TOWARD A RISK-BASED APPROACH TO THE ASSESSMENT OF THE SURETY OF INFORMATION SYSTEMS*

Gregory. D. Wyss, Sharon. K. Fletcher, Ronald D. Halbgewachs, Roxana M. Jansma, Judy J. Lim, Marty Murphy, and Paul D. Sands
Risk Assessment and Systems Modelling Department
Sandia National Laboratories
Albuquerque, New Mexico

ABSTRACT

Traditional approaches to the assessment of information systems have treated system security, system reliability, data integrity, and application functionality as separate disciplines. However, each area's requirements and solutions have a profound impact on the successful implementation of the other areas. A better approach is to assess the "surety" of an information system, which is defined as ensuring the "correct" operation of an information system by incorporating appropriate levels of safety, functionality, confidentiality, availability, and integrity. Information surety examines the combined impact of design alternatives on all of these areas. We propose a modelling approach that combines aspects of fault trees and influence diagrams for assessing information surety requirements under a risk assessment framework. This approach allows tradeoffs to be based on quantitative importance measures such as risk reduction while maintaining the modelling flexibility of the influence diagram paradigm. This paper presents an overview of the modelling method and a sample application problem.

INTRODUCTION

Recent years have seen a dramatic increase in the use of information systems (computers and microcontrollers) in areas of human life ranging from consumer products to manufacturing equipment and the conduct of commerce. This period has also seen instances of information system failure causing the loss of great sums of money and potentially endangering human life. As these systems become part of critical processes, we must ensure that the information systems enhance system performance and safety, not only during "normal" operations, but also under abnormal and potentially adverse conditions. Traditional approaches to the assessment of information systems have treated system security, system reliability, data integrity, and application functionality as separate disciplines. However, each area's requirements and solutions have a profound impact on the successful implementation of the other areas. A better approach is to assess the "surety" of an information system, which can be defined as ensuring the "correct" operation of an information system through the incorporation of appropriate levels of safety, functionality, confidentiality, availability, and integrity. Information surety examines the combined impact of each design alternative on all of these areas.

The idea of integrating these currently separate disciplines under an information surety umbrella is relatively new. Sandia National Laboratories has assembled an interdisciplinary team to work toward a risk-based quantitative method for assessing the surety of information systems under its Laboratory-Directed Research and Development Program. This team is developing risk-based modelling methods that are applicable to information surety problems as well as providing guidance in the areas required to support such a method (characterization of threats and mitigation strategies, risk and requirements taxonomies, etc.). This paper is a progress report on the ongoing work of this interdisciplinary team. The paper summarizes the risk assessment modelling methodology developed by the team, and demonstrates the methodology for a sample problem. The other areas examined by the team will be published in other forums.

CURRENT APPROACHES TO THE ASSESSMENT OF INFORMATION SYSTEMS

Traditional approaches to the assessment of information systems have often been based on an *ad hoc* or piecemeal

*This work was performed at Sandia National Laboratories, which is operated by Martin Marietta Corporation for the U.S. Department of Energy under contract number DE-AC04-94AL85000.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

approach under which individual requirements are generated to protect against various real or perceived threats. These requirements often relate only to a particular area of information surety (functionality, security, etc.), but their *combined* impact is rarely considered systematically. Analysts also tend to focus on particular areas to the neglect of others (e.g., protection against the most recent incident, or protection against active threats while neglecting passive threats). Furthermore, requirements are often based on a seemingly predetermined checklist rather than an objective assessment of specific project needs. Some past implementations of this approach have produced security systems with very strong "doors" but wide-open "windows."

Assessment of information systems under a total systems approach would help reduce many of these problems. We believe that an information system design and assessment approach should have the following characteristics:

- The approach should not be merely "checklist" or compliance-based, but should assess the consequences that would occur should the system fail to achieve its surety objectives (*appropriate* levels of functionality, security, etc.), and how much a user is willing to spend to avert those consequences.

- The approach should provide quantitative information by which tradeoffs between various design alternatives can be objectively evaluated.

- The approach should be readily extendable (i.e., it should support both "quick-look" studies and the extension of those studies to arbitrarily greater levels of detail as appropriate).

- The approach should provide guidance to help the analyst ensure that the model considers all appropriate threats, mitigation strategies, and consequences, and a quantitative screening technique to help the analyst decide which scenarios can be legitimately neglected.

- The approach should be easy to use, but powerful enough to cross the boundaries that currently separate the various domains encompassed by information surety.

- The approach should be supported by software that would ease model development and automate model solution.

These criteria suggest that a probabilistic risk assessment (PRA) methodology would be appropriate for assessing information surety. Our evaluation of common PRA methods such as fault tree and event tree analyses found that none was appropriate to *all* aspects of the information surety problem. We also found that the thought process for evaluating most aspects of information surety began by looking for ways that the system's proper "process flow" could be disrupted, diverted, or caused to exhibit undesired behavior. For these reasons, we examined more generalized directed graph techniques and found that influence diagrams provided a good point of departure for our risk assessment studies. This paper discusses our adaptations to the influence diagram formalism to perform quantitative assessments of information surety problems.

MODELLING METHODOLOGY

An influence diagram is a probabilistic network that consists of nodes and arcs (Howard and Matheson, 1984; Jae and Apostolakis, 1992; and Jae *et al.*, 1993). It must be singly connected and acyclic. The nodes can represent system states, decisions, or chance or deterministic occurrences, while the arcs represent the conditional dependencies between these occurrences. Decisions are represented by square nodes, while chance and deterministic events are represented by circular and double-circular nodes, respectively. These nodes ultimately influence a "value node" (diamond), which quantifies consequences for each possible system configuration.

If a chance node is dependent upon other nodes, it represents a *set* of conditional probabilities, where the probabilities are conditional upon the results of the node's immediate predecessors. A deterministic node is a special case of a chance node where all probabilities are either zero or one. Thus, an influence diagram consists of four distinct parts: the nodes, the influences upon the nodes (the dependencies between them), the "function" that determines which probabilities are to be applied given each distinct set of influences, and the conditional probabilities themselves.

The influence diagram formalism is conceptually similar to the event tree and decision tree formalisms that are applied in various disciplines of risk analysis. There are three distinct advantages to influence diagrams for information surety analyses. First, it is much easier for an analyst to visualize and gain an intuitive understanding of an influence diagram model than a comparable tree-based model. Second, influence diagrams show the dependencies between events explicitly, while this information is hidden from the analyst in event trees and decision trees. Finally, the influence diagram formalism supports both "backward" and "forward" construction of the model, while event trees and decision trees can only be constructed in the "forward" direction. The "backward" model construction process is similar to the method used for fault tree development in which an analyst systematically decomposes a selected event to determine its root causes. The "forward" model construction process is used in event tree analysis to determine the logical consequences of a particular event or set of events. The influence diagram formalism supports both model construction methods. This flexibility and power provides an important reason to use influence diagrams instead of either fault tree or event tree/decision tree methods.

Conventional influence diagrams have three primary disadvantages when applied to information systems. First, the node symbology, because it is so very general, does not reflect some of the ideas that we believe are important to represent in a risk model of an information system. Second, the traditional solution method does not show or even generate the detailed set of scenarios possible in the model (Jae and Apostolakis, 1992;

Shachter, 1986). The ability to examine these paths in detail is a primary advantage of the event tree and decision tree methods. Finally, the traditional solution method makes it difficult to determine which nodes are the most important for various aspects of the results and, hence, to determine where one should look to improve the system. This very useful information is key to the results of typical fault tree analyses. We have, therefore, proposed some extensions to the notation and a new solution methodology to remove these deficiencies.

BUILDING A RISK MODEL OF AN INFORMATION SYSTEM

Our objective in modelling an information system is not so much to provide a "probability of failure" for a system as it is to help identify and prioritize that system's risks. Only after the risks have been identified and prioritized can an analyst make informed decisions about whether particular risks are acceptable and, if necessary, examine strategies to reduce those risks.

The starting point for a risk model should be to identify the consequences and benefits from the proper or improper operation of the information system. One should consider each "information surety objective" (safety, functionality, availability, confidentiality, and integrity) as it might affect risk areas such as the mission of the system or organization; worker health and safety; public health and safety; as well as legal, regulatory, political, social, and environmental impacts. This assessment helps identify which system states should be either encouraged or avoided, and forms the basis for the risk assessment model.

There are many ways to build a risk model based upon an influence diagram, but two of the most useful are as follows. Under the first method, we explicitly identify the undesired state and work to find the immediate, necessary, and sufficient conditions for that state to occur in much the same manner as during fault tree construction. We continue to apply these criteria recursively until each event has been resolved into its fundamental causes along with the system conditions required for these causes to successfully act upon the system. These fundamental causes (basic failures and initiating events) will form the starting point in our search for ways to reduce or eliminate particular system risks.

The second method for developing influence diagrams begins by drawing a diagram to represent the normal functional flow of the system (including the hardware, software, and data aspects of the system). We then examine every node to find influences that can cause the system to deviate from normal functionality toward an undesired state. We add events to the influence diagram to represent these influences, and seek to find their fundamental causes in much the same way as would occur for the first method. In addition, if we suspect *a priori* that particular events or conditions might lead to an undesired state, we can use these nodes as starting points (initiating events) and expand them forward into their universe of logical consequences to determine how they influence the normal and even the abnormal operation of the system. This shows the value of working both "forward" and "backward" when developing influence diagram models.

Another powerful feature of influence diagrams is that, like fault trees and other modular directed graph techniques, they can support iterative refinement. Thus, it is possible to initially construct a "high level" model for scoping studies using only a few broadly defined nodes, and to later refine the model to incorporate a more detailed knowledge of the system, its operation, and its vulnerabilities. It is also possible to construct a model in which some phenomena are examined only in coarse detail (a "screening analysis") and others at a much finer level.

So far, our risk model only considers influences that can lead us toward undesired states. We must also consider "positive" influences that can reduce the ability of "bad" influences to accomplish an undesired result. These are called "barriers" because they act as impediments to undesired outcomes much like a fence system acts as a barrier to prevent unauthorized access to a facility. Barriers show up as influences (nodes with appropriate arcs) in the influence diagram. Since a barrier node typically depicts whether a particular barrier is present or active, it is often represented as a chance node that is not influenced by any other nodes (i.e., an unconditional chance node), where the probability value represents the likelihood that the barrier is *implemented*, and *not* the probability that it is *effective*. Barrier effectiveness is modelled in the node that the barrier influences, and takes the form of the conditional probability that a particular influence will actually cause the system to deviate from its intended function. This allows us to assess the effects of multiple barriers on a single "bad" influence both individually and in combination. For example, we could consider controlling access to a facility using badges, passwords, and biometrics (e.g., hand measurements, retina scans). Our node that represents the chance of a person being granted inappropriate access would be influenced by four nodes that represent: (1) a person trying to gain access (if nobody wants access, we don't have any risk and don't need any protection), (2) the presence or absence of a badging system, (2) the presence or absence of a password system, and (4) the presence or absence of a biometric access control system. The effectiveness of each combination of barriers is measured by the conditional probability that a person is granted inappropriate access. This probability varies depending upon which combination of controls (barriers) is in use.

Initiating events, basic failures, and barriers (engineered or otherwise) are not conceptually different from other chance or deterministic events. However, using different influence diagram symbols to represent these events would help the analyst to more quickly assess the completeness of the list of threats (initiating events and basic failures) and to identify any unmitigated threats (paths without barriers to undesired consequences). Therefore, we propose to extend the traditional influence diagram symbol set as shown in Figure 1. We propose using a house symbol to represent initiating events (as these can be likened to external events in a fault tree analysis, whose symbol they would share), and a triangle to represent a barrier. A basic failure would be represented by a circle, as it is essentially a chance occurrence (and is similar to a basic event in a fault tree analysis, which is also represented by a circle). It is possible to build a risk model of an information

system without these additional symbols. However, we believe that their use will give additional scrutability and utility to the risk models.

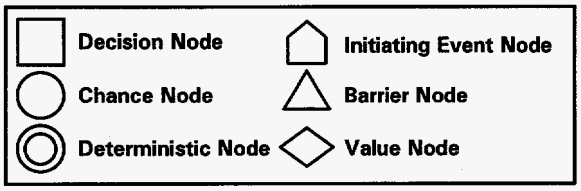

Figure 1. Extended influence diagram symbols.

CONSTRUCTING A SAMPLE RISK MODEL

Consider an application in which a robotic system is to automatically lift and move a large, heavy object. The system is composed of a crane-type hoist system and an automated controller (an "information system"). We assume that there are humans nearby who can intervene in the process should it go awry. However, if we expect them to intervene, we must develop and implement appropriate procedural instructions.

Our risk assessment begins by identifying the consequences and benefits from the proper or improper operation of the system. The system designer wants to use an information system to control the lift activity to reduce the time required to stabilize a swinging load before it can be set down. Potential consequences include lost productivity (setting the load in the wrong place so that it must be moved again), worker injury (the load hits or is set down upon a worker), system unavailability (the system must be operable when it is needed to prevent lost productivity), and, if the system is to move hazardous loads or operate near hazardous areas, public safety and environmental issues. While it is beyond the scope of this paper to develop each of these potential consequences, it is apparent that the more serious consequences are caused by the system moving, setting, or spilling something into an undesired location. This consequence will be used as the basis for the "value" node at the endpoint of our sample influence diagram.

We continue to build the sample risk model by constructing a model of the system's "normal flow of operation" and looking for influences that can cause deviations from that normal flow. Figure 2 shows this process in three steps. For this simple example, we model the system's normal operational procedure as a lift operation, a move operation, and a set-down operation. These operations must occur in sequence, so each operation's node on the diagram is influenced by its predecessor in the normal flow of operation.

The next step is to identify the influences that can cause the system to deviate from the normal flow of operation toward the consequences that we have identified. Step two in Figure 2 shows a number of influences that can cause this to occur, and the aspects of the normal flow of operation that they would affect. Owing to space considerations, this list is by no means complete. Note, however, that the "controller fault" node is not sufficiently detailed for assessment. Thus, we refine this node to incorporate its underlying causes just as we would successively refine the failures in a fault tree. Note also that both "initiating events" (e.g., bad software) and "unconditional random failures" (e.g., controller hardware failure) can influence the system away from its intended mission.

The third step is to incorporate "barriers" into the model. These influences act to reduce the probability that a "bad" influence will produce an undesired result. Step three in Figure 2 shows a number of potential barriers that might reduce system risk. Some barriers are procedural (e.g., preoperation checklists and visual verification of the ongoing process), while others are technological (e.g., a fault-tolerant controller hardware and an automatic position verification tool). The list of barriers should be viewed as providing *design options* that can be implemented individually or in combination.

This risk model provides some general modelling insights. First, a single barrier may affect more than one operation, initiating event, or threat. For example, a preoperation checklist might act both to identify damaged equipment and to ensure that the load is balanced. This is easily represented in the model. It is also possible to use more than one barrier against a single threat. For example, both an automatic position verification tool and visual monitoring of the process can act as barriers to ensure that an improperly entered target location does not cause the load to be taken to the wrong location. Finally, note that this model still contains one "unmitigated threat" in that the initiating event "bad sensors" is not directly mitigated by any barrier in the model (although, depending upon the design, the visual monitoring of the process might help to mitigate the *effect* of a sensor failure). Upon recognizing an unmitigated threat, a designer can either knowingly accept the risk posed by that threat or find a barrier to effectively mitigate it. This is, however, a conscious decision based on an identified risk, rather than a default design based upon ignorance (as might occur without the information provided by an analysis such as the one proposed here).

One of our objectives in designing a risk modelling tool is that it be readily extendable. This sample risk model seems to fit this criterion. The level of detail in this sample model might be appropriate for a high-level scoping study. If a more detailed study were required later, one could, for example, break down the "Controller Fault" node into a greater level of detail. This might identify new fault conditions and initiating events, and provide an opportunity to include new barriers as the system is better understood. We believe that models of this type can be extended to an arbitrary level of detail in much the same way as one would extend a fault tree analysis. This property makes this method a valuable tool for performing many types of risk assessment studies.

ENHANCED METHODS FOR SOLVING INFLUENCE DIAGRAMS

Many past applications of influence diagrams have been in the area of decision theory, where an analyst wants to determine which decision option will lead to the greatest possible utility or the lowest possible consequences. Thus, influence diagrams have been solved by successive simplification of the network using arc

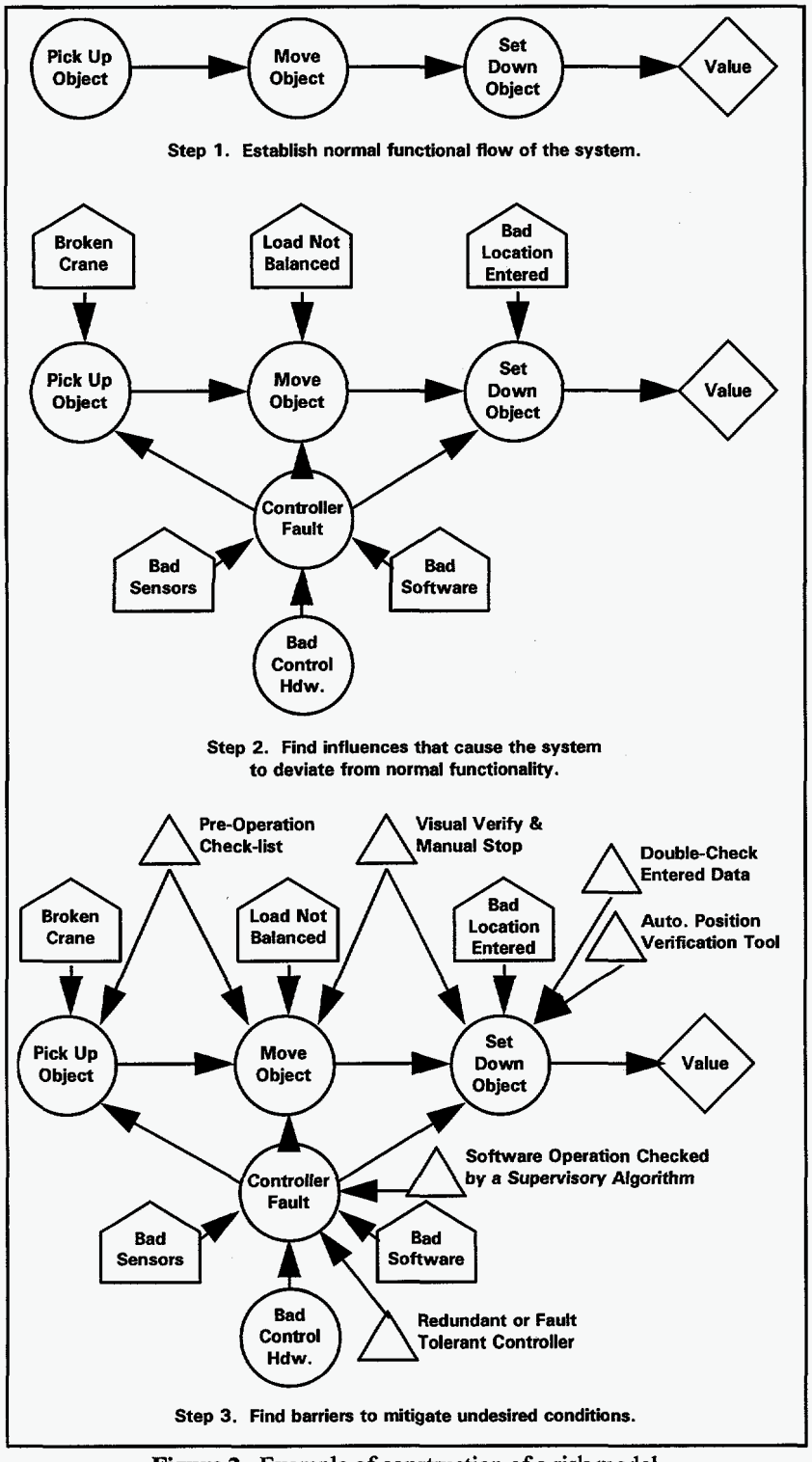

Figure 2. Example of construction of a risk model.

reversal and the node removal based on probabilistic rules (Jae and Apostolakis, 1992; Shachter, 1986). The objective is to obtain a network that consists of only two nodes (one representing the decision and one representing the "value" -- utility or consequences). Under such a solution method, the probabilistic and deterministic information is systematically combined until all that remains is the conditional probability that each possible decision will result in each of the possible outcomes. While this works well for some decision analyses, it provides little guidance, for example, as to where one might be able to improve the system given additional resources.

We believe that a risk assessment should provide not only a quantitative estimate of risk (the reliability of the actual quantitative value is always questioned), but also information such as a list of the system's most likely and riskiest paths; a ranking of the events, nodes, probabilities, and uncertainties that figure most prominently in the risk of the system; and a list of places where improvements in the system will lead to the greatest risk reduction. Our objective is to develop a solution method that allows us to obtain this information from an influence diagram. Similar information is routinely generated in event tree analyses (the list of event tree paths) and in fault tree analyses (the cut set importance measures). Therefore, we have explored the adaptation of fault tree and event tree solution methods to influence diagrams.

Other studies (Jae and Apostolakis, 1992) have demonstrated that it is always possible to translate an influence diagram into an event tree or a decision tree (this is not a one-to-one translation, as there are often many appropriate event tree representations for a given influence diagram). Event trees are solved through an exhaustive consideration of all possible paths, with the removal of physically or logically precluded paths. Solving an influence diagram in this manner would be conceptually very simple and would generate important results that cannot be obtained using traditional solution methods.

The event tree solution method, while an improvement over traditional solution methods, does not generate all of the types of event importance information that we would like to obtain from our risk analysis. At Sandia National Laboratories, we have discussed the potential for generating importance measures for event trees that are similar to those found through a fault tree cut set importance analysis (A.C. Payne, Jr., personal communication). While the details of the method are beyond the scope of this paper, the method can be summarized as follows:

First, consider each event tree path to be equivalent to a cut set from a fault tree analysis. Each event that occurs in that path

(each probability that contributes to the path) can be thought of as a basic event in the cut set. If each event tree question is limited to two possible outcomes, the group of cut sets that represent the event tree paths can legitimately be examined using all traditional cut set importance measures such as the partial derivative, risk increase, risk reduction, and Fussell-Vesely importance measures (Roberts *et al.*, 1981). Traditional cut set uncertainty analysis techniques (e.g., Monte Carlo and Latin hypercube sampling techniques) can also be applied without adaptation (Iman and Shortencarier, 1984, 1986). Therefore, we can extract all of the information we need from an influence diagram if we solve it as an event tree, translate the paths into cut sets, and evaluate them using traditional fault tree cut set importance measures.

The only restriction on this technique is that any event tree that is developed from the influence diagram must be constructed using only two outcomes per question (binary events). This does not represent a *theoretical* limitation on the methodology because other studies have demonstrated that there exists at least one "binary event tree" for each "multibranch event tree." It is, however, a *practical* limitation on the method because we may want to let a single node in the influence diagram assume multiple states. For example, the "Move Object" node in our sample problem might take on the values "Normal Move," "Too High," "Too Low," "Too Fast," and "Quivering." We would like to be able to translate this node into a single event tree question with multiple outcomes, but are prevented from doing so if we want to obtain event importance information. The translation from multibranch events to binary events is difficult and results in an event tree that is far more difficult to understand than the original multibranch tree.

Sandia has conducted some research directed toward developing importance measures for "cut set" expressions involving nonbinary events. This currently unpublished research indicates that importance measures similar to the partial derivative, risk increase, and Fussell-Vesely importance measures can be calculated with only slight modifications in the computational algorithm from their binary event counterparts. If certain minor additional computational assumptions are made, a measure that parallels the risk reduction importance measure can also be calculated for non-binary events. However, the uncertainty and uncertainty importance analyses for a cut set expression for multiple-outcome events are complicated by the fact that the probabilities for all outcomes for a particular event, when summed, must equal one. It has been demonstrated elsewhere that this problem can be solved using the multivariate Dirichlet distribution (Payne and Wyss, 1994).

CONCEPTUAL EXAMPLE COMPUTATION

To perform a risk computation on the influence diagram shown as Step 3 in Figure 2, an analyst would begin by developing an event tree question for each node in the influence diagram. While it is possible to make the first event tree question based upon any independent node (i.e., a node that is not influenced by any other node), a potentially more satisfying method is to start with the value node as the last event tree question and work methodically backward from there to bring in other nodes in a coherent manner. This influence diagram contains a value node, five chance nodes, five initiating event nodes, and six barrier nodes. Thus, the event tree that represents this diagram will have a total of 17 events (questions) -- one for each node in the diagram.

If each node in this diagram were to be considered a binary event, the theoretical maximum number of paths through the event tree would be 2^{17} (or 131,072 paths). Using multiple-outcome events would cause this number to increase rapidly. Clearly no analyst would want to graph this event tree -- even with the assistance of an event tree graphics software tool (Camp and Abeyta, 1991). Sandia's SETAC event tree analysis code suite (Wyss and Daniel, 1994), which is based on the EVNTRE code (Griesmeyer and Smith, 1989) provides an automated nongraphical facility for efficiently processing these large event trees. There is not yet a facility for converting the paths obtained by SETAC into cut sets for the performance of the cut set importance analysis described earlier. The development of new modelling and analysis software is one of the tasks remaining for the research team.

While the theoretical maximum number of paths through an event tree may be large, the actual number of paths realized in the event tree solution may be much smaller. The actual number of paths generated will depend upon such variables as the number of possible outcomes for each node and whether some paths are precluded based on physical arguments. For example, if the crane is broken and cannot pick up the object, it is not possible for the crane to be carrying an unbalanced load. In addition, most event tree analysis tools allow the user to define a truncation probability so that paths of negligible probability can be eliminated. This allows the analyst to concentrate on inferring results from a manageable number of paths.

MODELLING SUPPORT TOOLS

While risk assessment techniques are familiar to many hardware systems analysts, they are less familiar to those who deal in the realm of information system surety (computer security, data integrity, etc.). For this reason, our research team is seeking to provide a number of tools that will help persons who are not familiar with risk assessment techniques quickly become both comfortable and productive risk analysts.

It would be difficult to successfully use these modelling techniques without appropriate software. For this reason, we are developing graphical software that will allow a user to develop an influence diagram risk model, enter condition and quantification data, and perform event tree-based solution and importance calculations. The extent to which this vision can be fulfilled depends upon future laboratory funding and research priorities.

We are also developing tools to guide the thought process of a risk analyst in developing a model. Our objective is to help analysts ensure that their risk assessment does not leave out things that might be important to risk. We are developing a risk taxonomy to help the analyst evaluate the universe of consequences that might occur as a result of the normal or abnormal operation of the system. We are also developing a

"requirements taxonomy" to help those who write software requirements to consider how requirements that are written to solve a problem in one area (e.g., security) might inadvertently cause a problem in another (e.g., availability or functionality). These are implemented through a "risk identification matrix" and a "risk mitigation matrix" (Jansma *et al.*, 1995). The taxonomy and the matrix formalism encourage exhaustive consideration of how each surety objective affects all aspects of proper system operation. The risk mitigation matrix helps the analyst to make an initial determination of whether each potential risk has been adequately mitigated in the system design before the influence diagram model is constructed. Note that these tools force the analyst to examine the system with respect to its own design requirements instead of a preconceived checklist.

The methodology we have proposed considers the effects that barriers have in mitigating threats. Of course, the effectiveness of a barrier will vary depending upon the nature of the individual threat and the characteristics of the barrier. As a part of our research we are attempting to provide some level of guidance to help the analyst characterize both the strengths of potential threats and the strengths of barriers against these threats. The nature of this work is beyond the scope of this paper and will be published in other forums.

SUMMARY AND CONCLUSIONS

We have proposed a risk-based methodology to assess the "surety" of an information system, which is defined as ensuring the "correct" operation of an information system by incorporating appropriate levels of safety, functionality, confidentiality, availability, and integrity. The methodology is based upon an extension of the influence diagram formalism and a new solution technique that allows for the computation of traditional fault tree cut set importance measures from influence diagrams. The modelling technique is straightforward and intuitive, and was demonstrated in a simple sample problem. The computational solution method was outlined but not demonstrated owing to space considerations.

This research has shown that influence diagrams can be successfully applied to information system surety problems. With the extended symbology that we have proposed, the diagram itself is useful to identify unmitigated threats and determine which barriers might be appropriate to help mitigate particular threats (either active or passive). We have also shown that it is possible to generate risk importance information in the solution process that can be used to help identify the major contributors to risk and the best candidates for additional risk mitigation attention. We believe that these techniques can be used to make important contributions to risk management activities in the years ahead.

This paper represents a report of work in progress by an interdisciplinary team at Sandia National Laboratories operating under the Laboratory-Directed Research and Development program. The modelling methods described in this paper represent one facet of the work of this team. The other aspects of this work that were briefly described in the paper will be made available as this study approaches completion.

ACKNOWLEDGMENTS

The authors owe a deep debt of gratitude to Dr. Arthur C. Payne, Jr., of Sandia National Laboratories for his pioneering work in the area of applying cut set importance analysis techniques to gain insight into the results of event tree analyses. This work was funded by the Laboratory-Directed Research and Development program at Sandia National Laboratories.

REFERENCES

Camp, A.L., and Abeyta, L.P., 1991, "SANET 1.0 User's Guide and Reference Manual," SAND91-2864, Sandia National Laboratories, Albuquerque, New Mexico.

Griesmeyer, J.M., and Smith, L.N., 1989, "A Reference Manual for the Event Progression Analysis Code (EVNTRE)," NUREG/CR-5174, Prepared by Sandia National Laboratories for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Howard, R.A., and Matheson, J.E., 1984, "Influence Diagrams," Readings in the Principles and Applications of Decision Analysis, p. 721, edited by R.A. Howard and J.E. Matheson, Strategic Decision Group, Menlo Park, CA.

Iman, R.L., and Shortencarier, M.J., 1984, "A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use With Computer Models," NUREG/CR-3624, Prepared by Sandia National Laboratories for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Iman, R.L., and Shortencarier, M.J., 1986, "A User's Guide for the Top Event Matrix Analysis Code (TEMAC)," NUREG/CR-4598, Prepared by Sandia National Laboratories for the U.S. Nuclear Regulatory Commission, Washington, D.C.

Jae, M., and Apostolakis, G.E., 1992, "The Use of Influence Diagrams for Evaluating Severe Accident Management Strategies," *Nuclear Technology*, Vol. 99, pp. 142-157.

Jae, M., Milici, A.D., Kastenberg, W.E., and Apostolakis, G.E., 1993, "Sensitivity and Uncertainty Analysis of Accident Management Strategies Involving Multiple Decisions," *Nuclear Technology*, Vol. 104, pp. 13-36.

Jansma, R.M., Fletcher, S.K., Halbgewachs, R.D., Lim, J.J., Murphy, M., Sands, P.D., and Wyss, G.D., 1995, "Risk-Based Assessment of the Surety of Information Systems," presented at the 11th International Symposium on the Creation of Electronic Health Record Systems and Global Conference on Patient Cards, Medical Records Institute, Orlando, Florida, March 14-19.

Payne, A.C., Jr., and Wyss, G.D., 1994, "Coherent Sampling of Multiple Branch Event Tree Questions," paper presented at PSAM-II, San Diego, CA.

Roberts, N.H., Vesely, W.E., Haasl, D.F., and Goldberg, F.F., 1981, "Fault Tree Handbook," NUREG-0492, U.S. Nuclear Regulatory Commission, Washington, D.C..

Shachter, R.D., 1986, "Evaluating Influence Diagrams," *Operations Research*, Vol. 34, p. 871.

Wyss, G.D., and Daniel, S.L., 1994, "Recent Enhancements to Probabilistic Risk Assessment Software at Sandia National Laboratories," paper presented at the DOE EFCOG Integrated Risk Management Workshop, Albuquerque, New Mexico.