Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

A PARALLEL SIMULATION FRAMEWORK FOR INFRASTRUCTURE MODELING AND ANALYSIS

Donald D. Dudenhoeffer
May R. Permann
Elliot M. Sussman

Idaho National Engineering and Environmental Laboratory
P.O. Box 1625
Idaho Falls, ID 83415-3605, U.S.A.

ABSTRACT

Today's society relies greatly upon an array of complex national and international infrastructure networks, such as transportation, utilities, telecommunication, and even financial networks. While modeling and simulation tools have provided insight into the behavior of individual infrastructure networks, a far less understood area is that of the interrelationships between multiple networks. Specifically, how does an event in one network affect the operation of the other networks. This paper presents the work that is being conducted at the Idaho National Engineering and Environmental Laboratory (INEEL) to model and simulate these complex behaviors between coupled infrastructures.

1 INTRODUCTION

The Joint Program Office for Special Technology Countermeasures (JPO-STC) is an organization chartered by the Office of the Secretary of Defense (OSD), which is responsible for oversight, management, and execution of the DoD's Infrastructure Assurance Program (IAP). The "ability of DoD to plan and execute the National Military Strategy hinges on the availability of infrastructure assets. Accordingly, the IAP is dedicated to ensuring that critical infrastructure will be readily available when needed across the full range of military operations, seen as force protection, emergency response, and force projection. " (JPO-STC 2002).

Inherent in this concept is the "irreducible critical sub-network" that contains the mission essential assets (not all parts of a "Critical Infrastructure" are equally critical). Identifying and protecting from attack the critical sub-network (CSN) is an important unsolved problem when the assets are interdependent and we do not understand the emergent behaviors of these networks. Much modeling has been done in the case of individual networks such as power grids, communication networks, etc. It is the objective of this multi-year project, however, not only to model the individual networks, but also to incorporate and model the interdependencies between them. It is only through an analysis of these complex interrelationships that key assets can be identified and vulnerabilities can be adequately assessed. This paper describes the modeling and simulation framework being developed at the Idaho National Engineering and Environmental Laboratory (INEEL) to analyze emergent inter-network behaviors.

2 BACKGROUND

2.1 Research Motivation

Critical infrastructures are those physical and cyber-based systems essential to the minimum operations of the economy and government. They include telecommunications, energy, banking and finance, transportation, water systems and emergency services.

Due to advances in information technology (IT) and the necessity of improved efficiency, infrastructures have become increasingly automated and interlinked. Most modern commercial infrastructures are composed of a collection of interconnected networks that serve different purposes and have different owners. Indeed, even parts of the information residing on a single sub-network may have different purposes and different owners. Critical information is passed between these component elements to coordinate necessary functions. The complexity and interdependency of this critical information flow introduces vulnerabilities into the entire critical infrastructure. Deliberate attacks or accidental system failures may result in serious consequences to the nation.

This interrelationship among infrastructures and its potential for cascading effects was never more evident than on July 19, 2001 when a 62-car freight train carrying hazardous chemicals derailed in Baltimore's Howard Street Tunnel. This disaster, in addition to its expected effect on rail system traffic, automobile traffic, and emergency services, caused a cascading degradation of infrastructure components not previously anticipated. For example, the tunnel fire caused a water main to break above the tunnel shooting geysers 20 feet into the air. The break caused localized flooding which exceeded a depth of three feet in some areas. Additionally, the flooding knocked out electricity to about 1200 downtown Baltimore residences (Layton and Phillips 2002). Fiber optical cables running through the tunnel were also destroyed. This resulted in major disruptions to phone and cell phone service, email service, web services, and data services to major corporations including WorldCom Inc., Verizon Communications Inc., the Hearst Corp. in New York City, Nextel Communications Inc., and the Baltimore Sun newspaper (Ratner 2002). Disruption to rail services and its effects on the Middle Atlantic States were significant also. These effects included delays in coal delivery and also limestone delivery for steel production (Little and Adams 2002).

With the need to maintain base capabilities and functionality to meet national security needs, infrastructure owners have not sufficiently come to grips with the difficulty of subdividing a whole infrastructure according to its "criticalness". By and large today, the "owners" of Critical Infrastructures have not come up with a good way to approach the idea that there might be some Minimum Critical Infrastructure that only includes a subset of all infrastructure elements.

In analyzing the concept of some Minimum Critical Infrastructure, the key questions to be asked in evaluating the coupling between infrastructures include:

1) What is the effect on key capabilities (national, institutional, or facility) due to the emergent effect of a failure or disruption of one or more seemingly unrelated infrastructure items?
2) If an asset or capability is deemed critical, what vulnerabilities exist in regards to such emergent behaviors? Can such emergent behaviors be predicted and thus countered in sufficient time to minimize the effect?
3) If the asset is both critical and vulnerable, then it requires remediation. What metrics do you apply to prioritize remediation resources?

Even this simple approach is not currently well understood. What makes this review even more interesting as a research topic is that looking at each component individually does not capture the emergent behavior of the CSN as a network when under attack.

2.2 Terminology

The infrastructure modeling within this project is based on a graphical representation of the individual infrastructure networks as nodes and edges. Within this context, nodes and edges have specific meaning and are defined as such:

- *Node* – a physical entity that either acts as a source, produces, consumes, or transforms a resource.
- *Edge* – a physical or potentially virtual entity that acts as a conduit for flow. As described later in the paper, edges are further divided into flow edges and control edges.
- *Resource* – a commodity used within an infrastructure network.

Consider the following simple network example: a community water storage tank is connected via the water main pipe to a pump at a distribution station. The tank is considered a node since it is a source of the resource water. The pump likewise is a node since it transforms (i.e., imparts force on) the water. The pipe between the tank and pump is the edge joining the two nodes. To illustrate the interrelationship between infrastructures, now consider the edge that exists between the pump and the electrical power grid that provides the energy to run the pump. In this sense the pump is a consumer of the resource of electricity. A disruption therefore in either the water main system or the electrical system may impact the capability to deliver water for drinking, fire fighting, etc.

2.3 Simulation Requirements

The simulation framework required to model multiple networks and their interrelations presented some unique challenges in terms of both size and complexity. The driving requirements for the development of this project include:

- The ability to model tens of thousands of nodes;
- The ability to concurrently model at least as many edges as nodes;
- Visual representation of the infrastructure networks during the simulation run. This represents an important and driving aspect of the simulation development.
- The ability to drill down to multiple levels of granularity. For example, the ability to model nodes as distinct entities or to model nodes as individual networks consisting of sub-nodes;
- The ability to inject "attack vectors" which simulate attacks on specific nodes and edges, or at specific geographic locations;
- The ability to change parameters during simulation execution to facilitate "what if" analysis;
- Once the analysis is underway, the results are considered "sensitive" and potentially "classified" due to the nature of revealing known and potential unknown infrastructure vulnerabilities.

2.4 System Implementation

Due to the sheer mass of information and computational complexity, the utilization of a desktop PC proved to be impractical. The simulation has been parallelized in order to make it as scalable as possible. Distributing work among multiple processors allows the simulation of multiple infrastructures to be run simultaneously. The multiple infrastructure networks are split among the processors and each processor does the simulation on its part of the network at each time-step. This speeds up the process and allows for larger networks, as each processor addresses only a subset of the total system.

The structure of the parallel implementation was influenced by the need for an interactive system display to promote visual analysis and allow user interaction during a simulation run. The Master Processor / Working Processor architecture was designed in which the Master Processor controls simulation timing, conducts all system I/O, and produces the visualization.

The Message Passing Interface (MPI) was used to parallelize the code and facilitate information exchange between processors. The version of MPI used is MPICH, a freely available, portable implementation of MPI (ANL 2002). Development has been centered on both an SGI Origin 3800 with 64 400 Mhz processors, 6.4 GB of memory and 7.8 TB disk space, and a cluster machine consisting of 43 nodes with 1.2 GHz dual Athlon processors, 2 GB of memory and 20 GB of diskspace each. The final application is meant to run on a secure cluster machine which is currently under construction and will initially consist of 20 nodes with 500 Mz Athlon processors, 384 MB of memory, and 6 GB of disk space on each machine. MPICH 1.2.1 is installed on the SGI Origin and version 1.2.3 is running on the Mandrake cluster.

2.5 Related Work

In the domain of infrastructure modeling, numerous works and studies have focused on the modeling, simulation and analysis of single infrastructure elements. Few projects, however, have attempted to combine multiple networks into one model with the specific intent of modeling and analyzing the interrelationship between networks.

The National Infrastructure Simulation & Analysis Center (NISAC) is being established by the Los Alamos and Sandia National Laboratories to assess and study the interdependencies between national infrastructure elements. This effort most closely resembles the objective and goals of the effort at the INEEL.

The infrastructure modeling and simulation at the INEEL, however, differs in the scope or the bounds on which the model is trying to address. This project represents an effort not to model the national infrastructure grid, rather it attempts to model the infrastructures centered around a facility or institution and the functional capabilities required therein.

As an example, the initial data set for this project consists of the infrastructure elements at the Idaho Nuclear Technology and Engineering Center (INTEC), Figure 1.

Figure 1: INTEC

INTEC is located 53 miles west of Idaho Falls occupying 200 acres in the middle of the INEEL's 890-square-mile reservation. Its mission is to:

- Safely store spent nuclear fuel and prepare it for shipment to an offsite repository;
- Develop technology to safely treat high-level and liquid radioactive waste that resulted from reprocessing spent fuel;
- Remediate past environmental releases.

The nature of the work being conducted at INTEC coupled with the hazardous material that is stored there makes security of the highest priority. The objective of modeling INTEC is to identify potential vulnerabilities, identify emergent behaviors from infrastructure outages, and to assist in the development of protective strategies.

3 SIMULATION DEVELOPMENT AND VALIDATION

3.1 System Overview

The overriding objective of this project is to examine the interrelationships between infrastructure networks and more specifically, the emergent systems behaviors that develop when one or more nodes within the system are perturbed. The system being modeled is highly nonlinear and can be considered a complex system. A complex system being defined as:

> "one whose component parts interact with sufficient intricacy that they cannot be predicted by standard linear equations; so many variables are at

work in the system that its overall behavior can only be understood as an emergent consequence of the holistic sum of all the myriad behaviors embedded within." (Levy 1992).

Given the complex nature of the problem, an agent-based model (ABM) (Rocha 1999) was chosen to model the infrastructure elements. Agent is a term widely used in literature today, varying from software "assistant" to physical robotic entities. Within the context of this paper, an "agent" refers to a model of a physical entity, namely a Node or an Edge. The key characteristic of the agent and the simulations is that each agent exists as an individual entity which maintains a state, senses input, and possesses rules of behavior that act upon the inputs and either modify the state or produce an output.

Each network within the simulation is modeled as a connected graph, G = (N, E), where N represents the nodes within the network and E represents the edges between the nodes. Edges represent the only channel by which information or resources flow between nodes. An edge represents a path of bi-directional flow with a positive (+) value indicating the normal flow directions. The edge itself also possesses state properties such as capacity, geographical location, and current flow rate.

Edges are further subdivided into two categories, flow-edges and control-edges. Flow-edges nominally represent the flow paths for resources between nodes. Control-edges provide the communication path and also the control parameters for control signals between nodes. As an example, consider a water tank with and initial tank level of L. The tank is connected to a suction pump via a pipe. The tank and pump are nodes and the pipe is a flow-edge. When the tank level reaches a low level set point, LL, i.e., L <= LL, a control signal is sent via a control-edge to the pump that turns the pump off.

The simulation contains multiple networks distributed across multiple processors. The initial partitioning of the networks was conducted by hand and coded as part of the data input file. The networks were assigned to individual processors when possible. Edges provide both intra-rocessor and across processor communication between nodes.

3.2 System Architecture

Figure 2 illustrates the simulation architecture. This multi-network infrastructure simulation is based on parallel discrete event simulation (PDES) (Fujimoto 1990), but also incorporates elements of agent-based modeling (ABM).

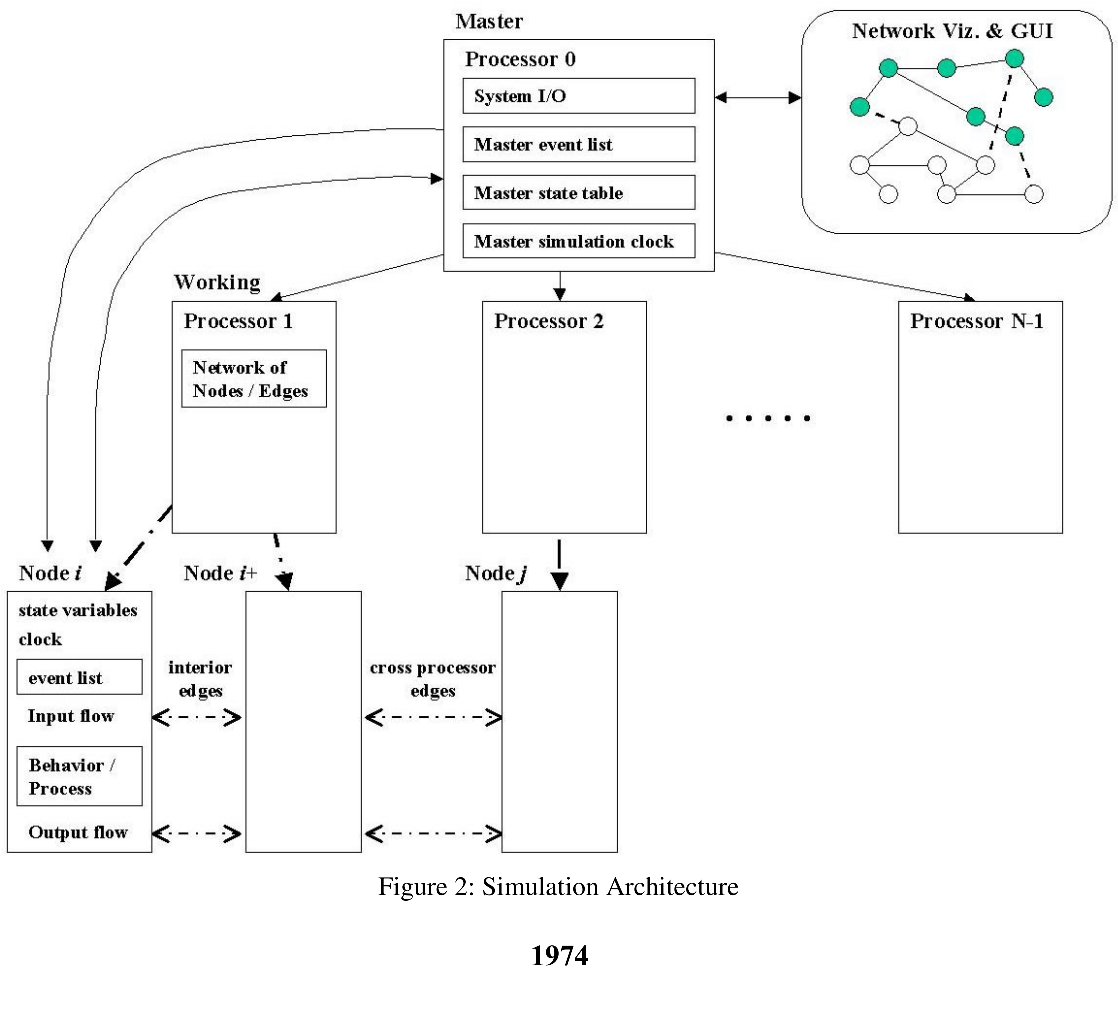

Figure 2: Simulation Architecture

Processor 0 is the Master Processor. The Master Processor controls the flow of the simulation, including the simulation clock, handles user input, provides the visualization, and also provides the final output. The Master processor also contains a table with state information for all nodes within the simulation. This table is used for simulation output and also to drive the visualization.

The Working Processors, processors 1…n, contain the network structures. The network nodes and likewise edges are contained within arrays of structures. Nodes are wholly contained on one processor, but if an edge connects nodes between processors, an instance of that edge is contained on both processors.

During a simulation step, the Working Processor loops through its array of nodes. The nodes first evaluate the event queue and modify any state parameters if necessary. Next the nodes evaluate their environment. A node nominally contains one or more input flow-edges and one or more output flow-edges. Each node contains its own behavior set or production model, which translates input flow into output flow. Likewise, the node may contain control-edges, which will trigger message passing to other nodes if specific internal state parameter levels are met.

At the end of the simulation step, if the process has altered the state of the node, a message is sent updating the Master State Table that in turn updates the visualization.

3.2.1 Synchronization

As stated earlier, this simulation is based on parallel discrete event simulation (PDE), but it is also influenced by agent-based modeling and also on the concept of sequential dynamic systems (SDS) (Barrett et al. 2000).

The division of a problem into partitions to support the powers of parallel execution results in the problem of synchronization: ensuring that the logical processes (LP) divided across the multiple processors occur in the correct order. Time management algorithms have been developed to help address this problem. These algorithms are normally classified as conservative or optimistic in nature. Conservative algorithms utilize "null" message passing between processors and "look ahead" to promote synchronization. One of the approaches of optimistic algorithms is to allow the free flow of time stamped messages. If a received message has a time stamp previous to messages already processed, the effect of processed messages are essentially undone and the system is "rolledback" to the time state of the newly received message. Fujimoto (2001) has an in-depth discussion of time management is parallel systems.

Time management is controlled by the Master Processor. The simulation utilizes a fixed-increment time advance approach (Law and Kelton 2000). The Master Processor dictates the simulation time step size and issues a message to each Working Processor to execute that time step. MPI "blocking receives" are then utilized to ensure that all processors complete the execution of that time step prior to the Master issuing the next time increment. This approach was adapted to ensure that the visualization reflected the same time-step across all infrastructure nodes.

Once a message is received from the Master to execute a simulation step, the Working Processors sequentially loop through their list of nodes and execute a simulation step. During the simulation step, each node checks their individual event list to identify events that should occur during the current time increment. The ordered looping through the nodes can possibly introduce artificialities given that one node may directly affect another node. An attempt to minimize this effect is to randomize the ordering of node simulation step execution with each network.

Messages between nodes are time stamped and added to the nodes event queue upon receipt. Messages are then executed if they fall within the current simulation time step.

3.3 Data

Data collection represents a tremendous effort within this project. The infrastructure data elements being collected do not exist in a readily available format. Subject matter experts (SMEs) have been utilized to interpret control diagrams, electrical schematic, piping diagrams, networking diagrams, etc. In this initial data gathering effort, the information has been incorporated into a Microsoft Access™ database. This database is incompatible with the ultimate size requirements of the infrastructure data elements, but provides a suitable prototype for development.

Data sensitivity is another issue that exists. The raw data and even the database are not considered necessarily sensitive, but once they are incorporated into the simulation, the potential to expose vulnerabilities dictates the need for a secure computing system.

Figure 3 illustrates the framework, which has been developed as part of the initial implementation. Data are collected via the Access database. It is then exported to a text file. The text file is then converted to an input format for the simulation. The simulation is to reside on a secure computer system with no connection to any outside systems. The database and simulation exists on physically separate machines. Uploading of the input file will occur via magnetic diskette or CD. Simulation output is both graphical via the interface and textual with a program generated output file.

3.4 Visualization

Visualization is a major component of the simulation program as a means of visual analysis. Although simulation end states are important, understanding the emergent behaviors, which occur during the simulation, is extremely important. The ability to identifying and observe these

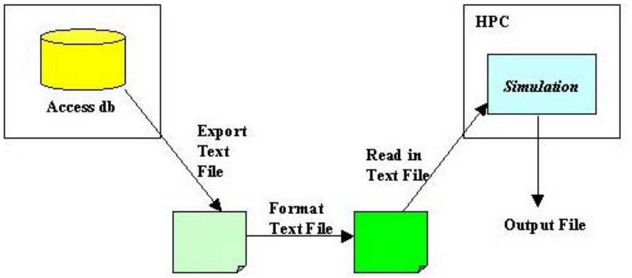

Figure 3: Data Management

emergent behaviors visually, thus became a design requirement. The visualization is still in development as prototype designs are under evaluation. The visualization will show each physical and virtual entity in each network infrastructure along with the coupling between the infrastructures. Each individual network can be represented separately in a layered fashion on a 3D plane. The user has the ability to select multiple viewpoints by adjusting the view angle, position, and zoom factor. Entity icons will be color coded to reflect the current state and functionality.

The ability for user interaction during the simulation is made possible through a graphical user interface based on a 3D geographical representation of the infrastructure. It is intended to allow users to watch the simulation as it develops and allow run-time interaction. This interaction includes the ability to zoom and focus on certain areas, select entities (nodes and edges) to view their current parameters, alter parameters, add future events, and start, pause or stop the simulation. This interaction will allow users to gain much greater insight into the critical infrastructure through online "what if" system perturbations than would be possible with just a final output at the end of the simulation.

Currently, two 3D visualization tools are being considered. VTK (`http://public.kitware.com/VTK/`) was the first choice for its functionality. VTK is an open source, multi-platform visualization package. It also has the ability to be parallelized, although at this point our project does not require intense rendering that would necessitate this. Another product being considered for the visualization is Hoops (`http://www.hoops3d.com/`). The graphical user interface, which resides on top of the visualization, likely will be implemented with either FLTK (`http://www.fltk.org/`) or QT (`http://www.trolltech.com/`).

Figure 4 illustrates one of the interfaces we are investigating. This was created using Hoops and QT. The nodes at the intersection of the edges are active objects. Clicking on one of them brings up a status box for that object.

Figure 4: Prototype Display and Interface

4 CONCLUSIONS

This paper presents the research that the INEEL is conducting in the area of infrastructure simulations. Specifically, the INEEL is modeling multiple infrastructures and the relationships that exists between them with the objective of gaining understanding into emergent behaviors that evolve as elements of the system are perturbed. This work is ongoing and still in its preliminary stages. The purpose of this paper is to present the design architecture and rational behind that design. In developing the simulation, only a few resources were available that discussed design in detail. This paper attempts to elaborate on those papers and present our methodology in approaching infrastructure simulation.

The development of parallel simulation code presented several challenges. Our implementation of message passing was based on MPICH which does not deal with objects. To utilize an object oriented approach we would have to incorporate additional library functions such as Object Oriented MPI (OOMPI); (`http://www.osl.iu.edu/research/oompi/`). In the interim, nodes and edges are modeled as structure data types.

Another issue dealt with "passing by reference" within functions. The SGI MIPSpro C++ version 7.3.1.2m compiler would not compile the C++ "call by reference" syntax. "Call by reference" had to be done via the C language syntax. This was not a problem on the cluster machine in which we utilized the Mandrake g++ version 3.0.4 compiler.

A conflict also existed with the use of MPICH 1.2.1 and the visualization package VTK. This conflict involved the use of a data structure called "List" in both MPICH and VTK which resulted in a link error. This was solved by doing a global find and replace, renaming the "List" in VTK to "Vlist".

This report reflects the initial efforts of a multi-year project. Future developments include the development and integration of subnodes and subnetworks around current nodes. This would allow the user to select the level of the granularity of a particular node to be modeled. The needed fidelity of the model and the modeling requirements may change depending on the analysis to be performed.

Another area to be explored is that of geographic effect. Currently the user can change the state of nodes and edges only by selecting the node or edge and changing a parameter. A future functionality is to insert an event centered at a geographic location and have that event affect all nodes and edges within a specific radius.

One area not currently being explored by this project, but which could potentially have a power effect, is the partitioning of the infrastructure networks between processors. This is currently accomplished by hand based on the user's best guess of message passing intensity and computational requirements. It is potentially inefficient and is also very time consuming. A more effective automated method would be of great benefit.

Data collections and verification is still in progress for modeling of the Idaho Nuclear Technology and Engineering Center (INTEC). Future reports will detail our simulation results and analysis from this data set as well as our success in incorporating new functionality into the simulation.

ACKNOWLEDGMENTS

This work is supported through the INEEL Long-Term Research Initiative Program under DOE Idaho Operations Office Contract DE-AC07-99ID13727.

REFERENCES

Argonne National Laboratory (ANL). 2002. MPICH-A Portable Implementation of MPI. Available online via <http://www-unix.mcs.anl.gov/mpi/mpich/> [accessed March 28, 2002].

Barret, C. L., H. S. Mortveit, C. M. Reidys. 2000. Elements of a theory of simulation II: sequential dynamical systems. *Applied Mathematics and Computation* 107:121-136.

Fujimoto, R. M. 2001. Parallel and Distributed Simulation Systems. In Proceedings of the 2001 Winter Simulation Conference, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 147-157. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers. Available online via <http://www.informs-cs.org/wsc01papers/018.PDF> [accessed March 28, 2002].

Joint Program Office for Special Technology Countermeasures (JPO-STC). 2002. About JPO. Available online via <http://www.nswc.navy.mil/IAP/jpo/about.htm> [accessed March 28, 2002].

Law, A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis, Third Edition.* Boston: McGraw Hill.

Levy, S. 1992. *Artificial Life – A report from the Frontier Where Computers Meet Biology*. New York: Vintage Books.

Little, R. and P. Adams. 2001, Tunnel fire choking East Coast rail freight. Available online via <http://www.baltimoresun.com/news/local/bal-te.bz.freight20jul20.story?coll=bal-home-headlines> [accessed March 28, 2002].

Layton, L. and D. Phillips. 2001. Train Sets Tunnel Afire, Shuts Down Baltimore. Available online via <http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A17542-2001Jul18>. [accessed March 28, 2002].

Ratner, A. 2001. Train derailment severs communications. Available online via <http://www.baltimoresun.com/news/local/bal-email19.story?coll=bal-home-headlines> [accessed March 28, 2002].

Rocha, L. M. 1999. Complex System Modeling: Using Metaphors from Nature in Simulation and Scientific Methods. BITS: Computer and Communications News. Los Alamos National Laboratory, November 1999. Available online via http://www.c3.lanl.gov/~rocha/complex/csm.html> [accessed March 26, 2002].

AUTHOR BIOGRAPHIES

DONALD D. DUDENHOEFFER is research scientist for the Idaho National Engineering and Environmental Laboratory (INEEL) in Idaho Falls, ID. He received his Masters of Science degree in Operations Research from the Naval Postgraduate School in 1994. His research interests include simulation, robotics, military operations and command and control. His email address is <dudedd@inel.gov>.

MAY R. PERMANN is a research scientist at the Idaho National Engineering and Environmental Laboratory (INEEL) in Idaho Falls, ID. She received her B.S. in Computer Science from Idaho State University. Her research interests include high performance computing (HPC) and scientific visualization. Her email address is <chafmr@inel.gov>.

ELLIOT SUSSMAN is a research scientist at the Idaho National Engineering. He holds a M.S. in E.E. from Southern Methodist University and a B.S. Electronic and Computer Engineering from Clarkson University. More than 20 years of professional experience in control systems design and implementation, data acquisition systems integration, and real-time computer applications. Principal Investigator for the INEEL's Laboratory Directed Research project: Critical Infrastructure Protection. His email address is <sussmaem@inel.gov>.