Texpert

A tool for safety professionals & design engineers

By Gary L. Winn, B. Gopalakrishnan, Magdy Akladios, Paul E. Becker, Michael Carr, D. Courtney Black, Shekar Chittibabu, and Rajaarun Prasad Premkumar

A WEB-BASED, OBJECT-ORIENTED SOFTWARE has been developed as an expert system to help design engineers and safety managers work together to design safer products. Called Texpert, the software is in its second generation of development and has been tested and validated by designers and safety professionals to iron out wrinkles and speed up processing as it evolves. Even at this early stage, Texpert can identify design hazards that are often overlooked or only addressed at the end-user phase. The program employs artificial intelligence and familiar interfaces in a user-friendly, highly visual environment. Limitations are being worked out gradually, but validation feedback suggests that the final product should put expert safety advice and expert safety rules into the hands of engineers early in the design phase, and keep safety managers involved in every phase—from design to product prototyping, through testing and finally production.

As new technologies move from the bench-scale stage through pilot testing and into actual operation, little attention is paid to the hazards that they might pose to worker safety and health (NIEHS). These hazards can be eliminated or reduced by implementing controls. Every first-year student in an academic safety program soon learns that the most reliable and cost-effective way to control hazards is through the application of engineering controls to "design-out" the hazard. Addressing hazards in the design phase is preferable to second- and third-priority administrative and PPE controls because it mechanically and permanently removes a particulate, ventilation or mechanical hazard, for example, before an exposure occurs, or at least reduces that hazard to acceptable levels. To use another example, a carefully engineered system may fully isolate a vibration and noise from ever reaching workers at their stations.

Gary L. Winn, Ph.D., CHST, *is a professor of safety and environmental management at West Virginia University, where he is also coordinator of the occupational safety and health doctorate program. He teaches safety management principles, safety compliance and engineering. Winn is a professional member of ASSE's Northern West Virginia Chapter.*

Safety professionals know that an investment in engineering to control exposures and hazards, particularly at the design phase, is the best solution simply because it lasts longer and is not as easily defeated as administrative or PPE controls. Although the costs may seem high, the optimal time to eliminate or reduce the risk of a hazard is during the design phase. Research shows that the cost of hazard control during the design phase is far lower than implementing such control measures after the project is completed, and also lower than the less-permanent administrative or PPE controls. In fact, the costs to eliminate or reduce exposures increase as an engineering design project moves closer to production (Alexander 696+). Ferry found that for every $1 spent on safety at the design stage, $20 were saved during deployment (Ferry).

Identifying a hazard during the design phase can require expert knowledge and experience in the fields of safety, industrial hygiene, ergonomics and health. Unfortunately, even designers from the best engineering schools typically have little, if any, coursework in safety. It would make sense for a civil engineering student responsible for a million-dollar highway to know something about the design of work zones and their regulatory requirements in order to protect highway personnel. It would also make sense for a structural engineer to know that s/he could easily help eliminate fall hazards by designing anchorage points so that workers tie-off whether on roofing steel or walking across beams.

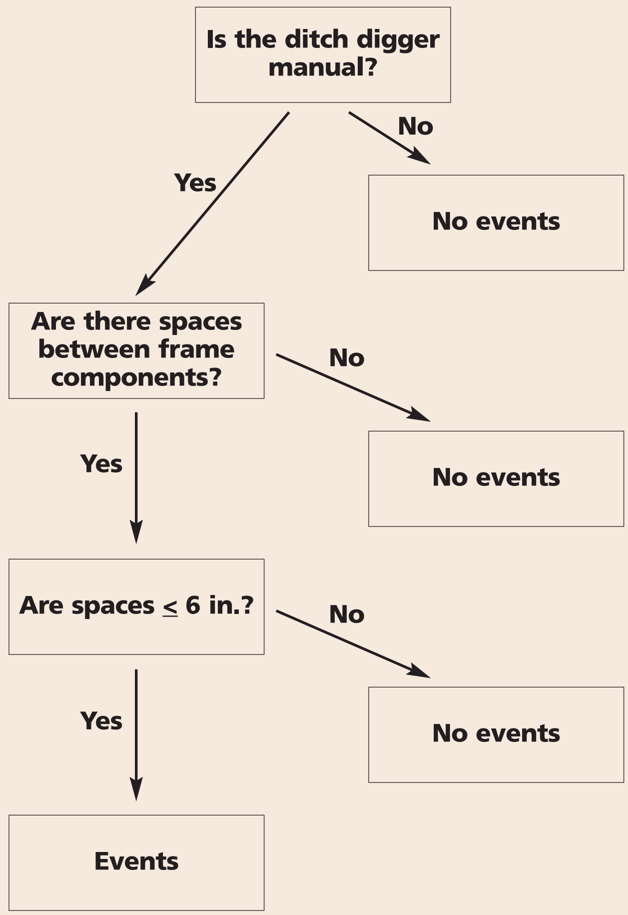
Safety design courses are simply not required for most engineers in training. Even the Accreditation Board for Engineering and Technology (ABET) does not require accredited engineering universities to offer a safety course for student engineers. At a time when some colleges are scaling back the required course hours for students, it is unlikely that design safety courses will be required in the future. How, then, are engineers to learn about design safety and how to incorporate cost-effective hazard controls through the best technology? How does a design engineer learn anything about OSHA, EPA or DOT regulations if these courses are not required in the undergraduate curriculum? If experience is unavailable or the design area has unfamiliar regulatory requirements, where can expert knowledge be found?

One way to efficiently link design engineering skills with expert knowledge in safety is through the use of expert systems built on a World Wide Web platform, and consisting of condensed expert knowledge of engineering design and relevant safety regulations and national consensus standards from groups such as ANSI or ASTM International. At West Virginia University (WVU), the authors have developed such an expert system for design engineers to have quick access to safety information and for safety managers to better understand the constraints and parameters of design. It is called Texpert, and it is now entering its third generation. To fully appreciate the utility of an expert system, one must understand the nature of such systems in general.

Computers have helped humans in many of the safety aspects of their lives. One way or another, we see that computer use is a major function in the field of safety and health. Early computer systems were used for calculation purposes such as RELVEC (Technical Research Center of Finland) aiding in fault tree calculation; and CRUNCH (Jagger), DENZ (Fryer and Kaiser), DEGADIS (Havens and Spices) and RISKIT (Kakko) for gas dispersion calculations. Computer use then moved from calculations to documentation, such as the use of CAFOS (Lihou) and SUPER-TREE (Knochenhauer). The third phase of computer-aided safety analysis was the development of programs aiding the diagnostic tasks in an analysis such as RIKKE (Haastrup, et al) and CAFTS (Poucet and Peterson) developed for fault tree construction from pipe and instrumentation diagrams (Suokas, et al).

Figure 1

Partial Decision Tree for the Frame Component



Expert Systems

By definition, an expert system is "a computer-based reasoning system that captures and replicates the problem-solving abilities of human experts. Expertise consists of accumulated knowledge about a particular domain of interest, an understanding of domain problems and skill at solving some of these problems" (Laurig and Rombach).

An expert system is a computer program that contains a knowledge base and a set of algorithms or rules that infer new facts from knowledge and incoming data. The correctly solved problem is based on the quality of the data and rules obtained from the human expert and how efficiently the rules can be extracted. While expert systems are designed to perform at a human expert level, in practice they will perform both well below and well above that level. The expert system derives its answers by running the knowledge base through an inference engine—a software program that interacts with the user—and processes the results from the rules and data in the knowledge base.

Expert systems usually contain two main components: a knowledge base and an inference engine program, enabling it to suggest conclusions. The knowledge base is programmed in an "if/then" logical rules structure. Such a structure is a series of *if* conditions that if met, *then* a specific result may be concluded. Following this model, an expert system will receive propositions to a certain line of questions, then use its inference engine to process the information into simple conclusions. It will compare the propositions to the facts and rules registered in its knowledge base. The knowledge base is where the knowledge rules of one or more human experts in a specific field or task is stored. It is an "intelligent" database, in that it can usually manipulate the stored information in a logical, natural or easy-to-find manner. It can conduct searches based on predetermined rules of defined associations and

Figure 2

Rules & Events Generated from the Decision Tree

If	Component selected is frames Mode of operation is manual Frame_space ≤ 6 in.
Then	**Event** Operator caught in frame leading to cutting, shearing or severing injuries Injuries due to loss of operator control
	Recommendation Design frame to have larger spaces between frame parts (>18 in.).
	Standard Operating Procedures Keep all body parts away from cutter bar during operation. Do not wear loose clothing while operating machine. Keep hands on handles during operation to maintain control.

relationships, as well as more-traditional data-search techniques.

Expert systems are of particular benefit to industry and science. They are widely used in medicine, oil exploration, weather forecasting, stock market predictions, financial credit, fault analysis and mine safety. These programs take the knowledge of one or more human experts in a field and computerize it so that it is readily available for use. Those human experts need not be physically present to accomplish a specialized project or task. Expert systems are only designed to be "expert" in a narrow and specific task or subject field. An expert system, then, represents multiple human experts and tries to imitate the experts' evaluation processes to offer a conclusion. An advantage of an expert system is that it may include the knowledge of many experts in one specific field, and do it rapidly and reliably.

A second-order advantage is use by the safety manager who may not also be an engineer. In the same way a designer is exposed to safety's expert rules and regulations, the safety manager is exposed to the steps a design engineer takes from conception to subassemblies to prototyping to production, and can work with the design engineer doing concurrent (in parallel) engineering rather than "over the wall" (independent or serial) design safety work.

The knowledge base of an expert system is made up of factual knowledge and sometimes heuristic knowledge. Factual knowledge consists of information that is commonly shared, usually found in textbooks or journals, and typically agreed upon by experts in that field. Heuristic knowledge, on the other hand, is experimental knowledge of performance; it is the knowledge foundation behind the expert system's "educated guess."

Humans use both types of knowledge every day to make judgments about new situations. Expert systems do the same, only faster. The inference engine of an expert system is usually set up to mimic the reasoning or problem-solving steps that a human would use to arrive at a conclusion. It simulates the evaluation process of relating the information and rules in the knowledge base to the answers given by the operator to a series of questions. The typical expert system will ask related questions with greater detail until it has collected enough information to suggest possible logical conclusions.

Selecting the proper component for a computer-based occupational safety and health information system—or, in this case, an expert system—requires two major classes of action: defining the problem and choosing appropriate tools to solve the problem (McNichols).

Texpert, which stands for "technical expert system," is a demonstration software developed to provide a useful, working link between the design engineer and "warehoused" expert safety knowledge. From its inception, the application was designed to operate in a web-based environment. The system's first generation architecture consisted of an HTML-based user interface connected to an expert system sever called Resolver, a common software from MultiLogic Corp.

Start of the System: Designing a Ditch Digger

Texpert's development team was composed of an expert group and a computer group. The expert group included industrial hygienists, ergonomists, engineers and safety professionals. This group was charged with developing a list of questions or data requirements needed to evaluate a ditch digger, determining the risk events likely to occur and developing the necessary recommendations or procedures to eliminate or reduce the risk of an event. All substitution and engineering controls were grouped under "recommendations" to eliminate the risk of an "event." The system was also designed to provide the user with standard operating procedures (SOP) that include all other types of controls such as administrative measures and the use of personal protective equipment (PPE). SOPs are provided as a bonus to designers, in case they decide against given recommendations.

The computer group was in charge of programming the expert system and giving it life on the web. This involved coding the knowledge base of the expert shell, developing necessary screens for user interface and data exchange using HTML, deployment of the knowledge base on a web server, and providing tools such as links, hypertext explanations and risk calculation.

The first task in developing the expert system was determining the level at which the evaluation would be performed. The entire machine, a common ditch digger, could be evaluated as a whole entity, a method that would provide an easy-to-understand point of view for the user. However, this approach would not facilitate efforts to evaluate potentially hazardous conditions that the team wanted the design engineer and safety manager to recognize in the machine. Another option would be to evaluate the system at the "nuts and bolts" level. This concept was quickly discarded since risk events usually do not occur at this level. The expert team settled on a "component" approach, defined as a part of the ditch digger that could stand alone or could be purchased as a unit. To begin the development of Texpert as a real expert system, the team decided to conceptually break the ditch digger into six components: frames, wheels, handles, energy source, cutting mechanism and controls/displays.

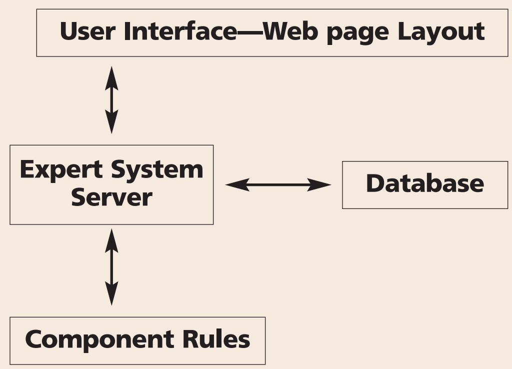
An expert system must reside in a shell that provides an environment for rules to be entered and executed. Shell selection depends on the task to be performed. The team's choice, MutliLogic's Resolver, is a knowledge-based system development tool that combines a rule editor with a flexible visual decision-tree interface and an inference engine. The shell can run under many operating systems and, with the help of a web runtime engine (NetRunner), can be run in a web environment. In addition, many designers who are familiar with Microsoft products are also familiar with Resolver. To develop a set of rules for each component, the expert team developed decision trees of questions. These questions led to either an event or a no-event situation. An event would be the possibility of a worker injury or illness. Figure 1 shows an example of a decision tree.

When a probable event was triggered, the corresponding advice-set was triggered—including standards, regulations and specifications—and design changes were provided for a less-hazardous product. SOPs were also included to provide operating instruction. The decision trees were then written as if/then rule sets. Figure 2 shows an example of the rules associated with the decision tree in Figure 1. Though simple, the illustrated rule is a useful example; it involves the operator getting caught between the frame and boom of the ditch digger, posing a risk of injury. Many other rules involved were more complex or required quick database searches for technical advice.

Validation of the rules for phase one involved a three-step process. First, experts examined the rules to determine their appropriateness and whether the values were correct. For example, in the frame-spacing rule shown in Figure 2, validation involved examining anthropometrics tables. It was determined that six inches was too small, since the 95th-percentile North American male has a hand length of 8.07 in. (Pheasant). The rule was written to reflect this information.

Figure 3

First Generation TEXPERT Architecture



The final step in validation involved human technology designers. Three designers with various backgrounds, including safety management, were asked to run the program independently and provide verbal and written feedback on the rules, the system and the information generated from the program; nonexpert users were also asked to assess the system. Comments from both groups were used to further improve the program. Comments involved question clarity, web page design and general usability factors. Figures 1 and 2 show how Texpert outputs could identify design hazards that may otherwise be transparent to the design engineer.

Output from the expert system was stored in Microsoft SQL Server. SQL, a universal query language, is used to draw data from the database.

Although effective, this initial architecture was not ideal. The system chosen had rudimentary data access tools. For every "read" or "write" to a database record, a database connection and user login had to be performed. Then data could be written to or obtained from the database. After the read/write was performed, a logout was initiated. Although the login/logout procedure was hidden from the user, the operation took longer to perform than database reading/writing, which proved to slow the system's web operating speed.

To overcome this issue, the team developed a new architecture that would maintain a single login to the database during the full length of time the user was logged into the system. To accomplish this, the team developed a web application using Java script and VB script to manage all of the Texpert database functions. Java script is used for client-side scripting, while VB script is used for server-side scripting. Texpert uses Java script for validating forms before they are submitted. It uses VB script for

Figure 4

Second Generation TEXPERT Architecture

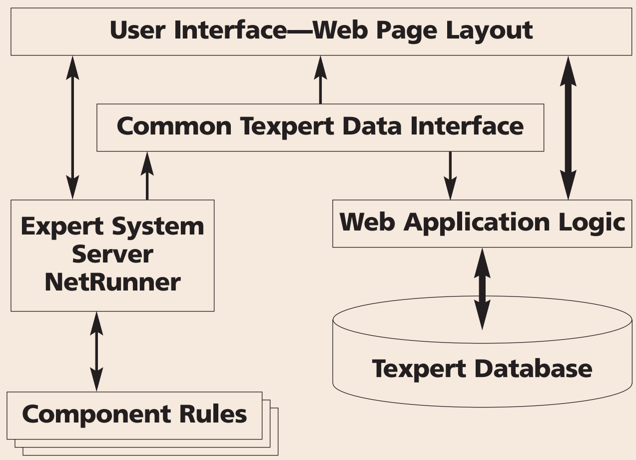
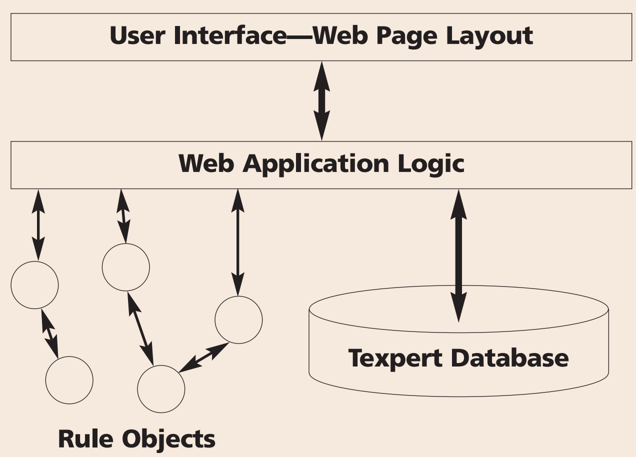


Figure 5

Proposed Third Generation TEXPERT Architecture



controlling the logistics of using the system and updating the database with user, component, evaluation and project management information. The web application eliminated the need for the expert system server to access the database, but required the development of a new method to obtain the results of an end-user session with the expert system. To communicate with the expert system, a component called the "Common Texpert Data Interface" (CTDI) was developed. This interface is the main form of communication between the expert system component modules and the Texpert web application. The CTDI reads an HTML output from the expert system, identifies and converts the input HTML into a standard output that is then sent to the web application. Output from the interface component is then managed and written to the database (Figure 4). The result is a much faster process between inquiry and conclusion.

The second generation architecture has allowed the development team to demonstrate and validate the concept and methodologies to potential end users. As the number of designed components increases, the number of rules and duplicate rules also grows, which makes maintainability of the system under the current architecture more difficult. To enhance the scalability of the system and speed the rule development process, the group has proposed to replace the expert system shell with a rule-based, object-oriented component model (Figure 5). Objects are instances of a particular subcomponent. For example, an electrical subcomponent object has properties such as current type, voltage and amperage. Components that have an electrical subcomponent, such as a control panel, would inherit all the rules and properties of the electrical subcomponent. The electrical rules would be written once and used by any component that uses electricity. In an object-oriented, component-based model, rules will be broken down into separate rule sets that are stand-alone in nature. A component-based architecture allows developed rule sets to become reusable objects within the system. These objects can be pieced together like a jigsaw puzzle to form new end-user components. The reusability of component modules decreases development time, reduces bugs, improves maintainability, and facilitates system growth and modification over the life cycle of the application; it also increases system speed.

The team learned an interesting lesson from working with designers: designers and safety experts alike describe their technology in a visual manner. To partially solve that problem in Texpert, a user's project is constructed by simply selecting graphical components from a list of those available and adding them to the project list. The team developed a prototype visual interface that allows the designer to choose components from a palette and place them on the page. The selected components can then be connected appropriately with lines via a point-and-click interface. A visual project builder will facilitate data entry for the designer. This, in turn, will increase utilization of the expert system by designers. The added information of component relationships will allow a more thorough review of the designer's project, facilitating better recommendations, which will lead to safer designs.

Generating a TSDS

In the validation phase, it was suggested that the team produce a single sheet or window with a succinct summary of design hazards and corresponding technical data and/or regulatory links to speed the extraction of expert data. The system is now configured to generate a technical safety data sheet (TSDS), similar to a job safety analysis that is familiar to safety managers. The TSDS is a "tear-off" printout the designer can easily export or place into another hard file. It lists all major design hazards in one column (e.g., fall hazards), links them with suggested corrective action (e.g., design anchorages with specifications) and generates regulatory citations (e.g., 1926 Subpart M: anchorages, fall protection). At this stage of Texpert, the TSDS is the end product.

System Limitations

The system has several limitations. First, the design team recognizes that no computer system will replace a human expert, since a human can react to unique situations better and see a larger picture than the computer system. The team is concerned that Texpert might be used as the only evaluation and not simply as a tool to aid designers. The second limitation involves the issue of length versus detail. To be useful, the software must have detailed information about many aspects of each component. If a user is entering a large technology with many components, processing the information can take a long time. If too long, the user may ignore Texpert altogether. However, without sufficient details, the information gained from the system would be so general that it would be of little use. The design team is continually addressing this issue, adding speed, simplicity and a more "visual" character.

Conclusion

Texpert is a work in progress; further refinements are planned before the product can be used with confidence on any scale. However, the program is available online for review and testing at www.computercomp.com/texpert.

The design team wants a group of acknowledged designers to put Texpert to the toughest test to identify weaknesses. Further, it wants a highly skilled group of safety experts to do the same. Testing the software across a wide range of professional experience will help improve the product; these tests are currently scheduled. In the end, Texpert will represent a high-water mark for artificial intelligence as a tool for design engineers working with safety managers. As with anything involving computers, no magic is involved. The quality of an expert system depends on the quality of the knowledge base used as input. While some rules are derived from rather abstruse experience that only can be obtained by "being there" or "doing that," one must always remember that it is human intelligence which must interpret the expert system's conclusion and decide whether it is true or even feasible. As all computer users know, expert systems are not infallible. ■

The goal is to put expert safety advice and expert safety rules into the hands of engineers early in the design phase, and keep safety managers involved in every phase—from design to product prototyping, through testing and finally production.

References

Akladios, M. and B. Gopalakrishnan. "Development of an Expert System to Help Design for Worker Safety." *Robotics and Machine Perception.* International Technical Working Group Newsletter. Bellingham, WA: Society of Photo-Optical Instrumentation Engineers, August 1998.

Alexander, D.C. "The Economics of Ergonomics." In *Proceedings of the Human Factors and Ergonomics Society (HFES) 38th Annual Meeting.* Santa Monica, CA: HFES, 1994. 696-700.

Ferry, T. *Safety and Management Planning.* New York: Van Nostrand Reinhold, 1990.

Fryer, L.S. and G.D. Kaiser. "DENZ: A Computer Program for the Calculation of the Dispersion of Dense Toxic or Explosive Gases in the Atmosphere." SRD R 152. Warrington, UK: UKAEA, Safety and Reliability Directorate, 1979.

Haastrup, P., et al. *RIKKE Users Manual.* Riso-M-2480. Roskilde, Denmark: Riso National Laboratory, 1985.

Havens, J. and T. Spices. "Development of an Atmospheric Dispersion Model for Heavier-Than-Air Gas Mixtures." Report No. CG-D-22, 23 and 24-85. Vols. I and II. Washington, DC: U.S. Dept. of Transportation, U.S. Coast Guard, 1985.

Jagger, S.F. "Development of CRUNCH: A Dispersion Model for Continuous Releases of a Denser-Than-Air Vapor Into Atmosphere." SRD R 229. Warrington, UK: UKAEA, Safety and Reliability Directorate, 1983.

Kakko, R. "Vapor Cloud Modeling in the Assessment of Major Toxic Hazards." *Journal of Loss Prevention Process Industry.* 2(1989): 102-107.

Knochenhauer, M. "ABB Atom's SUPER NET Program Package for Reliability and Risk Analysis." Presented at Symposium on Advanced Information Tools for Safety and Reliability Analysis. ISPRA, October 1988.

Laurig, W. and V. Rombach. "Expert Systems in Ergonomics: Requirements and Approach." *Ergonomics* 32(7): 795-811.

Lihou, D.A. "Computer-Aided Operability Studies for Loss Control." Presented at the Third International Symposium on Loss Prevention and Safety Promotion in the Process Industry, Basle, Switzerland, 1980.

McNichols, C.W. "Information System Building Blocks." *Microcomputer Applications in Occupational Health and Safety.* Chelsea, MI: American Conference for Governmental Industrial Hygienists, 1987. 27-57.

National Institute of Environmental Health Sciences (NIEHS) and U.S. Dept. of Energy (DOE). *Anticipating Occupational Hazards of Cleanup Technologies: Remembering the Worker.* Tech. Rep. Draft. Silver Spring, MD: NIEHS & DOE, November 1995.

Pheasant, S. *Body Space: Anthropometry, Ergonomics and the Design of Work.* 2nd ed. London: Taylor & Francis Ltd., 1996.

Poucet, A. and K.E. Peterson. "A Software Tool for Advanced Reliability and Safety Analysis." Presented at International SRE-Symposium, October 1987.

Suokas, P., et al. "Expert Systems in Safety Management." *Journal of Occupational Accidents.* 1-3(1990): 63-78.

Technical Research Center of Finland. "Vol. 1: Modeling," *RELVEC Manual.* 1986.

Technical Research Center of Finland. "Vol. 2: User's Guide," *RELVEC Manual.* 1986.

Your Feedback

Did you find this article interesting and useful? Circle the corresponding number on the reader service card.

RSC#	Feedback
36	Yes
37	Somewhat
38	No