LOGIC MODEL NEEDS FOR DIVERSE FACILITY TYPES

James R. Wilson
Idaho National Engineering Lab
Box 1625, MS-3412
Idaho Falls, Idaho
83415-3412

ABSTRACT

This paper compares the characteristics of fault trees (where initiators are developed within the fault tree) vs. event trees (where the nodes are developed by fault trees). This comparison requires some additional discussion on the subtlety of initiators. Difficulties when analyzing various reactor-type and processing facilities are discussed to illustrate the particular characteristics of each type of logic. The intent is to allow probabilistic risk assessment (PRA) analysts to be "bi-logical," or equally comfortable with event-tree or fault-tree logic, knowing when to apply each.

NOMENCLATURE

Enabler -- An event that occurs before the initiator (e.g., a latent fault or pre-existing condition) that contributes directly to the accident.

Event tree approach -- This terminology is applied to the typical nuclear reactor

analysis where each event tree begins with an initiating event, and each node is developed by fault trees conditioned upon preconditions set prior to that node.

Fault tree approach -- This term is applied to an analysis wherein the initiating events are developed within the fault tree.

Initiator -- An event that causes a system to exceed safety parameters, occurs at the time of that exceedance, and has units of reciprocal time.

Mitigator -- An event that occurs after the initiator, does not contribute to the accident frequency, but may affect the consequences of that accident.

Mitigator system -- A system in which most of the defence occurs after the initiating event.

Prevention system -- A system in which most of the defence occurs before the initiating event.

PRA -- Probabilistic Risk Assessment, using either the event tree approach or the fault tree approach.

DISCLAIMER

BACKGROUND

Probabilistic Risk Assessment (PRA) has become the accepted approach for predicting the future for design of nuclear and chemical plants, aircraft design, nuclear repository dose assessments, and environmental issues. For most people, the decision about whether to model initiating events and accident sequences using event trees or fault trees was settled in 1974 with the Reactor Safety Study. In that study, a fault-tree-only structure was abandoned after some effort because of the unwieldiness. Since then, the standard has been to portray the initiators, and event ordering on an event tree, with the event tree nodes developed by fault trees.

BUT, EVENT TREES DON'T WORK AS WELL

The event tree approach does not work as well in process industries, where the fault-tree-only approach (hereafter called "fault tree approach" or simply "fault tree") is more common than in the reactor world (Smith, 1976 and Frauenholz, 1993). About the same time as the Reactor Safety Study, a landmark study using the fault tree approach was done (Smith, 1976).

Under certain conditions, the event tree approach could require ten times as many pages, or could result in an incomplete analysis. For example, in one study

(Frauenholz, 1993), over 10,000 sequences were produced, with thousands of initiators that were difficult to group. With each of these initiators requiring at least one page for its event tree, this study would have required many times more pages for printing.

The characteristics of each approach will be discussed below to better understand their differences.

EVENT TREES

Event trees use inductive, or forward-looking logic, following the progression of the accident through time. Each train of logic starts with a postulated initiator, asks "What happens next?," and determines mutually exclusive scenario paths for which varying consequence states may be assessed.

The following are characteristics of the event tree approach:

1). Tracking of multiple consequence scenarios: A fault tree considers only one level of consequence. If the scenario does not produce a consequence that high, those events have no place on the tree. If a superset has a much higher consequence, that scenario is automatically deleted by the computer and all record of the higher consequence accident is lost. The event tree

"remembers" scenarios with varying consequence levels. When doing a "Farmer Risk Plot" or evaluating risk values over a range of consequence, event trees are a better representation.

2). When the consequences of accidents have not been well developed: Because of "forward-looking" logic, the event tree is an excellent tool for investigation of results. Therefore, the event tree excels where consequence end states must be developed or documented.

3). Compact representation of consistent subsystem interdependencies: The event tree is a very compact presentation when several scenarios involve time- or logic-ordered subsystems responding to the same accident initiator. For example, a common time-ordered shutdown sequence for a BWR is the following: high pressure injection, depressurization, low pressure core spray, and residual heat removal. For reactor-type facilities, the beginning of the accident may have very benign consequences immediately, but more severe consequences may develop over a period of time, depending upon the performance of these post-initiator systems. In cases like these, event trees are a better representation.

4). For "Mitigation" systems: For "mitigation" systems (Vail, 1992), such as nuclear reactors, most of the defense occurs after the initiator, in an attempt to mitigate the consequences of the accident. Characteristics of

such systems are significant energy management (e.g., post-accident core cooling), and severe accident consequences. Another reason for "mitigation" systems is when the regulator doesn't give credit for prevention (i.e., the initiating event must be assumed). Such systems are represented well by event trees.

TIME OUT FOR INITIATORS

The next section deals with the characteristics of fault trees, but first, because of the subtlety of many initiators (Wilson, 1982), a brief discussion is necessary.

An initiator can be defined as a "fault" or external event that causes a key safety parameter to go outside safety limits. That initiator occurs at the "time of the accident" and has units of a "frequency." These terms are in quotes because often they are difficult to apply, especially to multi-event initiators.

Frequency and Fault Conditions: Sometimes the initiator appears to be unitless or not to be a fault condition. For example, a chemical makeup error contributes to the TOP event of a tree, but the fault, operator makeup error, has units of "/demand." The number of makeups per year has frequency units, but is not a fault. In this case, the makeup error must be multiplied by the frequency of makeups per year to form a multi-event initiator that is a fault and has units of a

frequency.

Time of Accident: Sometimes the fault seems to occur long before the time of the accident. For example, an airplane mechanic forgets to put brake fluid in the plane, so it crashes upon its next landing. The fault, "mechanic leaves out brake fluid," does not occur at the time of the accident. In this case, the multi-event initiator is "Plane lands without brake fluid," which combines "mechanic leaves out brake fluid" with the "number of landings/year" for this plane. This initiator now has a frequency, is a fault, and occurs at the time of the accident.

Another case where the initiator occurs at a different time than the accident is the case of a reactor trip where the core melts 20 days later due to failure of residual heat removal. The initiator, "residual heat removal impacted due to...," may have occurred 20 days before the "accident," core melt. To understand this "violation" of the definition, we must define "enablers" and "mitigators" (Dunglinson, 1983). Enablers are latent faults or pre-existing conditions that occur before the initiator and prepare the scene for the initiator. Once the initiator has occurred, the accident exists. Various mitigation systems now come into play to try to control the consequences. The consequences may become more severe in time, due to the failures of mitigation systems, but the accident has already "happened." The mitigator cannot prevent the key safety parameter from going out of bounds, it can only modify the consequences. Understanding these principles of initiators is crucial in any field where initiators are not a given.

THE FAULT-TREE-ONLY APPROACH

The fault tree uses deductive or "backward-looking" logic. A consequence, or undesired TOP event, must be defined by the analysts and customers. Once the TOP event has been identified, all the valid causes or initiators for that consequence can be determined.

The following are strengths of the fault tree approach:

1). Fault trees logically develop initiators: Whereas the event trees developes accident-sequence end states, the fault tree looks backward in time to derive causes. Event trees require an identified initiator before development. These initiators can be developed various ways: based upon similar plants, the subject plant's operating history, design basis documentation, a HAZOP, FMECA, or by examining the focus of complex emergency response systems.

The fault tree excels where a logical search for initiators is desired or the thoroughness of that search must be documented. This may be necessary for one-of-a-kind plants, where operating history is limited, or where initiators are component failures or human errors instead of emergency response system failures.

2). Fault trees handle variable initiators better: The phenomenon of variable initiators (Wilson, 1993) is often a real test of the analyst's understanding of initiators. A variable initiator occurs when minor changes in a system (administrative controls or operator assumptions) can vary the scenario initiators. For example, one processing plant had a computerized plant emergency system that "looked over the shoulder of the operator," remaining passive until it picked up an illegal action by the operator and shut down the process. In such a system, a failure of this computer is not an initiator, because the accident does not happen until the operator makes the mistake. Thus, the operator is the initiator. However, at this particular plant, the operators wanted to know what the computer was thinking, so that their mistake wouldn't cause a shutdown. If the operators did know what the computer was thinking, they may fail to do their own thinking, or distrust their thinking when it disagreed with the computer. If the operators were to defer to the computer, trusting its decisions, then the computer could become an initiator. Thus, an administrative change, the operator being able to know what the computer was thinking, creates a new initiator!

Related to variable initiators is the concept of multiple initiators. Again, this is covered in more detail in Wilson (1993). Briefly, this occurs when any one of several events may act as an initiator, depending upon time ordering. For example, a room has a pipe carrying flammable liquid. The pipe leaks, then an operator walks in smoking a cigarette (contrary to posted warnings).

The initiator of the resulting explosion is the "operator enters the room while smoking." On the other hand, the room also has a explosion proof switch that has failed in such a way to represent a constant spark source. In this case, the explosion occurs immediately after the pipe leak, making the pipe leak the initiator. Time ordering declares the initiator to be the last event to occur. Thus, the logic must model two different scenarios due to the potential for either cut set event to act as an initiator. Admittedly, there is no reason why such insights can't be picked up by the event tree analyst, except that he is not looking in that area: He's assuming initiators and analyzing for consequences.

Also, when many operator interfaces exist (the plant is more manual than automatic), the initiators need to be looked at very carefully. Component failure mechanisms are very simplistic compared to the richness and creativity of human error, especially when it can act as an initiator.

Discovering variable initiators, multiple initiators, and human error initiators is more likely in the fault trees, whereas event trees discourage such thinking by locking in the initiator at the beginning as a "given".

3). Fault trees are more compact for "prevention" systems: Additional complexity, or lack of uniformity, occurs in processing systems that tend to concentrate on barriers to the initiation of the accident (i.e., the defense in

depth precedes the initiator, contrary to "mitigation" systems). Such "prevention" systems are the norm where little mitigation is possible after the accident. As opposed to "mitigation" systems where the accident may start small and grow large over time, the accident here happens quickly, with maximum consequence, allowing little response time afterwards. Evacuation is the usual response in such systems, not caretaking for a wounded and threatening system.

Because the complexity of the defense is up front, before the initiator, the analyst's attention must be directed here, requiring backward-looking logic.

4). Fault trees are more compact with multiple accident sites: Event trees excel where most of the scenarios revolve around a central location, like the reactor core. In process plants, a hundred accident sites (e.g., process vessels with different designs, operator interfaces, and chemistry) may exist. Each accident site is similar to reactor core, requiring unique initiators, and different system response or behavior. Thus, a full set of event trees for each accident site (with a new page for each initiator), times the number of accident sites, can result in an event tree analysis comprising many additional pages of logic compared to a fault tree analysis.

5). Inductive Design is More Thoroughly Tested by Deductive Reasoning: The design process tends to be inductive (forward looking, success

oriented). If the logic process that tests the designer's product is also forward looking, this commonality may result in a common cause, failure to discern the critical weakness. For this reason, the testing of the inductive reasoning of the designer by the deductive reasoning of the fault tree analyst seems a better mix. Indeed, this is usually done on the subsystem level, since most event trees use fault trees on this lower level. However, it wouldn't hurt, where other reasons favor the event tree approach, to do deductive reasoning on the system level, just as a check.

CONCLUSIONS AND RECOMMENDATIONS

No PRA group should do event-tree logic only, or vice versa. "When your only tool is a hammer, all problems look like nails." Analysts and companies must be bi-logical: We should all be comfortable with both approaches, using the appropriate one for each analysis.

REFERENCES

Dunglinson, C., Lambert, H., 1983, "Interval Reliability for Initiating and Enabling Events," IEEE Transactions on Reliability, Vol. R-32, No. 2, pp. 150-163.

Frauenholz, L. H., et. al., 1993, Fuel Processing Facility Final Safety Analysis Report, WIN-358, Westinghouse Idaho Nuclear Company, Inc.

Smith, T. H., 1976, A Risk-Based Fault Tree Analysis Method for Identification, Preliminary Evaluation, and Screening of Potential Accidental Release Sequences in Nuclear Fuel Cycle Operation, BNWL-1959, Battelle, Pacific Northwest Laboratories.

Vail, J. A., 1992, "The Selection of Probabilistic Safety Assessment Techniques for Non-Reactor Nuclear Facilities," ANS Topical Meeting on Risk Management -- Expanding Horizons, Boston, Mass., pp. 137-140.

Wilson, J. R. 1982, "Practical Probability - Initiators," Reliability Review, Vol. 2., No. 3, pp. 39-40.

Wilson, J. R., 1993, "Identifying Initiating Events", PSAII, ANS/ENS International Topical Meeting, Vol. 2, pp. 1275-1281.

Prepared for the

U.S. Department of Energy

Assistant Secretary for Environmental Management (EM)

Under DOE Idaho Operations Office

Contract DE-AC07-94ID13223