# Expert Systems

## *What SH&E managers need to know about software verification and validation*

**By Gary L. Winn, Bhaskaran Gopalakrishnan, Magdy Akladios and Rajaarunprasad Premkumar**

AN EXPERT SYSTEM is a computer program that solves specific, complex problems (Goetsch). Such a system relies on a warehouse of detailed information about fires, mine emergencies or other contingencies that demand precise, rapid responses to events which may be developing as the user sits at the terminal.

Although expert systems are used in many organizations where time and criticality are key, some have failed. Like human experts, an expert system will be wrong at times even if it contains no programming errors. For an SH&E professional, this is especially critical when the system will be used by those who cannot easily judge the accuracy of the advice from the expert system (Wentworth, et al).

### Ensuring Safety Is Critical

Any system crucial to safety and health decision making must be verified and validated. Verification and validation (V&V) ensures that the system provides accurate, consistent results. This article uses *TEXPERT* (Winn, et al) as a platform to discuss why both SH&E managers and design engineers must know the V&V process so they can ask the right questions of potential software suppliers. An SH&E manager must know that expert systems have been independently and thoroughly tested before they are used in real situations.

Technology advances have automated many complex systems that once required extensive human involvement. For example, vehicle system operations from automobiles to aircraft depend on automated use of data generated by both on-board and off-board sources (Raeth, et al). Diverse systems such as process, manufacturing and power generation plants also depend on a large volume of stored data that is transferred using computers and other complex equipment. The volume of data in each situation expands as task and system complexity increase (Raeth, et al). These systems must perform efficiently and flawlessly in order to achieve accurate, safe and cost-effective operation. If the systems fail, tragedies such as that which occurred at the Three Mile Island nuclear power plant and during the V-22 Osprey Tiltrotor crash would occur (Raeth, et al). These incidents highlight the importance of software quality assurance that is crucial when dealing with hazardous or high-risk environments.

Independent V&V can be defined as "a series of technical and management activities performed by someone other than the developer of a system to improve the quality and reliability of that system and to ensure that the delivered product satisfies the user's operational needs" (Lewis). It is largely a "transparent" step for software purchasers. The process consists of a team of people familiar with user needs and with the programmers and programming languages being used. The V&V team is independent of both groups, however, and maintains this position for the sake of credibility. In many cases, a V&V team can be hired by a programmer to test the software.

In the case of safety-related expert software that

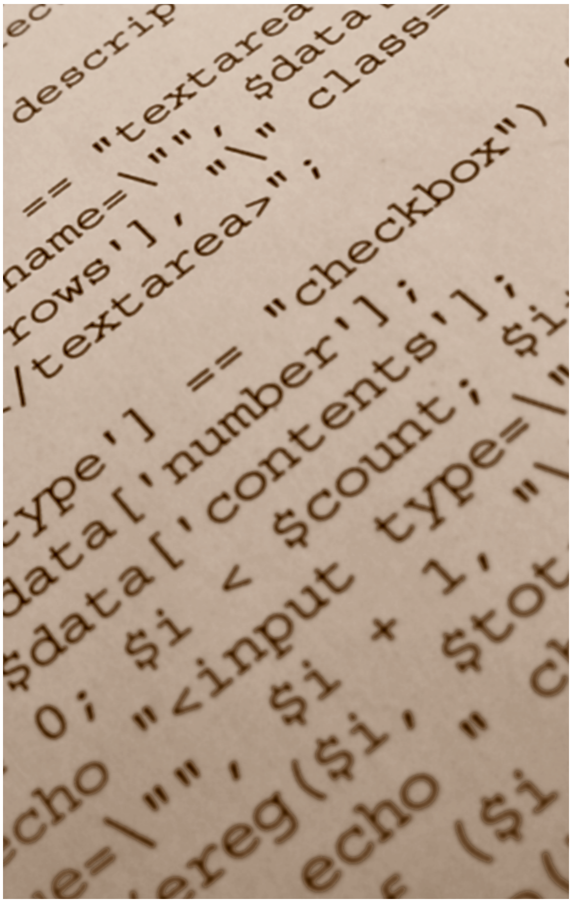
**Abstract:** *The practice of software verification and validation (V&V) has been overlooked in many safety-related expert systems even though it is central to a system being ultimately used to prevent hazards. This article examines the process of internal V&V of an expert system, explains those aspects of importance to an SH&E manager or design engineer, and reviews reusability issues of the actual metrics of V&V.*

**Gary L. Winn, Ph.D., CHST,** *is a professor in the safety management program within the Dept. of Industrial and Management Systems Engineering at West Virginia University (WVU) in Morgantown. He is a professional member of ASSE's Northern West Virginia Chapter. He holds a Ph.D. from Ohio State University, an M.A. from University of Dayton and a B.A. from Wright State University.*

**Bhaskaran Gopalakrishnan, Ph.D., P.E.,** *is a professor in WVU's Dept. of Industrial and Management Systems Engineering. He holds a Ph.D. in Industrial Engineering and Operations Research from Virginia Tech, an M.S. in Operations Research from Southern Methodist University and a B.E. in Production Engineering from the University of Madras.*

**Magdy Akladios, Ph.D., CSP,** *will be an assistant professor in safety engineering at the University of Houston, Clear Lake, beginning in the Fall 2005 semester. He holds a Ph.D. in Engineering, an M.S. in Engineering and an M.S. in Occupational Health & Safety Engineering, all from WVU. He also has a B.S. in Mechanical Engineering from Cairo University.*

**Rajaarunprasad Premkumar, M.S.,** *is a doctoral student at WVU, where he is studying expert systems and artificial intelligence.*



**Code reading requires the researchers to painstakingly read through code to detect visible errors such as logical correctness, and typographical and grammatical errors.**

might tell a worker what lever to pull or valve to close in an emergency, end users (purchasers) must be satisfied that problems of logic, endless loops and even plus or minus signs are outputs that are correct according to the knowledge team. The programmers and users will not know which action the expert software wants the user to take, so the V&V team is an independent link which assures that the outputs are not only logical and fast, but correct. Figure 1 shows the relationship between the users, programmer, V&V team and knowledge-base creator.

### Verification

The first step in the verification process is to check for requirements in the knowledge base; this represents what people have decided can or should be done (Figure 2). The primary purpose of conducting the "requirements verification" is to check the correctness and appropriateness of the knowledge base in terms of accuracy of expected results.

This is achieved using logical verification and rule verification. Logical verification is the verification of the expert knowledge for completeness and consistency for the domain model. In this context, completeness is the ability of the expert system to produce some conclusion for all possible inputs; consistency is the system's ability to produce a standard set of conclusions that are true for all possible inputs.

Rule verification checks for subsumed/redundant rules, inconsistent rules, dead-end rules, circular rules and unreachable conclusions (Wallace, et al). Subsumed rules are those that have identical "thens," but the premises of one are a subset of the other. Circular rules point back to a previous rule, which is not helpful in a software. Dead-end rules, if encountered, will stop the execution of the consultation abruptly for no apparent reason; they are not helpful.

Design and code verification activities occur in the next phase of V&V. This is achieved using partition testing. Test cases are selected using partitions of the input and output space as criteria and the team checks whether the specification addresses those cases (Wallace, et al). Using *TEXPERT* as an example, the results of verification for attachment components are discussed here. "Attachments" for an undesigned machine can be defined as different equipment attached to the machine that will perform special functions. In this case, attachments are determined to be buckets, air hammers or a grappling claw to manipulate an object at a safe distance. Not all attachments work the same and may pose special hazards.

Using an automatic validation function, the *Resolver* system generates a tree structure to identify dead-end rules and rules with unreachable conclusions. The questions and choices for this tree were then expanded one step at a time to check the

## TEXPERT

*TEXPERT* stands for "technical expert system." Developed by a team of researchers at West Virginia University, *TEXPERT* is a demonstration software created to provide a useful working link between design engineers and "warehoused" expert safety knowledge provided by computer in situations where the need for rapid safety decisions is mixed with complex technologies. The core idea was to develop an expert system that would help engineers who design hazardous waste remediation equipment to design a product which is minimally hazardous for humans to use when the equipment reaches the market.

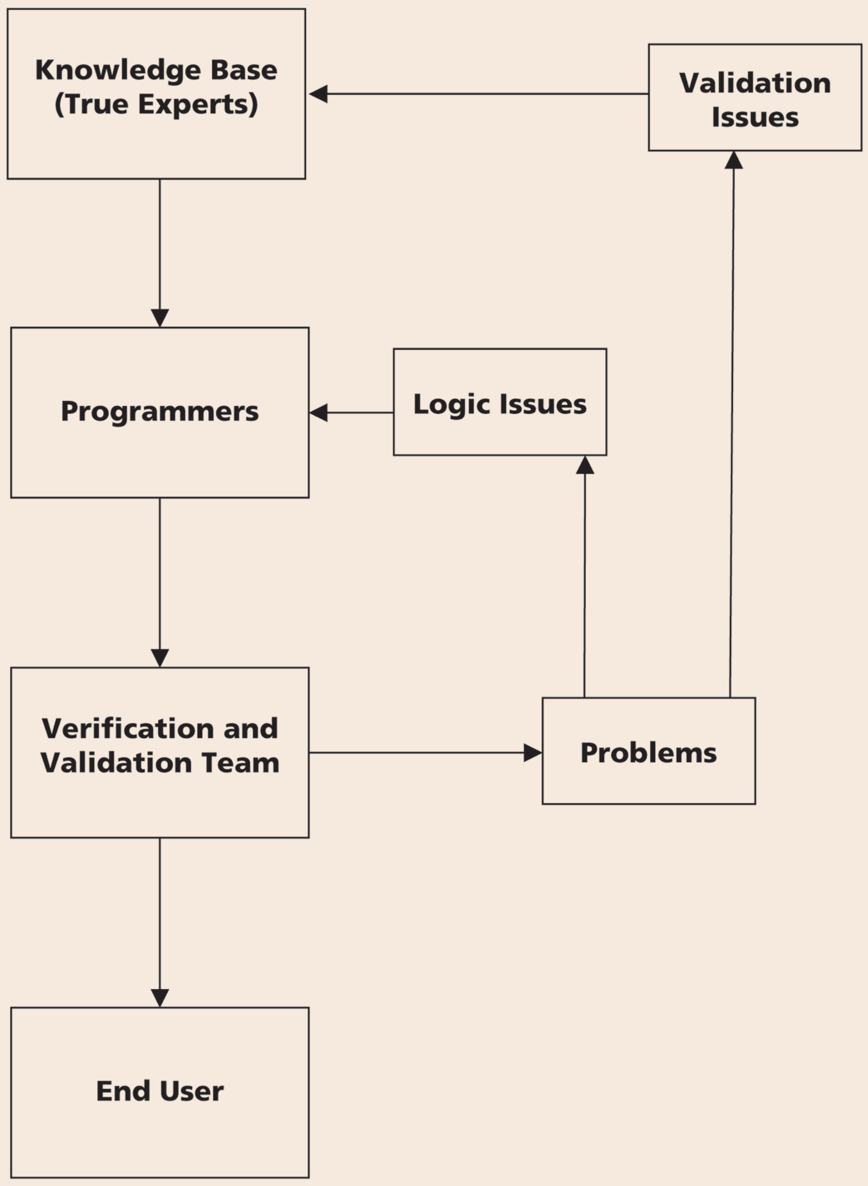
The system's architecture was designed with an HTML user interface connected to an expert system server called *Resolver,* a software product from MultiLogic Corp. *Resolver* is a knowledge-based systems development tool that combines a rule editor with a flexible visual decision tree interface and an inference engine (MultiLogic). If a wrong piece of information is given to a designer who is not aware of safety and health issues, a hazardous design may result or a technology safety data sheet (TSDS) may be created that contains potentially life-threatening recommendations instead of protective measures.

A TSDS is a technology-specific document designed to provide the identity and relative risk of safety and health hazards associated with the technology. Its purpose is to convey hazard information to workers who will interact with the technology to help them protect against potential hazards that may be associated with the technology. It provides potential hazard information to the user and SH&E professionals in a format that is easily understood [IUOE(b)].

The SH&E manager whose company may purchase or use technically complex equipment and is relying on a expert software should be confident that the system will be the safest possible tool for employees. An independent V&V process is critical to that assurance.

Figure 1

Expert Software Development Process


Knowledge Base (True Experts)
Validation Issues
Programmers
Logic Issues
Verification and Validation Team
Problems
End User

reason for further error. In the *TEXPERT* case, the system was not able to reach a conclusion because user input values for the questions implied that all safety requirements for the component had been met. The rules for the attachment component were studied for subsumed, circular and redundant rules. The "Rules" sidebar on pg. 49 illustrates the rules which suggest that no circularity or redundancy problems existed.

**Validation**

Validation is a process of executing an expert system and comparing test results to required performance (Lewis). Validation enables one to say unequivocally that the system is producing results only for the set of given input values. This process must be tailored to each expert system, matched to available resources and adapted to the testing methods to be used (Lewis). For *TEXPERT,* the validation process consisted of code reading and testing using in situ functioning equipment.

Code reading requires the researchers to painstakingly read through code to detect visible errors such as logical correctness, and typographical and grammatical errors (Wallace, et al). Logical errors occur when the system proceeds with the consultation to a user input value for a question for which the consultation should have been terminated. For example, a rule of the attachments components (shown below) was found to be logically incorrect.

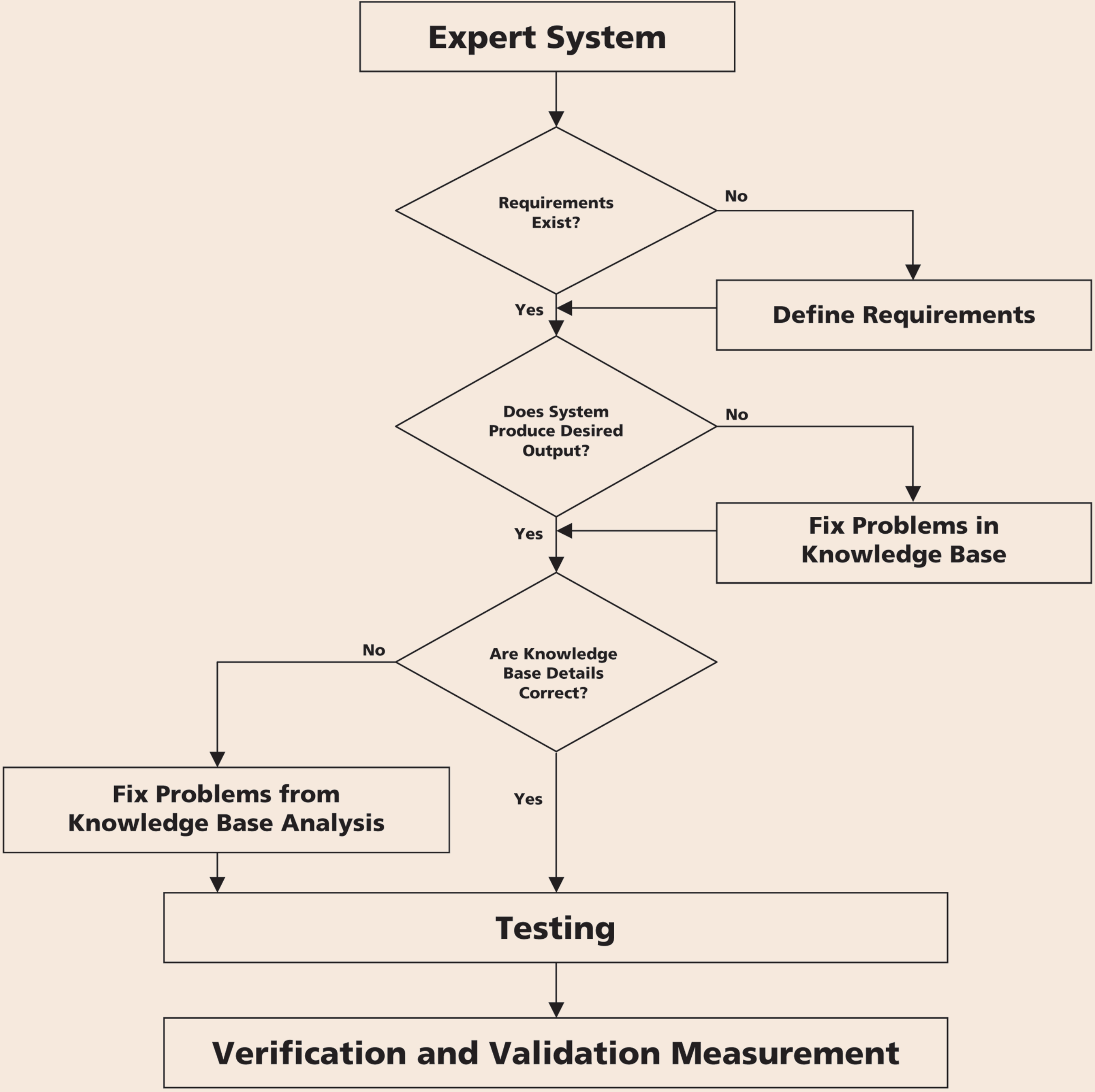
> **IF:** Are there assigned loads to this attachment? No.
> **THEN:** Event~401~ If overloaded, machine may tip or lose stability, and Standard Operating Procedures~401~ Assign maximum loads to the attachments to avoid tipping hazards.

The consultation should have ended with the user input value of "no." If no loads are assigned to the attachment, then the possibility of the machine tipping over is ruled out. The standard operating procedure (SOP) cited is ambiguous. It could be more clearly stated as follows: "Only assign loads that are less than the maximum permissible load for attachments to avoid tipping hazards."

Typographical and grammatical errors have the least effect on the performance of the *TEXPERT* system. These errors include misspelled words and logically incorrect sentences, which can alter the

Figure 2

## Steps in V&V Process


Expert System
Requirements Exist?
No
Define Requirements
Yes
Does System Produce Desired Output?
No
Fix Problems in Knowledge Base
Yes
Are Knowledge Base Details Correct?
No
Fix Problems from Knowledge Base Analysis
Yes
Testing
Verification and Validation Measurement

connotation of a recommendation or SOP. For example, "SOP~303~, Remove all bystanders from the are of operation," would have no logical value but would have tremendous value when "are" becomes "area."

"Testing in the context of verification and validation involves conducting tests to execute the complete expert system" (Wallace, et al). A test case executes part or all of the system to check whether user requirements are satisfied. In this case, the test plan was to break down the equipment into individual components and to validate each individual component. This is because the *TEXPERT* system consists of different components that are evaluated one at a time. The results were checked to see whether they are meaningful to the user inputs. If any anomalies were detected, changes to the knowledge in the expert system are recommended and the system is retested.

The pit viper system (photo, pg. 50) was selected for the test. That system is designed to perform decontamination, maintenance, modification and equipment reconfiguration operations in Hanford tank farm pits by remote means. Operations such as debris removal and cleaning are performed at the Hanford site located north of Richland, WA (Smith).

The system was divided into individual components and a team of safety experts with mechanical engineering expertise determined input values for components. The *Resolver* system was run on *TEXPERT* using the components of the pit viper system. Outputs were then compared with the technology safety data sheet (TSDS), which was prepared by a team from the IUOE National HazMat program (IUOE). This ensured that the expert system was producing valid outputs. Finally, a severity index of each event produced by the *TEXPERT* system was validated by comparing it with hazards identified in the TSDS.

The output of testing is shown in Figure 3. The event identified by the expert system was then compared with the TSDS ("TSDS" sidebar, pg. 50). Event~301~ correctly corresponds to the electrical hazard shown in "Section 4D" of the TSDS with the same hazard rating. By comparing the full contingent of other events to the TSDS, it was seen that *TEXPERT* is producing correct outputs and remedies in the form of recommendations and SOPs.

### V&V Measurement

V&V measurement is performed to check the effectiveness of the process in refining the system. V&V measurement can be achieved by defining metrics on which a quantitative assessment of the product or process can be obtained. Metrics are quantifiable measures of discrete quality attributes of documents, code and tests (Wallace, et al). This is an important step if the system is to be made reusable.

As industrial use of computer systems has grown, more software must be developed to ensure that they work perfectly. Software complexity has also increased at an exponential rate as newer programming methods and protocols are adopted. Since significant cost is associated with software development, many new techniques have been developed to make software components reusable and to build a repository of these components (Wallace, et al). These components can be taken at any time and incorporated into new software if applicable, thereby reducing cost and delivering a quality software.

Careful attention is required when components are reused in safety-critical systems. One must determine fit with the new system and the relationship between V&V activities of the reusable component as it is integrated into the new system (Wallace, et al). One must also understand the various differences between the original operating environment and the new environment. V&V activities using proper metrics help determine whether the software component can be used to build a larger system.

## Rules of Attachment Components in TEXPERT

### Rule Number 1

**IF:** Is equipment remote or ride-on? Remote.

**THEN:** *Event~301~* Possible electrocution if remote box was attached to machine by a tether or cable, and attachment comes in contact with a live electric line during operation. (*Confidence=10/10)*

*Standard Operating Procedures~301~* Utilize a policy that demands Call Before You Dig procedures. (*Confidence=10/10*)

### Rule Number 2

**IF:** Is the remote control box isolated? No.

**THEN:** *Event~302~* Possible explosion if attachment comes in contact with a gas line or other explosive material. (*Confidence=10/10*)

*Standard Operating Procedures~302~* Utilize a policy that demands Call Before You Dig procedures. (*Confidence=10/10*)

### Rule Number 3

**IF:** Are there bystanders in the vicinity of the operation of this attachment? Yes.

**THEN:** *Event~303~* Bystanders may get injured due to an explosion, flying objects or electrocution if standing in the vicinity if the operation of this machine. (*Confidence=10/10*)

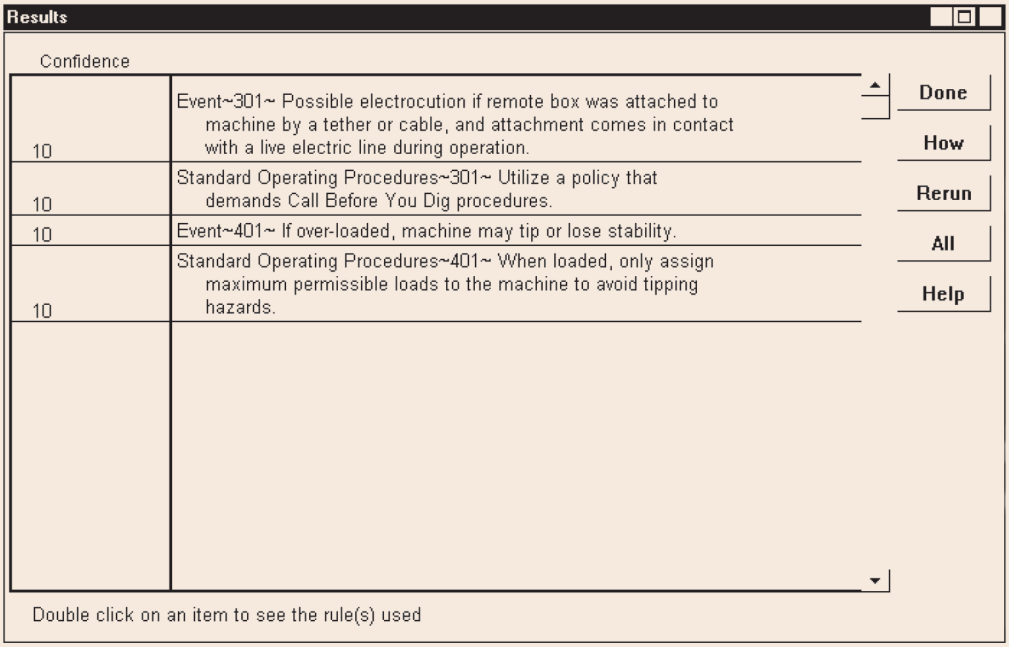
*Recommendation~303~* Incorporate a warning flashing light and/or alarm to warn bystanders of approaching hazards. (*Confidence=10/10*)

*Standard Operating Procedures~303~* Remove all bystanders from the area of operation. (*Confidence=10/10*)

*Standard Operating Procedures~303~* Utilize a warning line to keep bystanders at a safe distance from hazards. (*Confidence=10/10*)

Figure 3

## Results of Testing: Attachment Component



| Confidence | |
|---|---|
| 10 | Event~301~ Possible electrocution if remote box was attached to machine by a tether or cable, and attachment comes in contact with a live electric line during operation. |
| 10 | Standard Operating Procedures~301~ Utilize a policy that demands Call Before You Dig procedures. |
| 10 | Event~401~ If over-loaded, machine may tip or lose stability. |
| 10 | Standard Operating Procedures~401~ When loaded, only assign maximum permissible loads to the machine to avoid tipping hazards. |

Double click on an item to see the rule(s) used

## TSDS for Pit Viper System

### Section 4: Safety Hazards

*Hazard Category**

1: Could result in injury or illness not resulting in a lost workday.

2: Could result in injury or occupational illness resulting in one or more lost workdays.

3: Could result in permanent partial disability or injuries or occupational illness that may result in hospitalization of at least three persons.

4: Could result in death or permanent total disability.

N/A: Is not applicable to this technology and poses no appreciable risk.

**Adapted from Appendix A to MIL-STD-882D, Feb. 10, 2000, Dept. of Defense Standard Practice for System Safety.*

**A) Buried Utilities, Drums and Tanks**
**Hazard Rating: 1**

Buried tanks are present under the pits, but personnel will not be required to access them. Dome loads are strictly controlled on a task basis.

**B) Chemical (Reactive, Corrosive, Pyrophoric, etc.)**
**Hazard Rating: 2**

Although the vehicle the pit viper is attached to is not a part of this technology, hazards associated with the vehicle should be considered, such as diesel fuel, hydraulic fluid, and lubricants. Site-specific programs should be followed or developed taking into account the type of machinery used.

**C) Confined Spaces**
**Hazard Rating: N/A**

Since personnel access to the pits is prohibited, no confined spaces are associated with the pit viper technology.

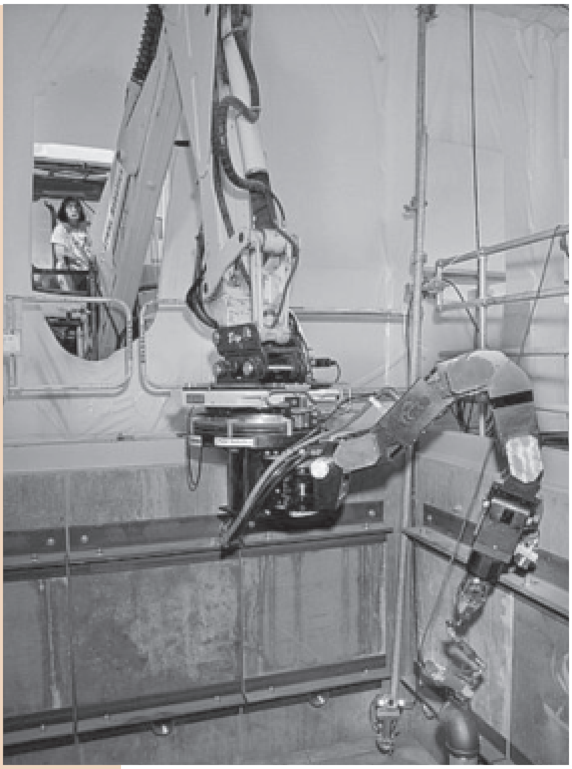
**D) Electrical**
**Hazard Rating: 3**

•The possibility of electrical shock is significant. System requirements include:

1) Hydraulic power unit: 480-volt AC 3-phase.
2) Control trailer: 480-volt AC 3-phase.
3) Camera and lights: 120-volt AC 1-phase at camera controller and lights operate with 12-volt DC at camera.
4) Cybernitx arm: 24-volt DC.

•The tools used with the arm may present their own electrical hazards and need to be evaluated.

•Power cords on power tools could catch on debris and other objects at the bottom of the pit. This may expose the electric power cord.



For *TEXPERT,* metrics such as lines of code, number of errors, number of errors by type (e.g., typographical, logical) and fault density were defined using the error data from the V&V activities. These metrics were selected because they are cost-effective and easy for safety managers to understand.

The lines of code metric is "any line of program text that is not a comment or a blank line, regardless of the number of statements or fragments of statements on the line" (Wallace, et al). It is the first metric that the V&V team will use because of its simplicity and cost-effectiveness. Through this metric, it is possible to estimate the effort and time scale during the development of the system. This information can be further used to develop assessment guidelines for selecting other metrics that will provide information about system effectiveness (Wallace, et al).

Fault density is computed by dividing the number of faults by the size (Wallace, et al). In software engineering terms, a fault is the encoding of human error into a software system. Fault density is useful to determine whether sufficient testing has been conducted based on the predetermined goals established. Also, fault densities can be used as a reference standard for comparison and prediction of system quality (Wallace, et al). Lower values of fault density imply that the system is working well and that no anomalies exist.

The number of errors metric provides an initial quantitative value of the incompleteness within the system. It is effectively used with statistical process control (SPC) techniques (Wallace, et al). More attention must be given to the development process of components identified by this metric when system reusability is being considered. Number of errors by type is used to uncover the most common error types in the system. This provides a means for tracking and categorizing errors that are similar in nature. Severity of the errors can be recognized using this metric; it can also be used to address the most common cause when reusability issues are considered. These metrics can provide valuable information about the overall testing process and error resolution process and focus on the areas of highest value.

Table 1

## Lines of Code for TEXPERT Components

| Component | Lines of Code |
|---|---|
| Attachment | 134 |
| Boom/Frame-Cabin | 158 |
| Boom | 220 |
| Boom/Wheels | 33 |
| Boom/Attachment | 54 |
| Boom/Engine-Motor | 32 |
| Frame-Cabin/Wheels | 65 |
| Compressors | 128 |
| Computer Systems | 204 |
| Conveyor Belts | 281 |
| Engine-Motor/Wheels | 43 |
| Engine/Motor | 263 |
| Fans and Blowers | 315 |
| Frame/Cabin | 272 |
| Hydraulic Lines | 85 |
| Engine-Motor/ Frame-Cabin | 185 |
| Other | 329 |
| Pipelines | 451 |
| Refrigeration Unit | 459 |
| Storage Tanks | 167 |
| Wheels | 322 |

In this test case, the lines of code are the total number of lines used to develop the expert system; this includes all individual component codes used to build the system. The number of lines of code used to develop the system was 4,200 (Table 1). Table 2 shows the number of errors and errors by type found in each system component. The errors that are most important from the V&V viewpoint are typographical, grammatical and logical errors. The severity of the errors is interpreted as 1 being the highest and 3 being the lowest. This helps identify those errors that require more attention when the system is being reused.

The fault density for the *TEXPERT* system was found to be 0.000479, assuming the weighting factors to be 10, 3 and 1 based on fault severity. This value suggests that the system will perform its function correctly and that most errors have been eliminated.

SPC involves the application of statistical methods to gather information necessary to continuously control and improve activities throughout the development process of the system (Wallace, et al). The advantage of using SPC is that it provides a quantitative measure of the system, which minimizes guesswork. It also helps detect errors early, which can reduce costs.

Table 2

## Number of Errors by Type Found in TEXPERT Components

| Component | Logical | Typo | Grammar |
|---|---|---|---|
| **Severity** | **1** | **13** | **12** |
| Attachment | 2 | 3 | 0 |
| Boom/Frame-Cabin | 0 | 0 | 2 |
| Boom | 1 | 14 | 3 |
| Boom/Wheels | 0 | 0 | 1 |
| Boom/Attachment | 0 | 2 | 2 |
| Boom/Engine-Motor | 0 | 1 | 0 |
| Frame-Cabin/Wheels | 0 | 4 | 0 |
| Compressors | 0 | 2 | 1 |
| Computer Systems | 0 | 3 | 0 |
| Conveyor Belts | 2 | 4 | 2 |
| Engine-Motor/Wheels | 0 | 6 | 2 |
| Engine/Motor | 1 | 5 | 1 |
| Fans and Blowers | 0 | 11 | 1 |
| Frame/Cabin | 0 | 9 | 3 |
| Hydraulic Lines | 0 | 3 | 0 |
| Engine-Motor/ Frame-Cabin | 0 | 3 | 1 |
| Other | 0 | 3 | 3 |
| Pipelines | 0 | 13 | 0 |
| Refrigeration Unit | 2 | 11 | 1 |
| Storage Tanks | 0 | 4 | 1 |
| Wheels | 2 | 3 | 2 |

Performing this analysis would prove to be of great help when system reusability and continuous control is required for the system (Wallace, et al). For the *TEXPERT* case, the SPC tools used were bar graphs and scatter diagrams. A bar graph is a frequency distribution diagram in which each bar represents a characteristic, and the height of the bar represents the frequency of that characteristic (Wallace, et al). A scatter diagram is a plot of the values of one variable against those of another variable to determine whether a relationship exists between them. If no apparent pattern is observed, then no relationship exists between the variables.

Figure 4 depicts a bar graph in which the horizontal axis represents the type of error and the vertical axis represents the number of errors obtained from the V&V process of *TEXPERT.* Using the metric, number of errors by type was plotted. Typographical error was the most common error type found in this case.

Similarly, a scatter diagram (Figure 5) was plotted using the lines of code by component on the horizontal axis and the number of errors by component in the vertical axis. The goal was to determine

Figure 4

## Most Common Error Types in TEXPERT

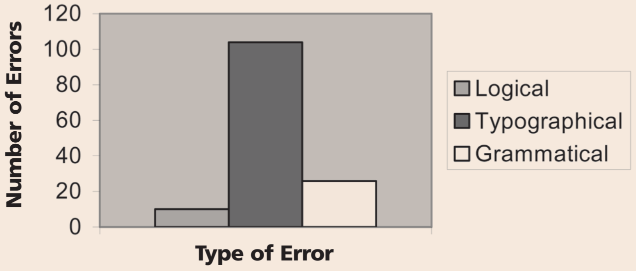
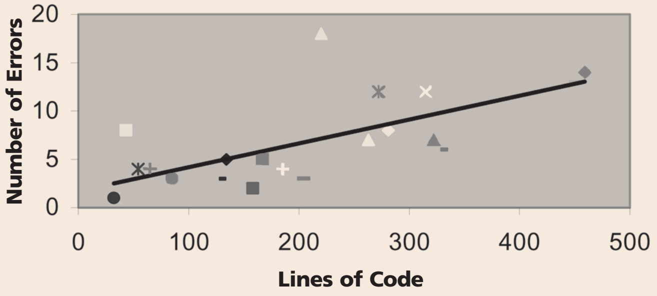


Figure 5

## TEXPERT Errors vs. Codes by Component



whether any linear relationship existed between the two variables. As Figure 5 shows, no pattern exists in the plot.

SH&E managers can ask to examine V&V documents. The software developers should have such information, although it is not commonly requested. The SH&E manager should ask how V&V was accomplished; what metrics are available for examination before purchase; how logical or knowledge-base patterns were resolved; and how these same problems will be accommodated in the future (i.e., are purchasers notified of problems or are they on their own after the purchase).

### Conclusion

The quality of an expert system or any other software system depends on the quality of knowledge used as input. Problem areas that may lead to rippling effects when a system is reused or modified must be identified, as this may lead to increased costs. Independent V&V provides insight regarding the software's quality and reliability, and it is a process SH&E professionals should confirm before they consider purchasing an expert software. ■

## What to Ask Before Buying an Expert System

- Is the software available for free or at a cost to the user? "Free" may suggest that corners were cut or that V&V was not performed.
- Does the software maker claim that V&V has been performed? Does the maker offer evidence in sales literature?
- If V&V have been conducted, have they been documented properly? Can a potential buyer review V&V documents generated as a result of the processes?
- Are V&V measurement metrics available for examining the efficiency of the software? Were the technologies tested actually representative of the buyer's intended uses?
- Are V&V measurement metrics available for examining the reliability of the software? The buyer would want to review data related to how many knowledge-base questions were addressed and how many logic bugs were corrected. Will the software maker disclose errors in terms of type and degree?
- To what degree has the software been debugged internally among the development team? How many developers actually ran through the whole software, how many times, and what errors did they find and correct?

The purchaser should be able to obtain reasonable answers to these questions. If the developer is hesitant or unable to respond to many of these questions, the purchaser should be cautious.

### References

**Goetsch, D.L.** *Occupational Safety and Health for Technologists, Engineers and Managers.* 4th ed. Columbus, OH: Prentice Hall, 2002.

**International Union of Operating Engineers (IUOE)(a).** "Pit Viper System: Human Factors Assessment Program: Technology Reports." Beaver, WV: IUOE. <http://www.iuoeiettc.org/Pdf%20files/HFA/Pit%20Viper%20Final%20TSDS%20in%20new%20format.pdf>.

**IUOE(b).** "Technology Safety Data Sheet: Human Factors Assessment Program—Technology Reports." Beaver, WV: IUOE. <http://www.iuoeiettc.org/Pdf%20files/Technology%20Safety%20Data%20Sheet.pdf>.

**Lewis, R.O.** *Independent Verification and Validation: A Life Cycle Engineering Process for Quality Software.* New York: Wiley Publications, 1992.

**MultiLogic.** *Resolver User Manual.* St. Paul, MN: MultiLogic Inc., 1998.

**Raeth, G.P., et al.** *Scalable Expert Systems for Adding Crisp Knowledge to Pilot-Vehicle Interfaces.* WL-TR-96-3087. Wright Patterson Air Force Base, Ohio: Wright Laboratory, 1996.

**Smith, R.L.** "Pit Viper Strikes at the Hanford Site." *Radwaste Solutions.* May/June 2002: 33-39.

**Wallace, D.R., et al.** "Reference Information for the Software Verification and Validation Process." NIST Publication No. 500-234. Gaithersburg, MD: National Institute of Standards and Technology, 1996.

**Wentworth, J.A., et al.** *Verification, Validation & Evaluation of Expert Systems.* Vol. 1. Washington, DC: Federal Highway Administration, 1995.

**Winn, G.L., et al.** "TEXPERT: A Tool for Safety Professionals & Design Engineers." *Professional Safety.* Oct. 2002: 32-37.