DISCRETE-EVENT SIMULATION FOR THE DESIGN AND EVALUATION OF PHYSICAL PROTECTION SYSTEMS

Sabina E. Jordan, Mark K. Snell, and Marcella M. Madsen

Sandia National Laboratories
Department 5838, Mail Stop 0780
1515 Eubank Blvd. Southeast
Albuquerque, NM 87123, U.S.A.

Jeffrey S. Smith and Brett A. Peters

Texas A&M University
Industrial Engineering Department
238 Zachry Engineering Center
College Station, TX 77843-3131, U.S.A.

ABSTRACT

This paper explores the use of discrete-event simulation for the design and control of physical protection systems for fixed-site facilities housing items of significant value. It begins by discussing several modeling and simulation activities currently performed in designing and analyzing these protection systems and then discusses capabilities that design/analysis tools should have. The remainder of the article then discusses in detail how some of these new capabilities have been implemented in software to achieve a prototype design and analysis tool. The simulation software technology provides a communications mechanism between a running simulation and one or more external programs. In the prototype security analysis tool, these capabilities are used to facilitate human-in-the-loop interaction and to support a real-time connection to a virtual reality (VR) model of the facility being analyzed. This simulation tool can be used for both training (in real-time mode) and facility analysis and design (in fast mode).

1 GENERAL PROBLEM

This paper examines the problem of designing and analyzing physical security systems that protect fixed-site facilities against intrusions by external threats as well as unauthorized acts by insiders. To function properly, these systems must first detect the adversary act, delay the progress of the adversary, and respond (typically with guards) to the intrusion or act.

Proper design and/or analysis of these systems include determining how well these functions work alone and in combination. For the purposes of this paper, we will differentiate between analysis functions and design functions by stating that analysis looks at the effectiveness of existing systems while design consists of creating new systems or modifying existing systems.

The following four modeling and simulation steps are currently performed as part of this design/analysis process:

1. Determining the performance of different detection and delay features, such as sensors and locks and entry control procedures, against the adversary (in terms of how they can be defeated, or the time or probability of detection involved in defeating them).
2. Determining whether alarm/assessment systems and entry control networks operate quickly and reliably enough to allow the guard force to determine that a security response should be started.
3. Determining whether there is enough delay built into the physical protection system, once the adversary has been detected, to allow the response team to interrupt – that is, arrive in time to confront – the adversary.
4. Determining how capable the response forces are at defeating the adversary in a battle, if one ensues, taking into account the numbers of combatants on both sides, their weapons, tactics, and other factors.

The first two analysis steps consist of inspecting the actual or designed system and performing or reviewing performance tests. Examples of performance tests include: trying to see if testers with false badges can pass through entry control points, conducting explosive attacks on doors, recording the processing times and operator decisions for simulated intrusions through sensors, and performing limited exercises to determine response force times. Historically, physical protection systems have been simple enough that interactions between subsystems did not need to be modeled. In those areas where systems had some interactions – such

DISCLAIMER

DISCLAIMER

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

as perimeter alarm sensing and assessment – actual installations of the equipment were often used to investigate these issues. Where systems were being designed, this often meant that physical test beds had to be built on site.

The latter two analysis steps determine how effectively the physical security system protects against the adversary. The third step produces a number of attack scenarios that stress detection and delay in the system. These scenarios, either identified by expert judgement or optimal path algorithms, are then played out in human-in-the-loop combat simulations to evaluate how effective response tactics and weapons are, if response occurs early enough.

the probability that detection occurs early enough that the response can intercept the adversary within the response force time). These tools incorporate simple analytical models that use point estimates of detection probabilities, delay times, and response force time. On the other hand, we use fairly detailed human-in-the-loop combat simulations to address response issues.

Table 1 lists 6 levels of analysis realism that can be used for performing security analysis. The term "Detailed performance models" for detection and delay models mentioned at level 3 can range from time-dependent parametric models to physics-based models of sensors or barriers. Table 2 compares the relative cost of each level of analysis realism.

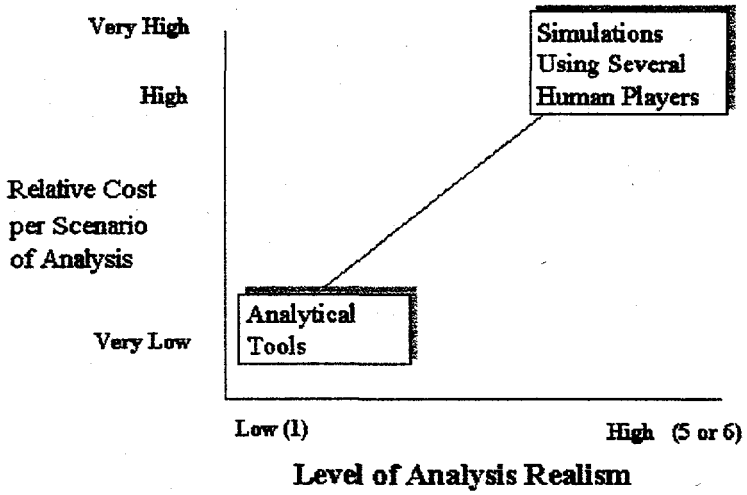

Figure 1: Comparison of Level of Analysis Realism and Cost-Per Scenario of Analysis for Existing Security Analytical Tools and Human-in-the-Loop Simulations

Figure 1 shows, qualitatively, the relationship between cost of analysis per scenario examined and level of analysis realism for analysis tools we currently use for steps 3 and 4 of the design/analysis process. By "cost of analysis" we mean primarily the manpower costs involved with collecting site information, entering it into analysis databases, modeling/simulating the scenario, and documenting the results. Analysis realism refers to how much of the detail of the actual system performance is realistically incorporated in the model (this is defined in more detail below). Low-detail tools, such as SAVI – Systematic Analysis of Vulnerability to Intrusion – described in Matter (1988), are used during step 3 to identify the paths with the lowest estimate of probability of interruption. (Probability of interruption is defined as

Due to the complexity of the problem, analytical security models such as SAVI stop at level 1 detail. Smith *et al.* (1998) also describe a level 1 analysis method using the simulation model described in this paper. At the other end of the spectrum, currently used combat simulations fall at levels 5 or 6. That is, the combat simulations are driven primarily by human participants. This leaves a gap of several layers of analysis realism that have not been adequately addressed. These levels can be covered imperfectly now by either attempting to alter the data for simple analytical tools to try to approximate the problem, performing what are called "table-top" analyses using experts, or using the level 5 or 6 simulation tools – and valuable human player time – to address the questions.

Table 1: Levels of Analysis Detail

Level	*Type of Model*	*Level of Detail of Detection and Delay Models*	*How Guard-Adversary Combat is Modeled*
1	Analytical (point estimates)	Parameters set using point estimates and/or aggregated values	Point estimates of Response Force Time
2	Analytical (stochastic)	Parameters set using distributions based on tests and/or uncertainty	Distributions for Response Force Time
3	Stochastic simulations with simple models	Detailed performance models including interaction between security features and time-varying performance	Node Adjacency models– if guards, at node i, see/are seen by adversaries at node j, what is the probability guards win the ensuing confrontation?
4	Stochastic simulations using agents	Detailed performance models including interaction between security features and time-varying performance	Computerized agents represent the behavior of security and/or adversary personnel.
5	Stochastic simulations using human commanders	Detailed performance models including interaction between security features and time-varying performance	Humans play the role of security or adversary commanders in the simulation.
6	Stochastic simulations using human participants	Detailed performance models including interaction between security features and time-varying performance	Humans play the role of specific security or adversary personnel in the simulation.

Table 2: Relative Costs/Per Scenario for Each Level of Analysis Detail

Level	*Type of Model*	*Cost/Scenario of Analysis*
1	Analytical (point estimates)	Very low – use classical optimization techniques to find an answer without explicitly addressing each scenario
2	Analytical (stochastic)	Low – individual scenarios are quickly analyzed but more scenarios need to be examined to find optimal ones
3 - 4	Stochastic simulations with simple models and/or automated agents	Moderate: Replications of each option need to be performed, with each replication modeling more details than the analytical models. May run faster than real time if run in a "batch" mode or in real-time if analyst/trainee interacts with system.
5	Stochastic simulations with humans serving as commanders	High: Several (2+) human players must be involved in performing each replication of each scenario in essentially real-time.
6	Stochastic simulations with humans serving as participants	Very High: Up to dozens of human players must be involved in performing each replication of each scenario in essentially real-time.

There are several valuable analysis questions that can best be addressed with tools covering these intermediate levels of detail (levels 2-4):

- Question: What is the value of detection by roving guards and area surveillance systems that are more complex than the line sensors used in the past? In some cases, a physics-based model of system performance may be required to fully evaluate these systems. There are also questions of where to place these sensors in two- or three-dimensional environments that don't arise when designing perimeter systems – either you alarm the entire perimeter or you don't.
- Question: How do we model cases where we want to see the effect of detailed scenario factors, such as weight of equipment adversaries are carrying, the

number of adversaries attacking a barrier or sneaking past a sensor, or the uncertainty in the results due to limited testing?

- Question: What is the effect on timeliness of the response of factors that may merely slow the response down – such as command and control problems or adversary diversions – without taking account (yet) for combat between the response and adversaries? In some of these cases, the response may not arrive in time so the ability to model combat is superfluous.
- Training need: How can a response commander be trained to make use of alarm system data to quickly close in on and engage adversary forces without having to perform a level 5 or 6 analysis?

Having identified a need for a new design and analysis capability, we will now discuss a prototype tool developed for providing these level 2-4 capabilities.

2 APPROACH TO THE PROBLEM

This research grew out of a realization that the security modeling requirements for levels 1-3 of analysis detail, that focus on detection and delay, could be met by an adaptation of a real-time simulation originally developed for controlling flexible manufacturing systems (Drake and Smith, 1996; Peters et al., 1996).

The general concept of using real-time simulation for manufacturing involves developing a single simulation model for analysis as well as shop floor control thus reducing software development costs. Like the manufacturing simulations, simulations of security systems require several levels of detail and may, in some cases, need to interact with actual security system components. The detail level can range from the entire system being simulated to the entire system being real. The development of separate software logic for all levels of detail causes duplication of effort and creates difficulty in maintaining consistency. Building on the previous research in the manufacturing control domain that led to the development of Arena RT™, this research extends the concept of single simulation logic for analysis and control to security systems simulation.

After the initial development of the simulation-based security system, it became apparent that the Arena RT-based system could also be extended to cover level 4 of analysis detail. Following is a brief overview of the simulation system that was developed for this purpose. Additional details about the simulation and the level 1-3 analyses see Smith *et al.* (1998).

3 SIMULATION DESCRIPTION

In the security simulation the facility layout is modeled as a graph. The nodes of the graph represent points along paths and positions of security features (doors, portals, gates, etc). Often there are or more sensors installed on or around these security features (e.g., motion detectors, cameras, etc.). Arcs connecting nodes represent paths. For example, a corridor in a building can be modeled using two nodes (one for each end of the corridor) and an arc joining the two nodes. The security features associated with each node determine the delay time and detection probability at that location. For example, a node representing a door would have a time associated with opening the (potentially locked) door

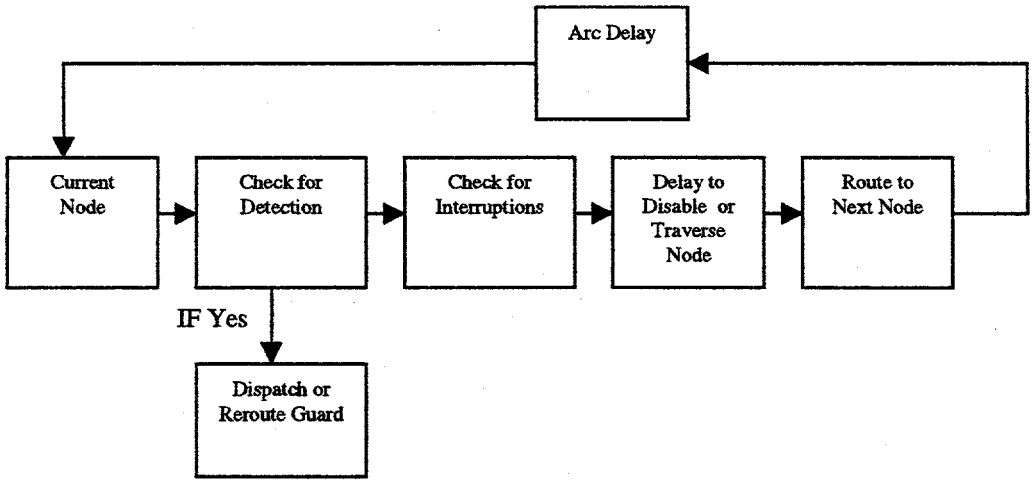

Figure 2: Entity logic for the intruder detection simulation

and a probability of this action being detected by an installed sensor. Each arc has a delay time based on the travel time between the locations represented by the connected nodes. The response team and the intruder behaviors determine the travel paths. As the simulation progresses, intruder entities and guard entities move through the building graph and interact with one another. Events occur probabilistically when entities (intruders and guards) arrive at nodes. Events include detection, interruption, and neutralization.

The simulation logic is quite straightforward, since much of the processing is performed in the VBA code. Figure 2 illustrates the logic as entities move through the system. Entities move from node to node through the graph, incurring the corresponding node and arc delays. Upon arrival at a node, the simulation checks for detection by sensors and/or guard/adversary interruptions. When intruders or guards are neutralized or when intruders reach the target, the corresponding entities are destroyed.

3.1 Entity Behavior

In a real security application, the intruders' goal can be simply to reach the target (e.g., where sabotage is the objective) or can be to retrieve the target and escape (e.g., where theft is the objective). The guards' goal is to neutralize the intruders. When intruder and guard entities come into contact with one another (called *interruption*), a battle ensues. The result of the battle is either neutralization of the intruder, neutralization of the guard, neutralization of both, or neutralization of neither. Interruption and battle outcome are probabilistic events based on the location of the two associated entities. The simulation model tracks the probability that intruders are detected, the probability that intruders are neutralized, and the expected time remaining before the intruder reaches the target when neutralization occurs. These performance metrics are used to evaluate the security system design.

Although the simulation logic is straightforward, implementing entities' behaviors within the simulation framework has proven to be quite interesting. In the manufacturing control models on which the security models are based, part routes are determined in advance and can be changed at the discretion of the shop floor control system. The analogous behavior for intruder entities is for them to follow a fixed path through the facility graph. Similarly, for guard entities, the analogous behavior is to either follow a fixed path to the target upon intruder detection, or to follow a preset "patrol" through the facility.

Behaviors become even more interesting when entities interact with one another. For example, if an intruder sees a guard, one logical behavior would be to exit the facility (i.e., run away). Similarly, taking the shortest path to the location of a detection or chasing an intruder would be logical behaviors for the guard. These types of behaviors have been implemented in the current version of the simulation. Cooperative behaviors (where multiple guards or intruders cooperate with one another in order to meet an objective) represent the next level of complexity. Cooperative behaviors have not yet been implemented in the simulation system.

For simple behaviors, entity routes can easily be handled using the SEQUENCES element in the Arena

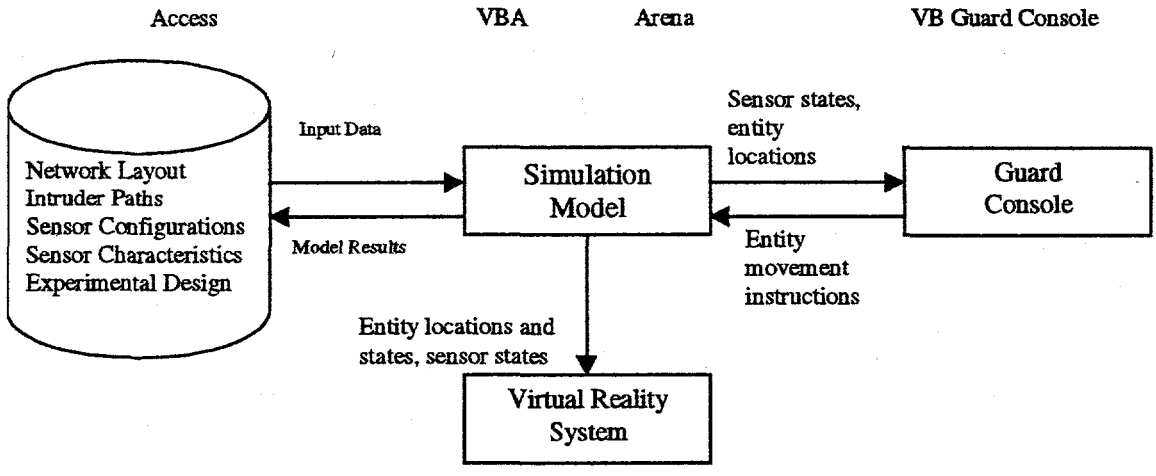

Figure 3: Simulation structure

modeling language. However, in order to implement more complex behaviors, entity routes are determined manually in a VBA block rather than using the traditional SEQUENCES element. This allows arbitrarily complex and dynamic routing logic to be used. For example, a shortest path algorithm can be used to find an egress path from the current node if the intruder is trying to escape after being detected.

Figure 3 shows the structure of the current simulation system. As shown on the left side of the simulation block in Figure 3, the network layout, initial intruder paths, and sensor information are stored in an Access database. This information is read in the VBA blocks through DAO (data access objects) links. This allows us to easily create/use the facility data in other design and analysis applications. Peters *et al.* (1996), Smith *et al.* (1998), and Peters and Smith (1998) provide a complete description of this integration between Arena and Access.

3.2 Real-Time Command and Control through a Guard Console

By incorporating the real-time communication features of Arena RT, the simulation system can begin to move to level 5 analysis. At this level, humans play the role of "commanders" for the guards, the adversaries, or both. The system structure for the real-time component is shown on the right side of the simulation block in Figure 3. The Guard Console provides a graphical view of the facility and displays real-time sensor state and entity location information. The simulation updates this information on a user-defined interval. In addition, the human commander can instruct entities to move to specified locations represented by nodes in the facility graph. These instructions are sent to the simulation through the VBA code responsible for routing entities. When an entity currently located at node *i* is instructed to go to node *j*, the VBA code determines the shortest path between nodes *i* and *j* and implements the path as the entity's route.

The level 5 analysis in this context is useful in several ways. The system can be used to analyze the system performance under the control of experienced command personnel providing for very complex (even cooperative) behavior. The system can be used as a simulator in training exercises where the goal is to train command and control personnel. In addition, the system can be used to identify and formalize command and control behaviors that can be later coded and included as part of the level 3-4 analyses.

3.3 Integration with a VR System

In addition to the guard console connection, the simulation system also integrates with a virtual reality (VR) system, providing the ability for non-analysts to visualize scenarios run in the simulation. The simulation model sends entity location and state and sensor state information to the VR system through a network connection on a user-defined interval (typically once per second). The VR system uses this information to dynamically render the virtual environment and avatars. The current implementation provides for "3rd party" viewing of the simulation in the VR environment. That is, people can move through the virtual environment and observe intruders and guards. Future work in this area will focus on allowing interaction with the running simulation through the VR environment. That is, a person will be able to "play" an intruder or guard in the virtual environment.

4 CONCLUSIONS

This paper describes the use of a discrete event simulation-based model for the design and control of physical protection systems. The system bridges the gap in the levels of analysis detail that exists in current security analysis tools. The key technology is the ability to have a communications mechanism between a running simulation and one or more external programs. A prototype security analysis tool was developed to allow human-in-the-loop interaction and to support a real-time connection to a virtual reality (VR) model of the facility being analyzed. This simulation tool is used for both facility analysis and design (in fast mode) and guard and supervisor training (in real-time mode).

REFERENCES

Drake, G. R. and Smith, J. S., "Simulation System for Real-time Planning, Scheduling, and Control," *Proceedings of the 1996 Winter Simulation Conference*, December 1996, Coronado, CA.

Matter, J.C., SAVI: A PC-Based Vulnerability Assessment Program, SAND88-1279, July 1988.

Peters, B. A. and Smith, J. S., "Real-time Simulation-Based Shop Floor Control," *Proceedings of ArenaSphere '98*, June 1998, Pittsburgh, PA.

Peters, B. A., Smith, J. S., Curry, J., LaJimodiere, C. and Drake, G. R., "Advanced Tutorial – Simulation-Based Scheduling and Control," *Proceedings of the 1996 Winter Simulation Conference*, December 1996, Coronado, CA.

Smith, J. S., Peters, B.A., Curry, J., and Gupta, D., "Prototype Software Model for Designing Intruder Detection Systems with Simulation," *Aerosense '98 – Modeling, Visualization, and Simulation Conference*, April 13-17, 1998, Orlando, FL.

AUTHOR BIOGRAPHIES

SABINA E. JORDAN is a Principal Member of the Technical Staff at Sandia National Laboratories. She has been involved in safeguards and security R&D for the past ten years, including security vulnerability assessment tool development. She holds a B.S. degree in Computer Engineering from the University of New Mexico and an M.S. degree in Electrical Engineering from the University of Southern California.

MARK K. SNELL is a Principal Member of the Technical Staff at Sandia National Laboratories. He has been involved in developing and using security vulnerability assessment tools since 1982. He holds a B. S. degree in Economics from Syracuse University and a Ph.D. in Operations Research from Cornell University.

MARCELLA M. MADSEN is a Distinguished Member of the Technical Staff at Sandia National Laboratories. She was project leader for development of risk assessments for radioactive material transportation in the 1980s and has been involved in program management for nuclear weapon security for the past 6 years. She holds a B. S. degree in Mathematics from the University of Minnesota and an M.A. in Mathematics from the University of New Mexico.

JEFFREY S. SMITH is an Associate Professor in the Department of Industrial Engineering at Texas A&M University. His research interests are in computer-aided manufacturing, shop floor control, and discrete event simulation. He holds a BSIE degree from Auburn University and MSIE and Ph.D. degrees from Penn State University.

BRETT A. PETERS is an Associate Professor in the Department of Industrial Engineering at Texas A&M University. His research interests are in facility location and design, computer-aided manufacturing, and logistics. He holds a BSIE degree from the University of Arkansas and MSIE and Ph.D. degrees from the Georgia Institute of Technology.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.