SDPS-2010
Printed in the United States of America, June, 2010
©2010 Society for Design and Process Science

MODELING OF LARGE-SCALE DISASTER-TOLERANT SYSTEMS

Theodore W. Manikas
Department of Computer Science and Engineering
High Assurance Computing and Networking Laboratories
Southern Methodist University
Dallas, Texas 75272-0122, USA

ABSTRACT

Disaster tolerance in computing and communications systems refers to the ability to maintain a degree of functionality throughout the occurrence of a disaster. We accomplish the incorporation of disaster tolerance within a system by simulating various threats to the system operation and identifying areas for system redesign. However, there are two limitations that need to be addressed. First, many systems are too large to be simulated in a time effective manner. Second, the current fault and attack tree models used to represent system threats are limited in scope, and do not effectively model disaster effects on a system. We address the first limitation by implementing axiomatic analysis to decompose a large system into smaller independent subsystems that can be simulated in a time effective manner. We address the second limitation by developing the cyber threat tree models, which expands upon the current tree models to provide a better representation of disaster effects. This paper describes these methods and shows examples of their applications.

NOMENCLATURE

p = Radix

INTRODUCTION

Disaster-Tolerant Systems

Recent events have demonstrated our vulnerability to disasters, both natural and man-made. This motivates the need to incorporate disaster tolerance into large system designs. Disaster tolerance is a superset of the more established approaches commonly referred to as fault tolerance. Models for disaster tolerance differ from those for fault tolerance since they assume that failures can occur due to massive numbers of individual faults occurring simultaneously or in a rapidly cascading manner as well as single points of failure. Therefore, a disaster-tolerant system can withstand a catastrophic failure and still function with some degree of normality (Szygenda and Thornton, 2005; Harper et al., 2005).

A significant obstacle in developing disaster-tolerant systems is the inability to model very large systems in a tractable amount of time with a suitable degree of detail. To address this problem, it is necessary to decompose large systems into smaller subsystems that can be modeled relatively independently, and then apply superposition principles to these subsystems to derive the total system behavior. We use concepts motivated by the Axiomatic Design approach (Suh, 2001) to develop the new Axiomatic Analysis approach in order to perform this system decomposition and re-connection. The motivation for developing and using an axiomatic analysis approach for large system decomposition is that we wish to perform decomposition while maximizing the property of subsystem independence in order to avoid the problem of masking failure modes due to subsystem interdependence.

In analyzing the effect of large system threats, there is a need to efficiently catalog those threats so that further analyses can be performed to extract common characteristics among the threats and to devise suitable countermeasures. A very large system can have an enormous number of potential threat scenarios and a simple list of these is insufficient for analysis and classification. To model system threats that may occur during disasters, we have developed the Cyber Threat Tree structure, which is a superset of the current fault tree and attack tree models. The Cyber Threat Tree structure is applied to the individual subsystems to model the effects of possible disasters, both natural and man-made, that would affect system performance.

Axiomatic Analysis

Axiomatic Design (AD) is a structured approach that has evolved from the technology of design (Suh, 2001). An axiomatic design approach would be highly desirable for the specification and implementation of large disaster tolerant systems in order to reduce the number of subsystem interdependencies that can lead to non-obvious cascading failures resulting in a disaster. Unfortunately it is impractical to employ the AD approach for most large network systems. These systems often evolve over time, which makes it impossible to formulate all of the system requirements before implementation. As a result, these systems may have unanticipated subsystem

interdependencies that degrade overall robustness. Therefore, we propose the use of a related but inverse process to AD that we refer to as Axiomatic Analysis (AA). With the AA approach, an existing large system is decomposed based on the axioms similar to those used in the AD approach (Mullens et al., 2005). Each subsystem is then small enough to be simulated in a time effective manner so that analysis can be performed and redundancy can be included only where needed. The axiom of subsystem independence allows the aforementioned unanticipated subsystem interdependencies to be uncovered. At this point, intelligent decomposition can occur and areas where redundancy should be added to enhance disaster tolerance can be identified.

Cyber Threat Trees

Classical fault tree analysis (Vesely et al., 1981) was developed to represent system failures that may result from component or subsystem failures. This approach uses Boolean logic operations to represent how combinations of these failures could lead to a system failure. Fault trees are represented as networks of Boolean logic operators where a fault is considered to either have occurred or not occurred.

Attack trees (Schneier, 1999) are similar to fault trees. However, they focus on system security and are an enumeration of possible attacks. The root of an attack tree represents a successful attack and the leaf nodes represent ways of achieving the planned attack. Similar to fault trees, attack trees also rely on binary-valued algebras.

Our new structure, the Cyber Threat Tree, is a superset of fault and attack trees. Cyber threat trees are based on multiple-valued or radix-p valued algebras over a finite and discrete set of values. When the radix p=2, the cyber threat tree reduces to a fault or attack tree depending on the nature of the disruptive events. However, modeling different operational modes other than just the binary case of failure or normal operation are critical in analyzing large systems in the presence of threats. The cyber threat tree structure allows for the modeling of partial system failures, which is essential for disaster-tolerant system design.

Objective

The objective of this paper is to describe the development and application of axiomatic analysis methods to large-scale system decomposition, and the development and application of cyber threat tree models to simulate the effect of possible disasters on the system. These methods and models will be used to assess the disaster tolerance of a given system.

AXIOMATIC ANALYSIS (AA)

Figure 1 illustrates an overview of the AA approach. A large system is decomposed into smaller subsystems, where each subsystem can be independently simulated. The resultant behavioral models of each subsystem are then combined to form the behavioral model of the original system.

In order to evaluate the effectiveness of the AA approach for large system simulation, we chose an example system that was small enough such that the entire system could be simulated. Next, we applied AA to the example system, and then compared the results of entire system approach with the AA approach. Our example system is a data communications network, which we will call our test network system (Figure 2). The network contains 25 servers and over 500 terminals. Connections are handled by five routers and 23 switches, and the network is spread across four buildings.

To decompose the test system into subsystems, a matrix is formed that relates the chosen measured metrics among the system components. This is a weighted adjacency matrix for a graph whose vertices represent system components and edges are weighted by the identified metric. Figure 3 shows a system matrix example. An "X" denotes the direct connections between the individual network components while the "∞" is used to indicate the connection between the component and itself. The system matrix is then decomposed allowing for the identification of relatively independent subsystems with minimal information transfer among the subsystems.

System Matrix Permutation

The first step in axiomatic analysis is to permute the system matrix. Methods for permuting matrices have been well defined in previous work (Easton et al., 2007; Karypis and Kumar, 1998) and those approaches are used here. The system matrix is permuted to attempt to transform it into a lower triangular matrix to be used for decomposition. Lower triangular forms are desirable since they automatically expose subsystem independence, which is an important part of the AA process. In a perfectly independent subsystem, the permuted adjacency matrix would take on the form of the identity matrix indicating all components are totally independent. Triangular forms show the relative independence of large system components. For large systems, the matrix will become too large to represent explicitly and will be sparse in that it will not be possible to obtain measurements of all metrics. However, transforming the system matrices into a form that is close to triangular is beneficial for exposing system component independencies.

System Matrix Decomposition

After the matrix has been permuted, blocks are found in the matrix to determine the subsystem boundaries and their components. Finer grained decompositions are individually faster to simulate and require fewer computational resources, but overall system response may not be as accurate due to interdependencies among the

subsystems that are ignored. Therefore, there may be tradeoffs in determining subsystems sizes for a given system.

For our test network system, the sorted matrix was reduced in size by summing matrix elements within a square window to generate a new element in the reduced matrix. This approach is useful for large matrices, as it decreases the number of elements to be examined. For our test system, an 8 x 8 window was used to reduce the matrix size. When the window operations had been performed on the matrix, the matrix was decomposed on the window boundaries. This method results in the system decomposition into four partitions as shown in Figure 4.

Subsystem Simulation and Total System Reconstruction

After the subsystems have been identified, each subsystem is simulated independently, either sequentially or in parallel. Next, the simulation results are combined to form the total system behavior. If the subsystems are completely independent, the principle of superposition can be applied, which means that the total system response is the arithmetic sum of the subsystem simulation responses. Since there will be some degree of subsystem interdependence, a weighted sum of subsystem responses or other adaptive approaches can be employed in this step allowing for the entire system response to be realized as a linear combination of decomposed subsystem responses.

CYBER THREAT TREES

During the subsystem simulation, cyber threat trees are used to model possible system threats. Recall that the cyber threat tree is based on Multiple-Valued Logic (MVL) algebra, as opposed to the binary logic used by traditional fault and attack trees. Therefore, the logical OR function expands to a MAX function, where MAX(x,y,z) = maximum value of {x,y,z}. Similarly, the logical AND function expands to a MIN function, where MIN(x,y,z) = minimum value of {x,y,z}. The literal selection-gate, denoted as J_i, is a unary operation whose output is 0 if the input logic value is not *i* and the output is the maximum logic value (in this case "2" for 3-valued logic) when the input is value *i* (Miller and Thornton, 2008).

Decision Diagrams

There are likely to be many potential threats present in the large distributed systems of interest, thus the cyber threat tree structure becomes unwieldy to manipulate due to its large size. Thus, decision diagrams, which are rooted directed acyclic graphs, are applied to represent cyber threat trees. For binary systems such as fault trees, the binary decision diagram (BDD) structure is often used (Remenyte and Andrews, 2006). For multiple-valued logic systems such as our cyber threat tree, the BDD structure is extended to a multiple-valued decision diagram (MDD) (Drechsler et al., 2000; Miller and Drechsler, 1998).

Figure 5 shows an example of a three-valued (p=3) MDD representing g, a MIN function of two variables x1 and x2, where the logic gate diagram is shown on the left and the corresponding MDD is on the right, the output value can be derived by traversing the graph. Table 1 shows the truth table for function g.

RESULTS AND DISCUSSIONS

Axiomatic Analysis

We used the OPNET network simulation modeling tool for our test system (www.opnet.com). The amount of traffic (bandwidth) between adjacent nodes in our test system was compared to determine how accurately the decomposed subsystems modeled the entire system (Table 2). The results indicate that the bandwidth error data for the subsystems are less than 0.1% when compared to the full system simulation. The error data collected indicate that decomposing a large system based on axiomatic principles can allow accurate modeling and simulation of very large systems.

We discovered a significant time advantage for using the axiomatic analysis approach versus full system simulation. The full system simulation took about 17 hours to run on OPNET, while the total time to simulate all subsystems was about 12 hours. The run times for the decomposed systems could be further improved by running the subsystem simulations in parallel, as opposed to in series. Further details can be found in (Spenner et al., 2010).

Cyber Threat Trees

Figure 6 shows a cyber threat tree decision diagram developed for an example system, which is power grid connected to three power generation plants: coal, hydro, and wind. There are three possible states of operation for this system: 2 = fully operational, 1 = degraded operation, 0 = non-operational. Therefore, this is a radix-3 MVL system, and the decision diagram is an MDD. The corresponding truth table is shown in Table 3, where "X" indicates "don't-care".

For this system, we observe that if the coal plant is non-operational, the power grid system will not be fully operational. However, if the coal plant is fully operational, then the power grid system will be fully operationally if both the hydro and wind plants are at least partially operational. If this system were modeled by the traditional binary fault tree model, then the second condition above would be identified as completely non-operational. Thus, the cyber threat tree structure can better model the system behavior when system components are partially operational. Further details can be found in (Ongsakorn et al., 2010).

CONCLUSION

The axiomatic analysis results for our test system show that the total simulation of decomposed subsystems is faster than a full system simulation, with minimal error in bandwidth results. These results indicate that the axiomatic analysis approach is likely to be effective for larger systems that cannot be simulated in their entirety with existing simulation tools.

Similarly, the cyber threat tree structure is likely to be a more accurate model of system operations in the presence of failures, when compared to the current fault and attack tree models. Recall that the goal of disaster-tolerant systems is to be able to function in the presence of component failures. Since the cyber threat tree can model partially operationally systems, it is suited for disaster-tolerant systems analysis.

ACKNOWLEDGEMENT

Funding for this research was provided by the Office of Naval Research (ONR) project N000140910784.

REFERENCES

Drechsler, R., Thornton, M.A. and Wessels, D., 2000, "MDD-based synthesis of multi-valued logic networks," *Proceedings of the 30th IEEE International Symposium on Multiple-valued Logic*, pp. 41-46.

Easton, D., Thornton, M.A. and Nair, V.S.S., 2007, "Axiomatic Design Process for Disaster Tolerance," *Proceedings of the 11th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI)*,

Harper, M. A., Lawler, C. M. and Thornton, M. A. , 2005, "IT Application Downtime, Executive Visibility and Disaster Tolerant Computing". *Proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2005)*, and *International Conference on Information Systems Analysis and Synthesis (ISAS)*, pp. 165-170.

Karypis, G. and Kumar, V., 1998, "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs," *Society for Industrial and Applied Mathematics Journal of Scientific Computing*, vol. 20, No. 1, pp. 359-392.

Miller, D.M. and Drechsler, R., 1998, "Implementing a multiple-valued decision diagram package," *Proceedings of the 28th IEEE International Symposium on Multiple-valued Logic*, pp. 52-57.

Miller, D.M. and Thornton, M.A., 2008, "Multiple-Valued Logic: Concepts and Representations", Morgan & Claypool Publishers.

Mullens, M.A., Mohammed, A., Armacost, R.L., Gawlik, T.A. and Hoekstra, R.L., 2005, "Axiomatic Based Decomposition for Conceptual Product Design". *Production and Operations Management*. Vol. 14, pp. 286-300.

Ongsakorn, P., Turney, K., Thornton, M., Nair, S. and Manikas, T., 2010, "Cyber Threat Trees for Large System Threat Cataloging and Analysis", *IEEE Int. Systems Conference*.

Remenyte, R. and Andrews, J.D., 2006 "A simple component connection approach for fault tree conversion to binary decision diagram", *Proceedings of the First International Conference on Availability, Reliability and Security*.

Schneier, B., 1999, "Attack Trees: Modeling Security Threats," *Dr. Dobb's Journal*, Dec.

Spenner, L., Krier, P., Thornton, M., Nair, S. and Manikas, T., 2010, "Large System Decomposition and Simulation Methodology Using Axiomatic Analysis", *IEEE Int. Systems Conference*.

Suh, N. P., 2001, "Axiomatic Design: Advances and Applications", Oxford University Press.

Szygenda, S.A. and Thornton, M.A., 2005, "Disaster Tolerant Computing and Communications". *Proceedings of the International Conference on Cybernetics and Information Technologies, Systems and Applications (CITSA 2005)*, and *International Conference on Information Systems Analysis and Synthesis (ISAS)*, pp. 171-173.

Vesely, W.E., Goldberg, F.F., Roberts, N.H. and Haasl, D.F., 1981, "Fault tree handbook," NUREG-0492, U.S. Nuclear Regulatory Commission, Jan.

FIGURES AND TABLES

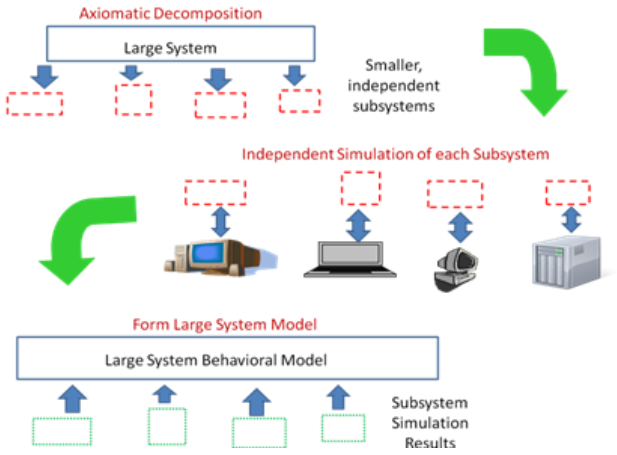

Fig. 1 Axiomatic analysis approach

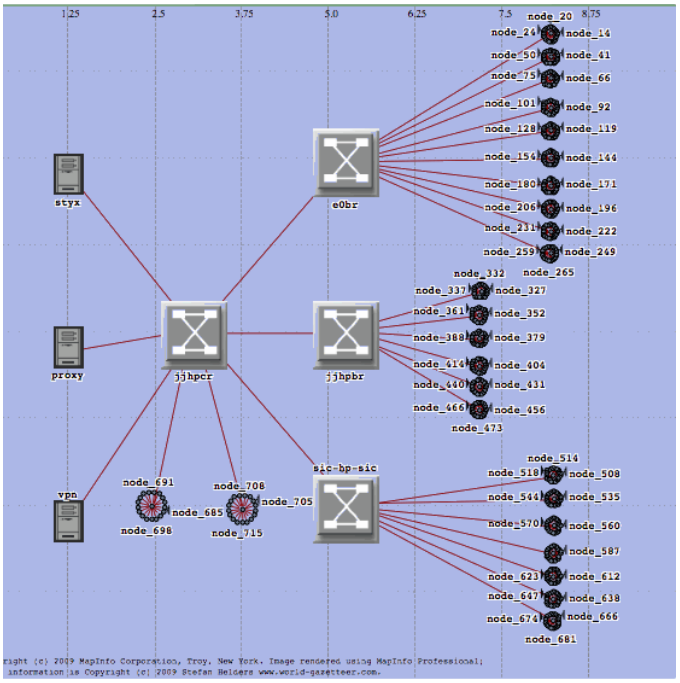

Fig. 2 Test network system

	sw1	ser1	ser2	ser3	sw2	wks1	wks2	wks3
sw1	∞	X	X	X	X	0	0	0
ser1	X	∞	0	0	0	0	0	0
ser2	X	0	∞	0	0	0	0	0
ser3	X	0	0	∞	0	0	0	0
sw2	X	0	0	0	∞	X	X	X
wks1	0	0	0	0	X	∞	0	0
wks2	0	0	0	0	X	0	∞	0
wks3	0	0	0	0	X	0	0	∞

Fig. 3 Example of system matrix

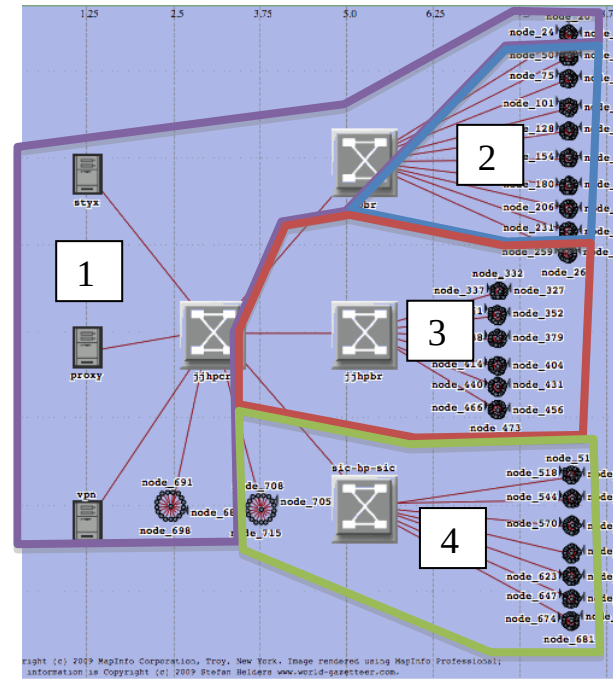

Fig. 4 Test network system with 8 x 8 window decomposition

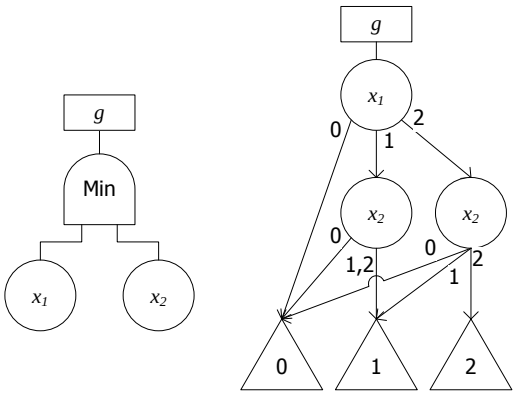

Fig. 5 Example of an MDD for 3-valued logic (MIN function)

Table 1 Truth table for 3-valued MIN function

x_1	x_2	**g**
0	0	**0**
0	1	**0**
0	2	**0**
1	0	**0**
1	1	**1**
1	2	**1**
2	0	**0**
2	1	**1**
2	2	**2**

Table 2 Average bandwidth error

Partition Number	Average Error (kbit/s)
1	2.3081
2	2.4004
3	2.1045
4	2.1406

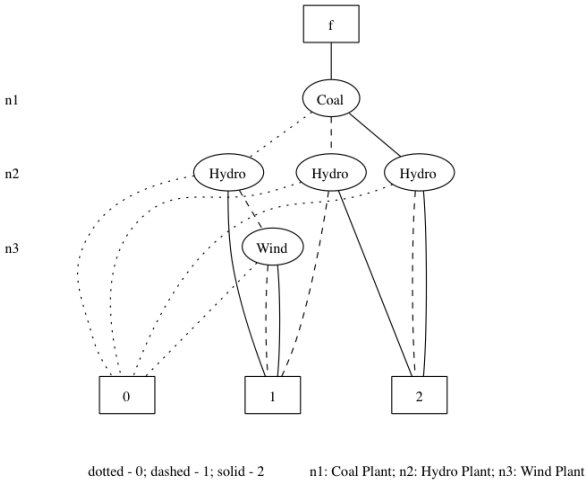

Fig. 6 Cyber threat tree decision diagram for an example power grid system

Table 3 Truth table for Figure 6

Coal	Hydro	Wind	**f**
0	0	X	**0**
0	1	0	**0**
0	1	1	**1**
0	1	2	**1**
0	2	X	**1**
1	0	X	**0**
1	1	X	**1**
1	2	X	**2**
2	0	X	**0**
2	1	X	**2**
2	2	X	**2**