BRIDGING THE EXPERIENCE GAP

I/ExperTutor

by Suresh Bhavnani

Today's market presents users of computer-aided design (CAD) and mechanical computer-aided engineering (MCAE) software with an ever-changing, ever-advancing array of products from which to choose. The problem is that, as software packages become more advanced, they require more sophisticated users. Experienced engineers can use these tools effectively, but those who have not attained the necessary level of knowledge often encounter limits to their productivity. A new product from Intergraph bridges the gap between individual experience and software complexity.

Matching tools with experience

Using finite element analysis (FEA) software packages, engineers can analyze the structural behavior of a mechanical part in simulated real-world conditions. Due to their complexity, FEA packages often require users to have years of experience with FEA techniques. Herein lies the problem for less-experienced engineers.

There are several ways to compensate for lack of experience. Vendor-provided training helps. For example, a trainee learns the functions of each button on a menu. But training cannot teach how to make correct input decisions to an FEA system. To help in this process, engineering companies assign seasoned engineers – commonly called experts – to assist less-experienced colleagues.

Unfortunately, dependence on experts also affects productivity. Experts can spend an inordinate amount of time answering questions from entry-level engineers or occasional users, leaving themselves less time to address more complex matters. Resignations and retirements take their toll. Experienced professionals depart, taking invaluable knowledge with them and leaving novice engineers with no source of expert advice.

The Molex Experience

Molex Incorporated, Lisle, Illinois, is a Fortune 500 company that had worldwide sales of nearly $572 million in 1989. The company specializes in designing and manufacturing connectors for a wide range of products, including computers, telecommunications equipment, home entertainment equipment, and appliances.

Molex engineers use MCAE systems to conduct FEA during the design phase of the parts they manufacture. The company requires such structural analyses to be performed by designers, few of whom are trained in FEA. To compensate, Molex engineers from all over the world depend on advice from an in-house FEA expert, Ron Pentz.

With approximately 15 years of experience in FEA, Pentz recognizes the capabilities of expert systems in FEA applications. Pentz saw a demonstration of I/ExperTutor at the 1990 International Intergraph Graphics Users Group conference and recognized its potential.

Pentz agreed to share his problem-solving techniques with Intergraph's software developers. His cooperation was important because the success of an expert system depends on having access to an interested expert who is willing to share his knowledge. In mid-1990, Intergraph held a number of knowledge engineering sessions with Pentz at Molex. In addition to interviews, these sessions involved testing the knowledge base with actual Molex parts to see if the FEM knowledge base could provide accurate advice. The sessions revealed several areas where the knowledge base required improvement, and modifications were made accordingly.

After working with the software and recommending a number of changes, Pentz commented: "Intergraph's I/ExperTutor-FEM is a comprehensive system for expert consulting in finite element techniques. The knowledge base addresses issues in engineering mechanics and general finite element methods applicable to various industry applications. Employing I/ExperTutor-FEM, users can select finite element types with support comparable to five years' practical field experience."

Ron Pentz, a Molex in-house expert, gives FEA advice to less-experienced engineers who work for his company all over the world.

Expert systems: capturing knowledge

In recent years, expert systems have proven to be ideal vehicles for capturing knowledge and experience from human experts and making it readily accessible to novice users. Expert systems are designed to simulate the decision-making processes of experts.

It is instructive to compare the architecture of a database with that of an expert system. A database stores data in a data file, which is separate from the program that accesses it. Similarly, an expert system stores knowledge in a knowledge base, which is separate from its access program. The access program in an expert system is called an inference engine. Further, just as databases have different representations for storing data, expert systems also can store knowledge in different ways.

One of the most common means of storing knowledge in expert systems is the rule representation. This representation requires knowledge to be written in an if-then format called rules, which are stored in a file called a rule base. A rule base is just one type of knowledge base.

Expert systems share another similarity with databases. A database access program can operate on a number of data files. Similarly, because stored knowledge is separated from the inference engine, several different knowledge bases can be accessed by a single inference engine.

In a rule-based expert system, the inference engine frequently causes several rules to be chained or executed in sequence to reach conclusions that satisfy a goal *(see Backward Chaining, pages 28-29)*. At the end of a session with the expert system, a user can ask the system to provide an explanation or justification for a goal that was reached. In response, the system shows all the rules that were chained to reach the goal. Some expert systems even allow the user to modify original input to explore what-if scenarios. These features allow the user to understand the logic on which the conclusions are based. Only when the system's underlying logic is clear will the user rely on its conclusions and increase his own level of knowledge.

Knowledge engineering

It is widely accepted that the most difficult task in developing an expert system is capturing knowledge. Engineering experts use informal knowledge, such as rules of thumb, short cuts, and intuitive guesses, every day to solve problems. This knowledge, also called heuristic knowledge, is informal and undocumented, and therefore difficult to relay to the developer of the expert system. This time-consuming and difficult process of obtaining knowledge is called knowledge engineering.

Knowledge engineering is an iterative process designed to ensure the quality and accuracy of stored knowledge. Knowledge engineers must interview experts to collect knowledge and then encode that expertise into the knowledge base. The knowledge then goes through extensive testing for reliability. The collection, encoding, and testing processes continue until the expert endorses the accuracy and behavior of the expert system *(see The Molex Experience, page 27)*.

With all its capabilities, expert systems technology also has limitations, and it is important to understand them. Unlike human experts, expert systems do not learn new knowledge, although they might find paths that were not preprogrammed. These systems are most successful where knowledge is fairly stable and is retained by experts willing to share their expertise.

I/ExperTutor

Recognizing the need for expert systems, Intergraph will release its I/ExperTutor product and the I/ExperTutor-FEM knowledge base in 1Q91. Intergraph chose finite element modeling (FEM) as the first application area for expert systems because FEM knowledge is well-known by experts and is standard across many different industries. The I/ExperTutor environment is different from traditional computer-aided instruction (CAI) tools because it provides advice to users' specific problems, not just preselected problems. Furthermore, as noted by Jeff Delmas, manager of Intergraph's Finite Element Modeling product (I/FEM), "I/ExperTutor is really a learning tool. It does not solve problems for the user but rather guides him through the decision-making process to arrive at solutions."

I/ExperTutor-FEM builds proficiency in FEA by helping non-expert users to solve their own unique problems. The FEM knowledge base, executing within the I/ExperTutor environment, prompts the user with simple questions and illustrations, based on if-then rules in a

Backward Chaining

The rule representation is one means by which knowledge is stored in an expert system. Rules in a rule base are independent if-then statements describing conclusions that result, given certain conditions. For example, in FEM, rules can aid the engineer in selecting the correct inputs into an FEA software package. These inputs might relate to element shape or order.

To reach a goal, the inference engine searches through the rules in the rule base to find the value of an attribute. Frequently, however, the value cannot be obtained by analyzing a single rule. In such cases, the inference engine "chains" appropriate rules to define a path that will lead to the goal. One of the chaining methods commonly used in expert systems is called backward chaining. It is described here.

The following rule base contains three FEM rules that might aid an engineer:

RULE 1: If model description = frame structure

then element type = line

RULE 2: If element type = line
and loads = axial torque and moments

then element shape = beam

RULE 3: If element shape = beam
and solver = I/FEM

then element order = low

If the goal for this rule base were to find the value of element shape, the expert system would take the following steps:

STEP 1: The inference engine searches for a single rule that concludes the value of element shape. It finds Rule 2.

STEP 2: For Rule 2 to execute, the "if" part of the rule must be true. This means that the value for element type must be known. Since the inference engine does not know the value for element type, it must find another rule from which to conclude its value.

STEP 3: The inference engine searches to find a rule that determines element type, a subgoal. It finds Rule 1. Thus, the chaining process begins.

rule base. The user selects answers from a displayed set of illustrated choices. The questions are high level in nature and inquire about matters familiar to a novice engineer. When all questions are answered, the system displays its recommendations. If the user requests, the system explains how it arrived at the recommendations, thereby teaching him more about FEA.

I/ExperTutor-FEM has undergone extensive testing. Alpha tests took place in-house, and beta tests were conducted at six different industry sites. Engineering companies greeted the package with much enthusiasm. Several test sites recommended useful modifications to the FEM knowledge base. In assessing this new software, the comments of Joe Shoults, coordinator of MCAE applications, Rosemount Inc., Burnsville, Minnesota, are typical: "Currently, the biggest obstacle standing in the way of implementing FEM into our engineering groups is that the existing software packages are hard to learn and not very easy to use. By using a product such as I/ExperTutor, it would be possible for even the casual users to implement FEM into their design/ analysis process."

Intergraph will provide various knowledge bases, containing knowledge of other applications, which can be executed in the I/ExperTutor environment. Knowledge bases will be offered in areas where there is a critical gap in experience that leads to limits on the use of Intergraph products. Currently, Intergraph is developing a prototype knowledge base to aid in the productive use of Intergraph's Engineering Modeling System (I/EMS) for mechanical design applications.

The future

Expert systems undoubtedly have a bright future as electronic tutors for less-experienced engineers. Using expert systems, these engineers will easily and efficiently take advantage of collective knowledge. Bill McClure, vice president, Intergraph Mechanical Design, Engineering, and Manufacturing, predicts: "With the development of I/ExperTutor, expertise can be transferred to less-experienced engineers, greatly reducing their dependence on training and human experts." For both expert and novice, expert systems will complement and augment the human engineer's capabilities.

STEP 4: For Rule 1 to execute, the value for model description must be known. Since the inference engine does not know the value for model description, it must find another rule to conclude its value. The rule base contains no such rule.

STEP 5: As the inference engine cannot find a rule to conclude model description, it now has to ask the user for its value.

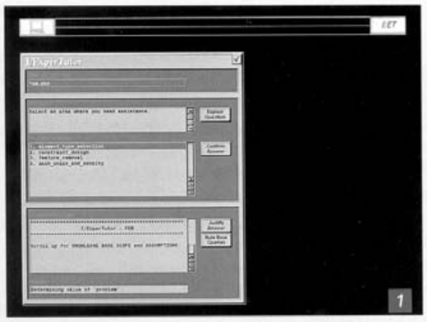
STEP 6: If the user enters the value of model description as frame structure, then the inference engine executes Rule 1. This results in the new fact that element type = line.

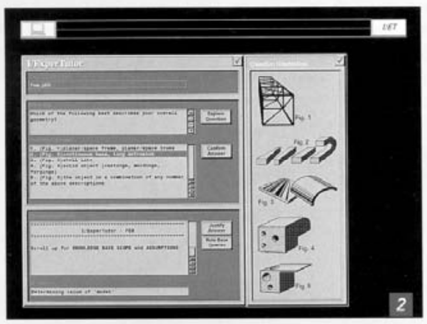
STEP 7: Now, as the subgoal has been satisfied, the inference engine remembers that it still has to find the value for element shape in Rule 2. Although it knows the value for element type, it now has to find the value for loads. There are no rules to conclude the value for loads, so the expert system asks the user for input.

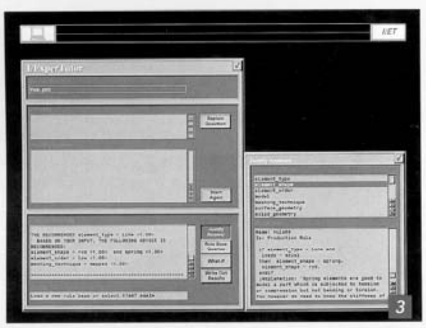
STEP 8: If the user enters the value of loads as axial torque and moments, then Rule 2 executes immediately. This results in the determination that element shape = beam, which satisfies the main goal, and the inferencing process ends.

These steps show how two rules were chained together to satisfy a goal. In the example, Rule 3 did not play a role because the knowledge it contained was irrelevant to any of the goals. If, however, the goal had been to find the value for element order, then the inference engine would begin by analyzing Rule 3 and chain backward through Rule 2 and Rule 1 to achieve the goal.

Although the rules in a rule base appear simple, the inference engine can trace complex paths through the process of chaining in its effort to satisfy a goal. The power of such a system lies in the fact that these paths do not have to be explicitly programmed.







I/ExperTutor in Action

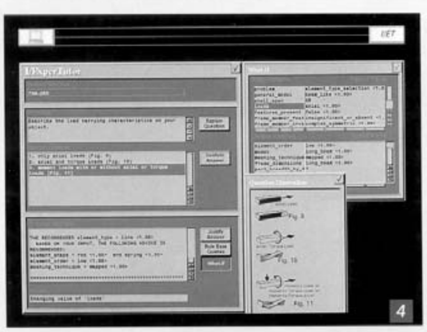
FIGURE 1 shows how I/ExperTutor appears after the FEM knowledge base has been loaded into the system. The user has chosen the "Element Type Selection" as the area in which assistance is needed.

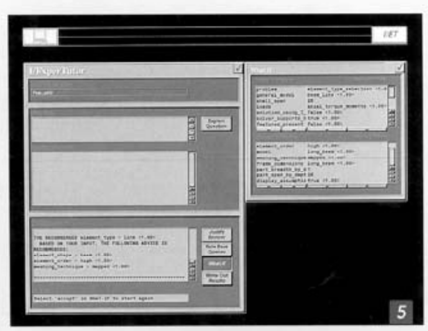
FIGURE 2 depicts the next stage, in which I/ExperTutor requires the user to describe the mechanical part under analysis. Onscreen illustrations provide a visual representation of possible choices (these pictures can be modified by the user to represent a company's specific parts).

FIGURE 3 demonstrates the stage reached after the user has answered all questions about his problem, as required by the system. For example, in the "Recommendations" field, I/ExperTutor has recommended for the analysis of the part that element type be lines and element shape be rods and springs. The user also has requested the justification of element type. The "Justification" field responds by displaying the final rule that executed to make a determination. A written explanation, explaining the logic of the rule, is also provided.

FIGURE 4 shows the system's what-if capability, where one of the inputs to the system is modified. In this example, the user is modifying the value of loads from "axial" to "axial, torque, and moments."

FIGURE 5 I/ExperTutor undoes all goals that are dependent on those changed and recomputes the values. ■





Suresh Bhavnani, a knowledge engineer with Intergraph Corporation, is product manager of I/ExperTutor.