An Axiomatic Analysis Approach for Large-Scale Disaster-Tolerant Systems Modeling[†]

Theodore W. Manikas, Laura L. Spenner, Paul D. Krier, Mitchell A. Thornton,
Sukumaran Nair, and Stephen A. Szygenda
Department of Computer Science and Engineering
High Assurance Computing and Networking Laboratories
Southern Methodist University
Dallas, Texas, U.S.A.

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. Unfortunately, many systems are too large to be simulated in a time effective manner. To address this limitation, an axiomatic approach that decomposes a large-scale system into smaller subsystems is developed that allows the subsystems to be independently modeled. This approach is implemented using a data communications network system example. The results indicate that the decomposition approach produces simulation responses that are similar to the full system approach, but with greatly reduced simulation time.

Keywords: Axiomatic Design, Disaster Tolerance, Modeling, Computer Networks, Cyber Security

1. INTRODUCTION

Events such as hurricanes Katrina and Rita, the 911 event, the attack on the USS Cole, and the Northeast U.S. blackout have demonstrated our vulnerability to disasters and motivated our need to find methods that provide some degree of tolerance for large cyberspace systems in the presence of disasters. Currently, the area of disaster tolerance [1] is a relatively immature research area as compared to the related areas of fault tolerance and disaster recovery. Hence, there is a real need to investigate the problem of disaster tolerance from a scientific and engineering point of view so that more effective and economical approaches can be developed.

One of the biggest obstacles hindering research in this area is the inability to model very large systems in a tractable amount of time with a suitable degree of detail, particularly integrated software/ hardware/networked systems. Such modeling must be accomplished so that system behavior can be obtained both in a normal operating mode and in the presence of a disaster. The sheer size of many practical systems has made the modeling of the total system as a single entity impossible. As it is necessary that large systems be decomposed into smaller subsystems that can be modeled independently, and then combined using superposition principles to derive the total system behavior, we use concepts motivated by the Axiomatic Design approach [2] 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.

2. DISASTER TOLERANCE

As described in [3, 4] Disaster Tolerance in computing and communications systems refers to the ability of infrastructure, software, IT systems, communications infrastructure, and business or organizational processes that depend on these systems, to maintain functionality throughout the occurrence of a disaster. The goal of Disaster Tolerance is to provide an ability to continue uninterrupted operations, despite the occurrence of a disaster that would normally interrupt organizational operations; where critical business functions and technologies continue operations, as opposed to

[†] This research supported by ONR project N000140910784

resuming them as is the common approach in disaster recovery.

Disaster tolerance is a superset of the more established approaches commonly referred to as fault tolerance in that a disaster may occur which causes rapid, almost simultaneous, multiple points of failure in a system that can escalate into wide catastrophic system failures. Most traditional fault tolerance approaches contain assumptions about individual component failure rate distributions that are usually considered to have a large degree of statistical independence. 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 either simultaneously or in a rapidly cascading manner as well as single points of failure. In other words, disaster tolerance is the characteristic attributed to a system that can withstand a catastrophic failure and still function with some degree of normality [1, 5].

3. AXIOMATIC DESIGN AND ANALYSIS

Axiomatic Design (AD) is a structured approach that has evolved from the technology of design. It infiltrates scientific principles into the design process in order to improve design activities [2]. As with any system design process methodology, the same steps are required: understanding customer needs; defining the problem needed to be solved to meet the needs; creating/selecting a solution; analyzing/optimizing the proposed solution; and checking the design against the stakeholder needs. The axiomatic design approach is a systematic method for capturing user requirements and transforming them into relatively independent design parameters. As described in detail in [2], the AD approach involves formulating a design matrix allowing different subsystems to be parameterized and simulated separately.

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. In the terminology of AD, the design process is envisioned as being composed of mappings among different domains. Initially, customer needs are formulated in the “customer domain” that are then mapped to the “functional domain”, followed by a mapping to the “physical domain”, and ultimately a mapping to the “process domain”. These design domains can vary depending on the system of interest [6, 7].

Unfortunately it is impractical to employ the AD approach for most large systems such as the Internet or the US electric power grid since these systems evolve over time and it is impossible to formulate all system requirements before implementation. The result is that many large systems have hidden or unanticipated subsystem interdependencies that degrade overall robustness. A good example of this effect is the power grid blackout in the Northeastern US that occurred in 2003 [8]. This large scale blackout was ultimately determined to be caused by a fault event that occurred in another state in the US that triggered a series of cascading failures that ultimately resulted in the disaster. For these reasons, 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 [9]. 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 will allow unanticipated subsystem interdependencies that occurred due to the evolution of the overall system topology to be uncovered. Once the interdependencies are uncovered, intelligent decomposition can occur and points where redundancy should be added to enhance disaster tolerance can be identified.

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.

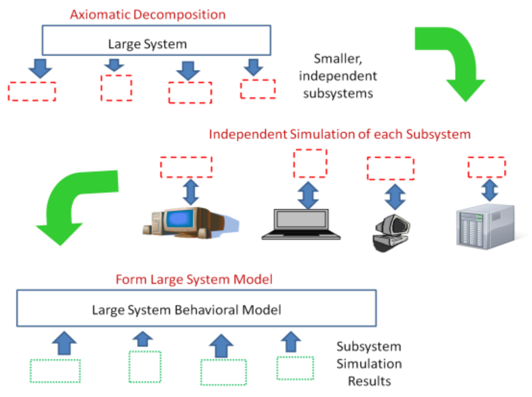
Axiomatic Decomposition
Large System
Smaller, independent subsystems
Independent Simulation of each Subsystem
Form Large System Model
Large System Behavioral Model
Subsystem Simulation Results

Figure 1. Axiomatic Analysis Approach

The tasks used to develop and validate our axiomatic analysis approach are illustrated in the flow chart in Figure 2. The results presented are obtained from phase 1 of the project that focuses on large system decomposition and redesign for disaster tolerance enhancement.

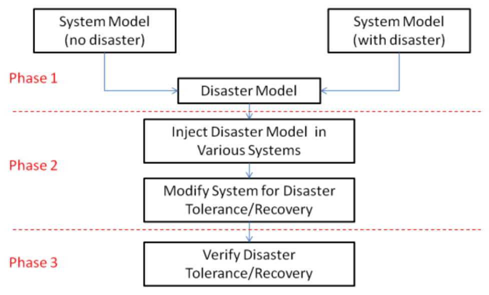
System Model (no disaster)
System Model (with disaster)
Phase 1
Disaster Model
Inject Disaster Model in Various Systems
Phase 2
Modify System for Disaster Tolerance/Recovery
Phase 3
Verify Disaster Tolerance/Recovery

Figure 2. Design and Validation Methodology for Axiomatic Analysis Approach

4. AXIOMATIC ANALYSIS APPLIED TO OPNET SYSTEM MODELS

Our current approach utilizes existing modeling tools for the decomposed portions of the large system followed by employing the principle of superposition to combine the subsystem modeling responses to obtain the response for the entire subsystem. Superposition is only valid when the decomposed subsystems are linearly independent. While we do not expect to decompose the large system with exact independence, our approach uses our newly developed axiomatic analysis technique to decompose large systems into relatively independent subsystems. To validate this methodology, we first model an entire system (this initial system must be small enough that it can be modeled in its entirety), then we decompose this system and model each of the subsystems independently and infer total system response through combining subsystem modeling results. By comparing the results of modeling the system in its entirety with the response obtained from the AA-based decomposition, validation of the methodology can be achieved.

We have chosen an example system and commercial tools to be used for the validation of our decomposition approaches; a university data communications network and the commercial tool, OPNET. The example communications network was chosen because it is small enough that it can be modeled as a whole using OPNET while also being large enough that various decomposition methods can be evaluated. The OPNET software package (http://www.opnet.com) allows users to develop solutions for issues related to application performance management; network planning, engineering, and operations; and network research, and development. Using the OPNET Modeler software, simulated network scenarios can be created in order to collect data regarding the operation of the network. As with most modeling programs, the runtime of OPNET increases dramatically for extremely large systems due to computational constraints. Therefore, we are using OPNET to validate our AA approach by simulating the subsystems of the decomposed large system resulting in overall decreased simulation runtime.

The example system used as test case for the validation of this approach consists of approximately 25 servers and more than 500 terminals. Network connections are processed by five routers and 23 switches. All of these components are geographically spread across four buildings. A model for this network has been created using the OPNET software package and is shown in Figure 3. In the figure, "styx" and "proxy" are servers, "e0br" and "jjhpbr" are switches, and "jj0hp2" and "fnode_38" are workstation clusters.

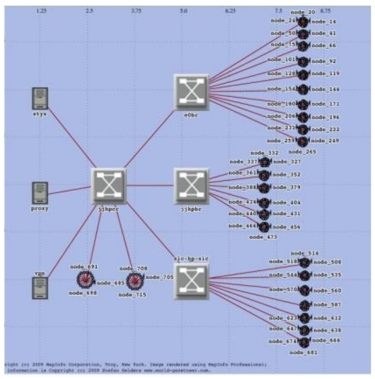

Figure 3. OPNET Representation of the Test Network

5. RESULTS

We performed the following tasks to implement the axiomatic analysis approach on our test system. First, we ran an OPNET simulation on the entire system, examining the transmission bandwidths between all components. The simulation of the overall system required 18 hours of real time for the simulation of 24 hours of network operation. This overall system simulation was necessary to obtain a baseline result to compare to the decomposed system simulation results.

The switches e0br, jjhpbr, sic-hp-sic each connect to independent sets of workstation clusters. Since the axiomatic analysis approach requires decomposing the main system into independent subsystems, we selected each switch and its associated workstation clusters as a subsystem. We ran an OPNET simulation for each subsystem, simulating 24 hours of network operation as per the baseline simulation and accumulated measures of bandwidth among the components for axiomatic analysis. Table 1 shows the wall-clock simulation times – note that the total simulation time for the subsystems (10 hours) is less than the time for an entire system simulation.

Table 1. OPNET Simulation Times for Test Network

System	Hours
Entire system	18
e0br	5
jjhpbr	2
sic-hp-sic	3

For axiomatic analysis, if the subsystems are completely independent, then the summation of the bandwidth results from each subsystem should equal the bandwidth results of the original system. Tables 2 and 3 show the bandwidth error (absolute difference between the subsystem total and full system simulation results). The bandwidth units are Kbps (Kilobits per second).

In Table 2, we note that the row for switch jjhpcr has significant bandwidth errors for communication between the servers. This is to be expected, since the three subsystems all communicate with the servers through switch jjhpcr, there is interdependence between the subsystems. However, for communication between switch e0br and its workstation clusters, the bandwidth errors are very small, which indicates that subsystem e0br is essentially independent from the other subsystems. We can see similar results in Table 3 for the bandwidth results for switch jjhpbr and sic-hp-sic.

6. CONCLUSION

The results of Axiomatic Analysis on our test system indicate that we can decompose a large system into smaller, independent subsystems, that can be simulated independently and then reconstruct the total system response by combining the results back together. This approach results in a reduction of the total simulation time without significantly affecting the total system simulation results. The ability to decompose large-scale systems is especially important when simulating the effects of disaster conditions on these systems, thus our approach has potential applications for the analysis of system disaster tolerance, which will be the next phase of our research.

REFERENCES

[1] S.A. Szygenda and M.A. Thornton, "Disaster Tolerant Computing and Communications", in **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)**, July 14-17, 2005, pp. 171-173.

[2] N. P. Suh, **Axiomatic Design: Advances and Applications**, Oxford University Press, Oxford Series on Advanced Manufacturing, New York, New York, ISBN 0-19-513466-4, May 2001.

[3] C. M. Lawler and S. A. Szygenda, "Challenges and Realities of Disaster Recovery: Percieved Business Value, Executive Visibility & Capital Investment", **International Journal of Business Information Systems (IJBIS)**, 1746-0980, Issue 4, Volume 3, 2008.

[4] D. Easton, M.A. Thornton, V.S.S. Nair, S.A. Szygenda, "A Methodology for Disaster Tolerance Utilizing the Concepts of Axiomatic Design", **IIIS Journal of Systemics, Cybernetics and Informatics**, vol. 6, no. 4, 2008.

[5] M. A. Harper, C. M. Lawler, and M. A. Thornton, "IT Application Downtime, Executive Visibility and Disaster Tolerant Computing", in **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)**, July 14-17, 2005, pp. 165-170.

[6] B. Gumus and A. Ertas, "Requirement Management and Axiomatic Design". **Journal of Integrated Design & Process Science**. 2004. 8(4): p. 19-31.

[7] C. Togay, A.H. Dogru, and J.U. Tanik, "Systematic Component-Oriented Development with Axiomatic Design". **Journal of Systems and Software**. 2008. 81(11): p. 1803-1815.

[8] Goldreich, S. "Northeast Blackout Ups Pressure For Electricity Reliability Standards." **CQ Weekly** 61.34 (06 Sep. 2003): 2151-2152.

[9] M.A. Mullens, A. Mohammed, R.L. Armacost, T.A. Gawlik, and R.L. Hoekstra, "Axiomatic Based Decomposition for Conceptual Product Design". **Production and Operations Management.** 2005. 14(3): p. 286-300.

Table 2. Absolute Error Between Subsystem and Full System Results for Servers and Switch *e0br*

	jjserver	jjserver_0	styx	proxy	vpn	jjhpcr	e0br	e0s105a	e0s130a	e1s105a	e1s129a	e1s129b	e3s105a	e3s105b	e3s105c	e3s129a	e3s130a
jjserver	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jjserver_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
styx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
proxy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
vpn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jjhpcr	51	51	51	51	51	0	130	0	0	0	0	0	0	0	0	0	0
e0br	0	0	0	0	0	106	0	6	6	6	6	6	6	6	6	6	6
e0s105a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e0s130a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e1s105a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e1s129a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e1s129b	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e3s105a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e3s105b	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e3s105c	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e3s129a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
e3s130a	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Table 3. Absolute Error Between Subsystem and Full System Results for Switches *jjhpbr* and *sic-hp-sic*:

	jjhpbr	jj0hp2	jj1hp1	jj2hp1	jj2hp2	jj3hp1	jj3hp2	sic-hp-sic	sic-s48-1	sic-hp24	sic-hp48-1	sic101amer	sic-smc1-101	patt214hp	s307-hp122
jjhpbr	0	12	12	12	12	12	12	0	0	0	0	0	0	0	0
jj0hp2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jj1hp1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jj2hp1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jj2hp2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jj3hp1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
jj3hp2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sic-hp-sic	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10
sic-s48-1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
sichp24	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
sichp481	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
sic101amer	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
sicsmc1101	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
patt214hp	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
s307hp122	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0