bases, is their relative inability to handle novel situations—although here, too, potential solutions have been available for some time in the non-AI literature.

An even more critical limitation is the lack of generalizability of many expert systems described in the literature (e.g., Kinnucan, 1984). While various techniques and tools have proven useful in developing a variety of expert systems, the expert systems themselves have not been easily generalizable to new problem domains, to more powerful inferencing, or to more complex systems generally.

Solutions to these problems will not come from technology per se. Without viable theory to drive development, solutions to such problems will be ad hoc, at best.

A Theory Base

Over the past 15-20 years at the University of Pennsylvania (e.g., Scandura, 1971, 1973, 1977, 1980, 1981, 1984), we have developed much of the theory necessary for generating expert systems and generators of the sort hinted at above—expert systems that would be based on: (1) a uniform, general, and highly modular form of representation which combines the useful features, for example, of productions and frames; (2) a control-mechanism that would incorporate both goal- and data-driven inferencing in a context-sensitive environment, as well as allow complete interchangeability of problem domains; and (3) similarly modifiable higher-order knowledge, making it possible to deal with novel problems. Currently available theory, most of whose basic assumptions have been empirically verified, also provides a basis for automatically generating expert systems as well as for interactive explanations and/or tutorials.

Implementation

During the past few years, I have been guiding the efforts of a small private company involved in implementation of various aspects of the structural learning theory. One product ("MicroTutor II


EDUCATIONAL TECHNOLOGY/November, 1984
Tutorial Arithmetic") deriving from these efforts is based on a general-purpose Rule-Tutor. The "Rule-Tutor" system was designed to take as input any problem (content) domain, represented in terms of rules, and automatically provide detailed interactive diagnostic testing and needed instruction.

More recently, we have begun a new, still more general effort using the IBM PC. Ultimately, this system will automate the process of constructing fully interactive expert and/or tutorial systems.

Toward this end, we have just completed the initial version of a general-purpose program generator. This generator is very user-friendly and already makes it possible to generate program code directly from flow diagrams, and to add newly created routines to its working library. Although it still lacks sophisticated input/output library routines, the generator already is proving extremely useful as THE primary tool in developing (the rest of) itself.

Once this phase of our development has been completed, we plan to move in two further directions. One will be to turn this program generator into a family of area-specific expert system generators—thereby making it possible for users (experts) to generate expert programs without even having to construct flow diagrams. It will be sufficient to merely solve representative problems in the domain interactively (with the system). The other direction will involve adding a general-purpose "inference engine" (as suggested above), which will allow the experts we create to solve novel problems (for which they have not been programmed). Ultimately, we also plan to build a general-purpose tutorial/explanation system, which in conjunction with specific experts, will make it possible to train people to become experts.

With regard to the above, I should emphasize that these extensions have been largely preplanned. They will NOT be ad hoc additions, but rather will follow directly from explicit designs worked out in accordance with empirically tested basic theory.

References


Scandura, J.M. Problem Solving in Schools and Beyond: Transitions from the Naive to the Neophyte to Master. Educational Psychologist, 1981, 16, 139-150.

Support for the research on which this article is based was provided by Scandura Training Systems, Inc.

What does ED TECH say?

Education Technology—the world's preeminent magazine of educational media and technology.

Forthcoming Articles

Following is a brief sampling of the scores of articles scheduled to appear soon in the pages of Educational Technology Magazine. If you wish to be sure to see these articles, consider taking out your own subscription to the magazine. Note: Subscription prices remain unchanged for 1985—and the rates are identical to those paid in 1979.

- An Interactive Research and Development Model for Educational Applications of Technology. By Paul R. Pintrich et al.
- The Role of Instructional Theories in the Evaluation of Computer Courseware. By Margaret E. Bell.