

Survivability and Simulation

David A. Fisher

Software Engineering Institute

Abstract

The past few years have shown a need for fundamentally new approaches to the security and survivability of large scale networked systems. Infrastructures and other modern systems pose problems that arise from many factors: partial information, complexity of combined interactions of very large numbers of human and automated participants, ubiquitous access, loss of centralized administrative control, the growing threat of automated attacks, and a shift in priorities for security from confidentiality of information to availability of services. Survivability research seeks new methods, including emergent algorithms, diversity, and dynamic trust validation, to ensure that systems satisfy their most critical requirements. Easel is an automated simulation tool for research in survivability, infrastructure assurance, and other applications that must contend with incomplete and imprecise information.

Introduction

Survivability [EFL 97] is a relatively new research domain that seeks new techniques for mission fulfillment in unbounded systems such as the Internet, electric power, telecommunications, military command and control, and health care infrastructure. The role of infrastructure assurance is to ensure the survivability and fulfillment of these missions [PCCIP 97]. It is about managing business risks to satisfy critical mission requirements in the face of competition and adversity [LF99]. It is about making trade-offs to ensure continuity of services in the absence of complete and precise information.

Survivability

Survivability is defined as the ability of a system to fulfill its mission in the presence of attacks, failures, and accidents [EFL 97]. Survivability and infrastructure assurance are *not* about adherence to industry standard security practices. They are about confidentiality, information integrity, correctness, and dependability only to the extent that these particular quality attributes are the most critical requirements of the mission [LML98]. Successful infrastructure assurance depends on sound business decision making and cannot be achieved through technological solutions alone [LF99].

Security technologies are often effective when

- the primary objective is confidentiality or integrity of information
- the system has a clearly defined perimeter with centralized administrative control
- administrators have complete insight into all aspects of the system
- all components and participants within the system are known, authenticated, and trustworthy

Perhaps most important is the assumption that applications can tolerate a binary fortress model of security in which either all attempted intrusions are successfully repelled or the system as a whole is compromised. Security solutions are inadequate if even a few of these conditions are absent. Unfortunately, modern large scale networked systems, including critical national infrastructures [PCCIP 97], seldom satisfy any of these conditions.

Unbounded Systems

An unbounded system is any system in which the participants (human or computerized) have incomplete or imprecise information about the system as a whole [FL99]. The boundaries of unbounded systems are not precisely known. Interconnections among participants in unbounded systems change constantly. Furthermore, the trustworthiness, and often the identity, of participants is unknown. Centralized administrative control cannot be fully effective in such systems. These are the characteristics of critical national infrastructures, of the Internet, and of electronic commerce. They characterize most social, economic, and biological systems, and most activities one participates in every day. Such systems contrast dramatically with the assumptions of closed, centrally controlled computer systems and with the assumptions underlying many modern computer security technologies.

Security Technology

Technologies based on fortress models of security are successful in transparent, trusted, centrally-controlled systems with clearly defined perimeters because they exploit those constraints in their environment. Success in survivable systems and infrastructure assurance likewise depends on exploiting the properties of unbounded systems to full

advantage. It is unlikely that any large networked system can be molded to satisfy the constraints of fortress-based security. In contrast with traditional security methods that attempt to build impenetrable walls around trusted insiders, a fundamental assumption of survivability is that no participant or component of a system is immune to all errors, failures, and compromises.

Survivability Technology

Survivability research seeks solutions that exploit the inherent characteristics of unbounded systems to ensure survivable missions in the presence of compromised components [ISW97, ISW98]. Unbounded systems offer opportunities for cooperation without the cost and vulnerabilities of coordination. They enable distributed specification and local optimization with their attendant resistance to standardized attacks. They facilitate robust and resilient solutions in which no component is essential and compromises of individual components will not cascade.

A survivability architecture is an abstraction that specifies the critical characteristics required for mission survivability in an unbounded system. A survivability architecture consists of the local actions of each node and the protocols of interactions among neighbor nodes. Any algorithm that fulfills mission-critical requirements by exploiting the characteristics of an unbounded system is an emergent algorithm [FL99]. Emergent algorithms instantiate survivability architectures for specific mission requirements.

Emergent Algorithms

Emergent algorithms [FL99] differ from conventional hierarchical and distributed algorithms: they operate in the absence of complete and precise information; they do not have central control, hierarchical structure, or other single point vulnerabilities; and they achieve cooperation without coordination. Mission requirements are satisfied in the form of global system properties that emerge from the combined actions and interactions of all system components. For reasons of mission survivability, our research considers only emergent algorithms that do not have single (or any fixed number of) points of failure. Work in emergent algorithms draws on a variety of techniques, including algorithmic methods for generation of emergent properties [For 91a] .

For reasons of practicality and affordability, we consider only those emergent algorithms in which the cost of each node (whether measured in dollars, CPU cycles, storage requirements, or communication's bandwidth), is less than proportional to the number of nodes in the system. The effectiveness of this approach can be further enhanced by dynamic trust validation among the participants.

Need for Simulation

Although the benefits of ad hoc development of emergent algorithms have been demonstrated, a rigorous process for deriving emergent algorithms from mission requirements is prerequisite to their widespread use in automated systems. Our intuitions about the global effects of the local actions and interactions among large numbers of nodes are seldom correct. The problem of designing emergent algorithms is especially difficult because it begins with the desired global properties and attempts to determine what simple combinations of local actions and interactions would produce those effects over time in a large scale network. An effective design methodology will depend on greater understanding of the influences of local action and interaction on emergent global properties and on the sensitivities of emergent properties to local variations. The obvious (and probably only) means to answer these questions is by simulation of emergent algorithms and of the unbounded systems in which they operate. This recognition has opened a new area of research for simulation of unbounded systems.

Limitations on Accuracy

Current simulation systems do not produce accurate predictions of the behavior of unbounded systems. By definition, unbounded systems are incompletely and imprecisely defined. Thus, a simulation of an unbounded system must be able to produce accurate results based only on incomplete information. Current models, however, require complete information and thus are always built with assumptions or inaccurate information. The ability to operate on abstract specifications and simulate at various levels of abstraction is a long-standing need of many applications but is not provided as a feature of existing simulation systems.

Equally important, all object-based models (both physical and computerized) are inherently inaccurate because they are based on complete representations as objects. This might be acceptable when dealing with small numbers of nodes or when great care is taken to differentiate between which modeling results are likely to be valid. Such remedies seldom, if ever, succeed in differentiating inaccurate results when modeling complex or large-scale

systems. Furthermore, as the number of subsystems in a model increases, the inaccuracies of each subsystem pervade the whole after a few iterations and guarantee that all simulation results will be inaccurate. This may account for the pervasive failure of large-scale simulations to produce accurate results. These problems are aggravated in unbounded systems where the numbers of components are very large and a primary purpose of simulation is to accurately predict the global effects of local activities. Because accuracy and completeness are not simultaneously achievable when describing the physical world, accurate simulation is feasible only if the simulator can guarantee accurate results from accurate but incomplete specifications.

Other difficulties in simulating unbounded systems include

- the need for thousands to tens of thousands of nodes per simulation
- the lack of linguistic mechanisms in programming languages for making incomplete and imprecise specification
- the inability of object-oriented computations to describe and reason about the real world
- the need to combine information about a system from multiple knowledge domains
- management of multiple simultaneous beliefs of the various stakeholders in an infrastructure
- integration among separately developed simulations
- exponential increases in computational cost that accompany linear increments in the granularity or number of nodes in a simulation.

Easel Simulation System

These considerations have led to a new approach to simulation called Easel, an emergent algorithm simulation language and environment [Fis99]. Easel employs a paradigm of property-based types (i.e., describing abstract classes of examples by their shared properties) to simultaneously address all of the above simulation problems. Because Easel is property-based, it can be used to give accurate, but incomplete, descriptions of anything. In combination with an appropriate automated logic system, it can be used to produce accurate conclusions about examples from the physical world. This contrasts with physical models and automated simulations that depend on representations of objects, where descriptions must be complete (and thus inaccurate), and in which conclusions are accurate only for the model but never for their extensional interpretation in the real world. While traditional modeling and simulation systems answer all questions without a mechanism for a user to determine which answers are accurate, Easel reports what additional information is needed to continue toward an accurate result. Easel also supports multiple levels of abstraction, multiple simultaneous belief systems, distributed specification, and dynamic graphic depictions.

Easel is a discrete event simulation language but adds limited support for continuous variables. The linguistic limitations of traditional programming systems for incomplete and imprecise description are overcome by the use of quantifiers, adjectives, improper nouns, pronouns, and other forms of anonymous reference in the language. In combination with property-based types, these mechanisms provide a semantic framework for referring examples of any type, whether real or imagined, and whether from the computational, mathematical, or physical worlds.

Development Status

Easel is currently under development. Preliminary versions of the Easel Language Reference Manual (ERLM) and the Easel Author Guide (EAG) are scheduled for release with the alpha version of the system at the end of February 2001. A working version that includes the translator, interpreter, graphic depiction system, and run-time support (but with an incomplete set of operations) is now being tested. The beta release is scheduled for September 2001. Until then, much work is required to validate the language design for a few of the most important applications, to complete the system design and implementation, and to measure and tune system performance. We anticipate that additional requirements will be discovered during this process.

Research and Development Partners

External partners and sponsors are needed for continuing Easel development and applications. Resident affiliates are sought for research in survivability, simulation, and emergent algorithms. Partners are also sought for experimental development of applications in infrastructure assurance and other unbounded domains. Except for research and development partners as described above, we do not anticipate beta sites for Easel outside Carnegie Mellon University.

Summary

Survivability requires new techniques for mission assurance in the absence of complete and precise information. Promising approaches include using emergent algorithms for resilient and robust systems in the presence of multiple component compromises, dynamic trust validation for marginalizing compromised components and participants, and cooperation without coordination for avoiding single points of vulnerability. The Easel simulation system holds promise as an effective research tool for fulfillment of critical mission requirements in infrastructures and other applications that must operate with incomplete information. Easel also overcomes several long-standing barriers to accuracy, scalability, and abstraction in large-scale simulations.

Copyright © Carnegie Mellon University 2000

Acknowledgements

The work described here includes contributions from Howard F. Lipson, David A. Mundie, Alan C. Christie, other members of the survivability research team at CERT/CC and many graduate and undergraduate students at Carnegie Mellon University.

References

- [For 91a] Forrest, S., Editor. *Emergent Computation: Self-organizing, Collective, and Cooperative Phenomena in Natural and Artificial Computing Networks*. Cambridge, MA: MIT Press, 1991.
- [PCCIP 97] Presidential Commission on Critical Infrastructure Protection. *Critical Foundation—Protecting America's Infrastructures*. The Report of the Presidential Commission on Critical Infrastructure Protection, October 1997.
- [EFL 97] Ellison, R. J.; Fisher, D.; Linger, R. C.; Lipson, H. F.; Longstaff, T. A.; & Mead, N. R. *Survivable Network Systems: An Emerging Discipline* (CMU/SEI-97-TR-013). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, November 1997, revised May 1999 [online]. Available WWW: <URL: <http://www.sei.cmu.edu/publications/documents/97.reports/97tr013/97tr013abstract.html>>.
- [ISW 97] *Proceedings of the 1997 Information Survivability Workshop*. San Diego, CA, February 12-13, 1997. Software Engineering Institute and IEEE Computer Society, April 1997. Available WWW: <URL: <http://www.cert.org/research>>.
- [ISW 98] *Proceedings of the 1998 Information Survivability Workshop*. Orlando, FL, October 28-30, 1998. Software Engineering Institute and IEEE Computer Society, 1998. Available WWW: <URL: <http://www.cert.org/research>>.
- [LML98] Linger, Richard; Mead, Nancy; & Lipson Howard. "Requirements Definition for Survivable Network Systems." *Proceedings of the Third International Conference on Requirements Engineering (ICRE)*. Colorado Springs, CO, April 6-10, 1998. IEEE CS Press, 1998.
- [FL99] Fisher, David & Lipson Howard. "Emergent Algorithms: A New Method for Enhancing Survivability in Unbounded Systems." *Proceedings of the Hawaii International Conference on System Sciences*. Maui, HI, January 1999. IEEE Computer Society, 1999.
- [Fis99] Fisher, David. "Design and Implementation of Easel, A Language for Simulating Highly Distributed Systems." *Proceedings of MacHack 14, 14th Annual Conference for Leading Edge Developers*. Deerborn, MI, June 1999.
- [LF99] Lipson, Howard & Fisher, David. "Survivability—A New Technical and Business Perspective on Security." *Proceedings of the 1999 Security Paradigms Workshop*. Caldeon Hills, ON, September 21-24, 1999. New York, NY: Association for Computing Machinery, 1999.