Adapting an Agent-Based Model of Socio-Technical Systems to Analyze System and Security Failures

Christine Cunningham*
MIT Lincoln Laboratory
Lexington, MA
Christine.Cunningham@ll.mit.edu

Antonio Roque
MIT Lincoln Laboratory
Lexington, MA
Antonio.Roque@ll.mit.edu

ABSTRACT

In this paper, an agent-based model of a socio-technical system is built by modifying the components and domain of an existing model. This includes adapting an agent decision-making and communication model, a task workflow model, and a performance model. The goal is to enable the modeling of scenarios related to critical infrastructure system failures, both those that are intentional (e.g. computer security violations) and those that are not.

This scenario models the non-intentional complex system failure which resulted in the 2003 Northeast Blackout, focusing on the people, tools, and organizations involved. The adapted model formalizes and explains the Blackout's causes. An agent-based framework allows us to implement the model and perform multiple iterations of the simulation. We study features of the output, and conduct experiments adjusting the assignment of tasks to increase the system's robustness to failure.

Categories and Subject Descriptors

I.6.3 [**Simulation and Modeling**]: Applications

General Terms

Security

Keywords

Agent-Based Simulation, Teamwork, Social Simulation

1. INTRODUCTION

Computer security, and system robustness in general, is not just a problem of machines but of teams of people using machines.[3][7][21] Part of the problem involves external attackers (who are themselves often in teams, using machines), and part of the problem involves insider threats and user errors.[17] Because of this, security researchers build and study computational models of humans working with computers[9][2][20], including for cyber range events, when these models are used to generate traffic.[14][5]

We study system and security failures using agent-based models of socio-technical systems. Agent-based modeling[13] involves autonomous and proactive programs which communicate peer-to-peer. Socio-technical system approaches involve models of humans, their organizations, the tools they use, and the interaction between all of these.[19] Agent-based models of socio-technical systems have previously been applied in the context of air traffic systems of air traffic controllers and pilots[19], economic production/consumption networks[16], and more.

To perform our modeling, we identified a scenario involving physical infrastructure which would make a good example of how to model general traffic generator scenarios. The specifics of the scenario are incidental to us in the following respect: what is most important is that the scenario is a realistic system since it was based on an actual event and a large-scale environment. The scenario enables us to develop a methodology grounded in reality which allows us to simulate different missions. It also enables us to change user behaviors (and therefore network traffic) based on changes in complex user profiles.

In this paper, we adapt an agent-based model of socio-technical systems developed by Crowder et al.[4]. Crowder et al.'s model was developed to study engineering team work over a scale of several weeks; our interest is in security-related scenarios that develop over much shorter time-frames, so adapting the model is a primary concern. We recognize the importance of the team-related aspects of system and security failures: the agents involved in decision-making, the tools they use, and the ways they communicate. We therefore selected Crowder et al.'s model as a starting point to represent a scenario as an agent-based model, and used simulation experiments to study the agents, their tools, and the ways they interact.

2. BLACKOUT SCENARIO

2.1 Motivation

System failures are disruptions of normal functions, and security failures can been seen as *intentional* system disruptions.[22] For this reason, we study general system failures with a special interest in those failures that were or could be caused intentionally.

Finding a good example of a socio-technical system for such a scenario is challenging because of the need for a scenario that is well-documented, realistic, and relevant to critical infrastructure. Security and privacy concerns make finding such information difficult. Our approach is to consider

*This work was initiated while affiliated with Williams College.

Distribution A: Public Release. This work is sponsored by the Test Resource Management Center under Air Force Contract #FA8721-05-C-0002.
Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

the general security case, identify a past case whose causes could plausibly have been intentional and computer-related, and to study the effects of those causes.

The scenario we identified is the 2003 Northeast Blackout, which was the largest blackout in North American history. It affected 50 million people (including over 20 million in the New York City and 8 million in the Toronto metropolitan areas) and cost an estimated 6 billion dollars[11], revealing vulnerabilities in the infrastructure and management of the electrical power grid.

Electrical power grids are generally considered one of a nations's *Critical Infrastructure and Key Assets*, whose effectiveness and security are vital to maintain.[12] Power grids are a possible target for attack,[15] and therefore of interest from a security perspective.

To follow is a description of those key participants and events which are important to model for our purposes. Unless otherwise stated, the description details in Section 2 are taken from the North American Electric Reliability Corporation's "Final Report on the August 14, 2003 Blackout in the United States and Canada."[18]

2.2 Key Participants

Figure 1 shows the organizational structure of the socio-technical elements contributing to the 2003 Northeast Blackout.

Reliability Coordinators (RCs) cover multi-state regions; they are responsible for monitoring and coordinating their multiple **Control Areas** (CAs) as well as the CAs of their neighbors. RCs must provide yearly, monthly, and daily energy consumption predictions, as well as contingency analyses for managing electrical flow during unanticipated situations. One of the tools that RCs use is a **state estimator**, which enables these contingency analyses.

CAs have real-time responsibilities such as ensuring the *n-1* criterion: that given the current conditions, instability or cascading outages would not occur as a result from one single contingency. To do so, CAs are responsible for manual and automatic load shedding, and for coordinating with their RCs. In an emergency situation, CAs must also inform their neighboring systems of the potential impact of their condition on the stability of the grid. CAs are not centralized by city or state and may share jurisdiction over power lines, which in the 2003 Northeast Blackout led to communication challenges. One of the tools that CAs use is an **alarm system** to help detect dangerous load conditions.

Power grid operations involve a great deal more complexity than described here. For our purposes, we focus on those elements of the socio-technical system of people, machines, and their interactions that contributed to the 2003 Northeast Blackout. As shown in Figure 1 and described below, the main problems occurred in RC1's state estimator, CA1's alarm system, and in communication between CA1 and CA2.[1]

2.3 Key Events

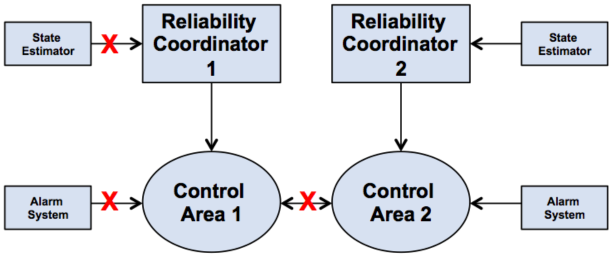


Figure 1: Organizational structure of relevant entities in the 2003 Northeast Blackout.

Start of the day, August 14th, 2003: electricity demand had been higher than predicted for the previous two days. Four to five vital capacitors had been misclassified as non-vital and taken off-line for upkeep, and a few lines were unexpectedly down, causing CA1 to operate close to capacity.

12:15pm: a line in CA1's contingent went down; this in itself was not a major problem, but a human operator then forgot to restart CA1's state estimator, which then remained unavailable.

2:14pm: a hidden race condition in the CA1's alarm system surfaced, causing the alarm system to fail. CA1 operators were unaware that the alarm system was down.

2:32pm: CA2 called CA1 about a tripped line that they shared. CA1 did not follow up on the call because their alarm system showed no problems, and they were still unaware of the alarm system's failure.

3:05pm: a power line in CA1 failed due to contact with trees. CA1 was unaware of this power line failure (due to their alarm system failure), and even if they had been aware would not have had a contingency plan (due to RC1's state estimator failure).

3:09pm: an RC1 operator made an error that prevented the state estimator from going back online.

3:19pm: CA2 called CA1 again about the problem they saw. Again, CA1 did not follow up on the call because they saw no problems.

3:23pm: two more power lines failed after contact with trees. CA1 operators became aware of problems, but had limited options due to the short time frame and lack of contingency plans. The electrical load from the failed lines was shifted onto a single line, which overloaded and started a cascade that shortly afterwards resulted in the blackout.

2.4 System Failure and Security

The 2003 Northeast Blackout was caused by a combination of human, system, and communication failures. This in itself is worth studying, but we are particularly interested in its potential for studying the vulnerabilities of socio-technical systems to computer network attacks. To this end, for each system failure we identify a computer attack that could have caused that system failure.

- CA1's state estimator unavailability was due to a mode confusion caused by human error. However, in a computer attack scenario this could have been due to malware or insider sabotage.
- RC1's alarm system malfunction was due to a race condition in its code. There is no reported indication

[1] RC1 is the Mid-continent Independent System Operator, based in midwestern USA and Manitoba, Canada; RC2 is the Pennsylvania-New Jersey-Maryland Interconnection; CA1 is First Energy; and CA2 is AEP. Both CAs primarily (though not exclusively) cover regions in the state of Ohio. For clarity, we use these role abbreviations rather than actual organization names elsewhere in this paper.

that this race condition was intentionally introduced, but the same effects could have been introduced by malware.

- The tripped wires were due to tree contact caused by unusually hot weather and high demand, which caused lines to heat and sag more than usual, but the Aurora Test suggests that such an effect is achievable by computer attack.[15]
- The lack of communication between CA1 and CA2 was due to a low trust in each other's alarms, but the same effect could have been produced by a denial-of-service attack on the communication system.

The 2003 Northeast Blackout was not the result of a computer attack, but each of the socio-technical system's failures could have been caused by a computer attack without changing the way that the system reacted. What we call our **Blackout scenario** therefore is an extraction of the socio-technical security vulnerabilities displayed during the 2003 Northeast Blackout.

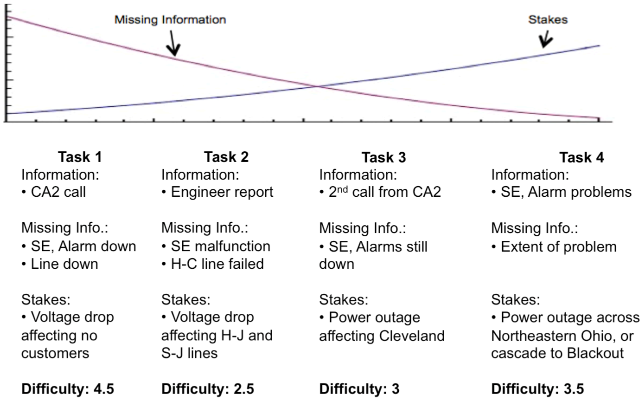
3. AGENT-BASED MODEL

3.1 Original Model

Given our need for realistic user agents when implementing the scenario, we next sought out an appropriate user model. We chose an agent-based model developed by Crowder et al.[4] which applied socio-technical systems theory to modeling work teams. Crowder et al. developed their model with concepts from psychology, management, and computer modeling, as well as with quantitative and qualitative data collected from multidisciplinary engineering teams. Their model describes how a task's requirements cause team members to communicate among themselves, and the cognitive mechanisms that integrate the results of that communication.

The Crowder et al. model includes a Task Workflow Model which describes the steps required to complete a task, dependencies between the steps, the difficulty of each step, and the team member responsible for completing the task. Additionally, an Agent Model with components such as *Trust* and *Shared Mental Models* uses a set of equations to describe the interaction of those components while performing a task. The model produces a set of completion and working times for performing the task, as well as a measure of task quality. Finally, a Communication Model describes the way that agents request information as needed, to increase their ability to complete a task step. More details about these models is contained in Section 3.2 where we describe adapting them to our scenario.

Crowder et al.'s model provides "a general framework for modeling teamwork...useful for modeling team work in many different domains."[4] Crowder et al. state that an optimal application of their model involves conducting questionnaire studies and performing regression analyses with subject matter experts in the domain of interest, to determine how to modify the model's equations. However Crowder et al. also state that "the current model still provides a promising start for the simulation of complex team working, in engineering organizations and more generally."[4] Our approach is to therefore to exploit the generality of the framework by using it as a starting point as suggested, adapting it to our domain of interest, and exploring the analytical power this provides in general to the scenario. A secondary goal is determining the extent to which this level of adaptation fulfills our long-term research needs, compared to the need to perform a full domain study.



Task 1	Task 2	Task 3	Task 4
Information: • CA2 call	Information: • Engineer report	Information: • 2^{nd} call from CA2	Information: • SE, Alarm problems
Missing Info.: • SE, Alarm down • Line down	Missing Info.: • SE malfunction • H-C line failed	Missing Info.: • SE, Alarms still down	Missing Info.: • Extent of problem
Stakes: • Voltage drop affecting no customers	Stakes: • Voltage drop affecting H-J and S-J lines	Stakes: • Power outage affecting Cleveland	Stakes: • Power outage across Northeastern Ohio, or cascade to Blackout
Difficulty: 4.5	**Difficulty: 2.5**	**Difficulty: 3**	**Difficulty: 3.5**

Figure 2: Determining Task Difficulty

3.2 Adapting the Model

3.2.1 Task Workflow Model

The focus of this study is in the socio-technical system involved in reacting to the problems that led up to the 2003 Northeast Blackout; in other words, the humans and technologies shown in Figure 1, their structure, and the way they worked together. In particular, the moments in which agents in the system, confronted with an indication of grid failure, either failed to resolve the difficulty (as happened historically) or succeed in resolving it. Because of this, we model the Task Workflow Model of our Blackout scenario as a series of interactions that led to the blackout as described in Section 2.3.

Figure 3 shows the **Task Workflow Model**. It includes 4 tasks, each which has an agent assigned to it (either CA1 or RC1) and a *Task Difficulty* value between 0 and 5, following Crowder et al.'s use of semantic labels on that scale.

The Task Difficulty was not provided by the Blackout Report, so we had to determine it as part of developing the model. To guide this process, we produced qualitative descriptions of the information available to the agents, as well as of the stakes involved. As the scenario progresses, more information is available to the agents about the nature of the problem; in that way the tasks are easier. As the scenario progresses the stakes are higher, though, so in that way the tasks are harder. This analysis produced a quantitative estimate based on the dynamic between these qualitative factors, as shown in Figure 2.

In the 2003 Northeast Blackout, each task was unsuccessfully completed, leading to the cascade and blackout. In our Blackout scenario, the team has the opportunity to successfully complete any of the tasks, resulting in a better outcome and ending the scenario. Although CA2 is not assigned a workflow task, CA2 influences CA1's *Competency* during task communication as described below.

Task 1 represents the interaction at 2:32pm in which CA1 received a telephone call from CA2 about a tripped line that they shared. In the 2003 Northeast Blackout CA1 did not

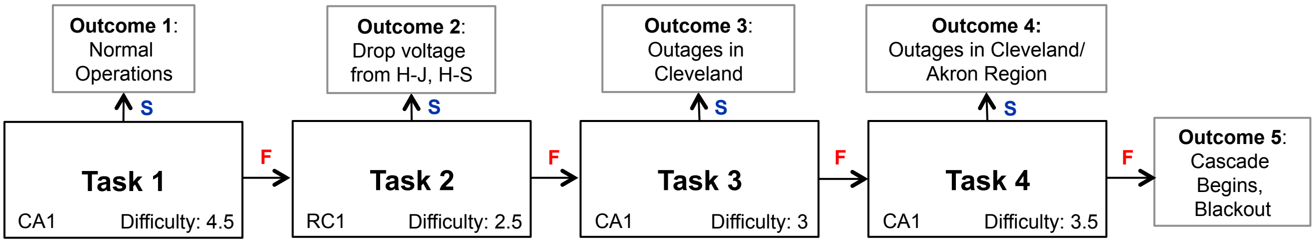

Outcome 1: Normal Operations
Outcome 2: Drop voltage from H-J, H-S
Outcome 3: Outages in Cleveland
Outcome 4: Outages in Cleveland/ Akron Region
S
S
S
S
Task 1
CA1
Difficulty: 4.5
F
Task 2
RC1
Difficulty: 2.5
F
Task 3
CA1
Difficulty: 3
F
Task 4
CA1
Difficulty: 3.5
F
Outcome 5: Cascade Begins, Blackout

Figure 3: Blackout Scenario Task Workflow Model. S=Success, F=Failure

follow up on this issue because their alarm system, which was malfunctioning, did not register the problem. The task involves following up on the problem anyway and managing the problem without the benefit of the state estimator, which was also malfunctioning. This is a difficult task, as it involves determining that there is in fact a voltage problem due to the line being down and that the alarm system and state estimator are malfunctioning, and either resolving the problems with the alarm system and state estimator or solving the voltage problem without the state estimator to provide a contingency plan. Because of this, Task 1 is given a high difficulty value. Resolving the task correctly would result in normal operation (beyond the relatively minor line being down) and the end of the scenario; not resolving the task correctly leads to the next task.

Task 2 represents the event at 3:09pm in which a RC1 operator made an error which prevented the state estimator from coming back online. This task involves correctly bringing the state estimator back online, checking for possible problems as a consequence, identifying the power line failures so far, and remedying the situation. Bringing the state estimator back online would not have been very difficult, but subsequently identifying the power line failures without CA1's alarm system would be more challenging. Furthermore, the best solution to this situation involved higher stakes: dropping voltage from several other lines which would have left other customers without power. Because of this the problem is given a moderate difficulty level.

Task 3 represents the interaction at 3:19pm in which CA1 received a second telephone call from CA2 about the line problems CA2 was seeing. Once again, CA1 did not follow up because their alarm system was still malfunctioning. Solving this task would involve the same subtasks as Task 1, but would be somewhat less difficult because the fact that CA2 had called twice provided a greater amount of evidence of a problem. However, successfully resolving the problem involved higher stakes: at this point, the best outcome possible was to drop voltage resulting in power outages in the city of Cleveland, Ohio.

Task 4 represents the status at 3:23pm, when CA1 finally becomes aware of the problems and has one last chance to resolve the situation. Resolving the problem includes deciding how much voltage to drop, without the benefit of RC1's state estimator, under a very tight time constraint. The stakes are high, as successfully completing the task nonetheless results in outages in the cities of Cleveland and Akron and their surrounding region, and unsuccessfully completing the task results in the cascade and blackout affecting 50 million people.

3.2.2 Agent and Communication Models

The Crowder et al. model was developed from an engineering domain that took weeks and months to perform, rather than the shorter time-frame involved in the Blackout scenario. This led us to make several changes to our **Agent Model**.

The Crowder et al. Agent Model produced several outputs: the *Completion Time* tracks how long it takes an agent to finish a particular task; these are combined from all tasks and agents to produce a Total Completion Time. The *Working Time* tracks how much time the individual agent spends working on a task; these are combined from all tasks and agents to produce a Total Completion Time. The *Quality* describes the degree of excellence of the task once finished; these are combined from all tasks and agents to produce a Total Quality.

In our Blackout scenario, the agents were working under strict time constraints. They had a short amount of time to decide how to resolve problems they encountered, and at the end of time they had to address the problem, such as dropping power or shifting loads, but even if the operators dropped power, they might not drop enough. Doing nothing before time ran out was one way of handling the task, though in our scenario this was always the wrong decision. Because task completion time was a constraint, as shown in Figure 4 we removed the *Completion Time* and *Working Time* components, as well as the *Availability*, *Learning Time*, *Response Rate*, and *Communication Frequency* components which likewise were dependent on longer timescales.

Next, we considered the *Shared Mental Models*, *Motivation*, and *Communication Frequency* components. The main distinctions between those components was that Shared Mental Models did not directly affect Quality, and Competency affected Communication but Motivation did not. However, the distinction between these components is more important in Crowder et al.'s use case than in ours: for example, they may be interested in changing these values to determine whether it is more cost-effective to invest in increasing team Motivation, or team Shared Mental Models. Furthermore in the Crowder et al. model, the equations that drove the algorithms behind these components contained numerous coefficients derived from regression analyses of Likert-scale questionnaire data taken from their engineering domain. To limit dependence on these domain-specific coefficients, we therefore merged the Shared Mental Models and Quality components into the Competency component, and included a normally (Gaussian) distributed value with a standard deviation of 1 in the equation for Competency as a way of partially substituting for their effect.

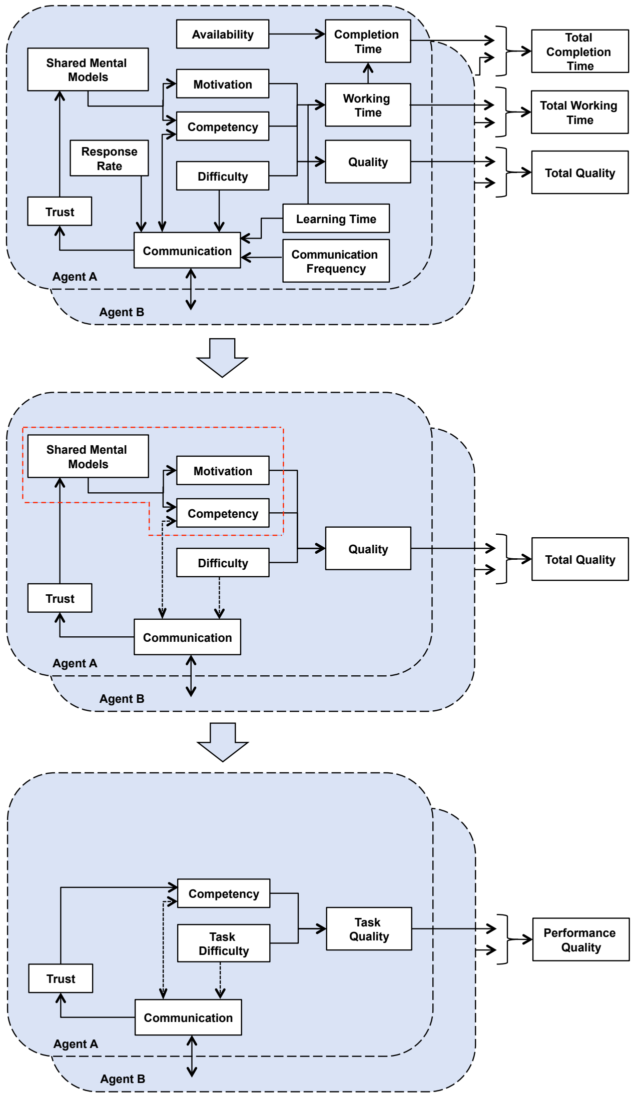


Figure 4: Adapting the Agent Model. Top: components from Crowder et al. model. Middle: after removing time-related components not relevant to our domain. Bottom: after merging competency-related components.

The components of our final Agent Model are shown at the bottom of Figure 4.

Crowder et al.'s version of *Trust* is an input value, modified by the success or failure of the agent's previous interactions with a team member on longer-lasting tasks. Because our tasks and time scales are different, our version of Trust is an input value on the 0-5 scale to produce the base trust τ_b, modified by a normally (Gaussian) distributed value υ_τ with a standard deviation of 1 to produce a working trust τ_w.

$$\tau_w = \tau_b + \upsilon_\tau \tag{1}$$

Crowder et al.'s version of *Competency* is an input value, which is increased by interactions with other team members. Our Blackout scenario agents also have a base input value C_b modified by a normally distributed value υ_C. This may also be modified by an increase in competency due to interactions with other team members δ_C to produce a working competency C_w.

$$C_w = C_b + \upsilon_C + \delta_C \tag{2}$$

We use Crowder et al.'s equation for δ_C as shown in Equation 3.2.2, and similarly cap the possible Competency gain to 0.3.

$$\delta_C = \frac{15 + 3(C_p + C_r)}{100} \tag{3}$$

Crowder et al.'s model assumes that an agent will continue to seek communications with other team members (thereby increasing Competency) until the agent has a Competency sufficient for the task difficulty. In the Blackout scenario, the CA1 agent has a chance to gain competency from the CA2 agent, but does not seek out further competency gain due to the Task time scale and absence of other agents to refer to. Therefore although our Competency can in principle drive Communication, in this scenario's tasks it does not.

In our scenario, for a task n the *Task Quality* Q_n is a (0,1) value describing whether the voltage problems were *completely* resolved, because partial solutions did not stop the process leading towards blackout. This is determined by comparing the agent's working Competency to the Task Difficulty D_n. If $C_w \geq D_n$, then the Task Quality is 1 (success); otherwise it is 0 (failure).

The overall Performance Quality Q_P then expresses the performance on the i tasks of the Blackout scenario as an integer between 0 (for blackout) to 5 (for best outcome).

$$Q_P = \frac{5}{4}(4 - \sum_{n=1}^{i}(1 - Q_n)) \tag{4}$$

Finally, our **Communication Model** is shown in Figure 1. Communications are tied to the workflow model: each Task defines a communication that occurs between agents as part of their involvement in the system.

3.3 Implementing the Model

Following Crowder et al., we implemented the model using JADE, the Java Agent DEvelopment framework.[1] JADE is a Java based open source Agent-Oriented Middleware whose strengths for our purposes are its template behaviors and its communication structure. JADE has been used by other researchers in a variety of settings: to develop frameworks assisting in collaborative design[6], to built platforms for collecting feedback from patients for researchers in healthcare settings[8], and to to study coalition structure generation in distributed algorithms,[10] for example.

JADE complies with the FIPA message sending protocol. Messages are asynchronous and all agents keep a queue of messages. Sending a message involves specifying the receiving agent's agent identifier. For the purpose of this project, the various agents sent and received messages based on sender agent identifier and a message type, though this could have been improved by the use of JADE ontologies. Because of the required behaviors for the power grid agents (executing their required jobs at regular intervals as well as always being ready to respond to an event or communication from another agent), we used a parallel behavior that included a ticker behavior and a cyclic behavior.

4. SIMULATIONS

4.1 Baseline Simulation

We ran 12,500 iterations of the Blackout scenario to build a baseline. We ensured an iteration through the parameter space of all possible inputs by cycling through the base competency settings as shown in Table 1. In this way we are sure to explore all possible combinations of team competencies, rather than be tied to a representation in which the teams are all of mid-level competency or high-level competency.

Table 2 shows the results, along with the number of times each outcome occurred. The outcomes described here are those shown in Figure 3 and described in Section 3.2.1: Outcome 1 is *Normal Operations*, Outcome 2 is *Drop Voltage from H-J and H-S lines*, Outcome 3 is *Outages in Cleveland*, Outcome 4 is *Outages in Cleveland/Akron region*, and Outcome 5 is *Cascade Begins, Blackout*.

As a metric for determining the efficiency of the agent system, we used the percentage of iterations in which $Q_p > 0$. This is the percentage of Non-Blackouts, the number of times the scenario ended in Outcomes 1-4 (i.e. was resolved by dropping voltage from lines or regions, even if resulting in smaller local outages) instead of reaching Outcome 5, (i.e. ended in a cascade leading to a major blackout as in the 2003 Northeast Blackout.) The total and percentage of Non-Blackouts is shown in the last row of Table 2.

Surprisingly, the baseline data set included no outcomes in Outcome 4. After examining the detailed logs, we identified the dynamic that caused this situation: Task 3 acted as a filter to remove any Agents who would be able to succeed at Task 4. Figure 5 shows the last two tasks and final three outcomes of the Task Workflow Model, along with the characteristics of agents who reach each outcome. Note that Task 3 has a difficulty of 3, so any agent whose competency is greater than or equal to 3 will succeed at that task[2] and exit the scenario at that point, and those agents that fail will proceed to Task 4. Therefore, the only agents who are left to reach Task 4 are those with a competency less than 3. However, Task 4 has a difficulty of 3.5, so all agents who reach that task will fail at it. Failing at Task 4 leads to Outcome 5 so all agents who reach Task 4 will end up in Outcome 5. Succeeding at Task 4 leads to Outcome 4, but all agents fail at Task 4, so no agent will reach Outcome 4.

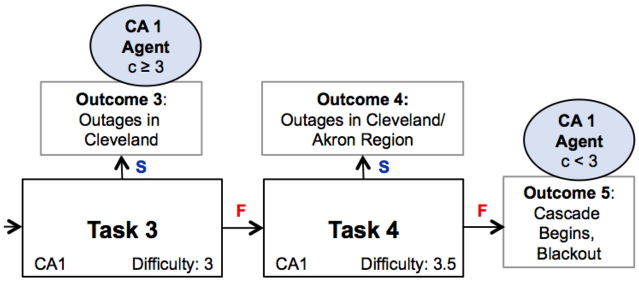


Figure 5: Dynamic behind absence of Outcome 4 in Baseline Simulation

Insight into the structure of the scenario is one of the benefits of this type of simulation. In this case we have identified that outcomes are impossible if the following requirements are true: tasks are strictly sequential; tasks consistently increase in difficulty; an earlier task includes an outcome that ends the simulation; task completion depends on directly comparing task difficulty with an unchanging agent characteristic.

Having understood these requirements, we can seek ways to modify the functioning of the socio-technical system to improve the overall quality of outcomes. Keeping the task definition constant, we identified the requirement of comparing task difficulty to unchanging agent characteristic as being the most promising for change. We used this model's notions of teams and task assignments to perform experiments in varying the policy which determines what tasks are delegated to what agent.

4.2 Delegation Experiments

To explore a policy which might improve upon the situation where Outcome 4 never occurs, we implemented an agent team policy in which, upon reaching Task 4, that task is delegated to another agent. This was effected by replacing the CA1 agent with a CA1 agent with a different Competency. The rationale for this is that CA organizations are actually made up of numerous agents. We were previously assuming that a single CA agent would handle every task, but it is equally reasonable to assume that a CA organization would have a set of agents, and that the organization's policy would be to randomly assign tasks that have been received.

We hypothesized that this policy would improve the Total Non-Blackout metric to a statistically significant amount, which it did with a p-value<0.0003 on a simulation of 3125 iterations; the data is shown in the Delegation 1 column of Table 2.

However, this is slightly unrealistic because it assumes that the CA team has a reliable way of knowing that they were in a task that should be delegated to another agent. We therefore hypothesized that delegating *each* of the tasks in the scenario to different agents would significantly increase performance. We implemented this as instantiating a new agent (with a new competency rating) for each task in Figure 3.

[2] As described in Section 3.2.2, task success is determined by comparing an agent's Competency to the task's Task Difficulty to determine a Task Quality indicating success or failure.

Table 1: Experiment Settings

	CA1 **Trust**	CA1 base	CA2 base	RC1 base
	in CA2	**Competency**	**Competency**	**Competency**
1	VH (4.5)	VH (4.5)	VH (4.5)	VH (4.5)
2	VH (4.5)	VH (4.5)	VH (4.5)	H (3.5)
3	VH (4.5)	VH (4.5)	VH (4.5)	M (2.5)
...	...	...	...	...
625	VL (0.5)	VL (0.5)	VL (0.5)	VL (0.5)

Table 2: Experiment Simulations

	Baseline	**Delegation 1**	**Delegation 2**
Total Iterations	12500	3125	3125
Outcome 1	1273 (10.2%)	314 (10.0%)	630 (20.2%)
Outcome 2	4495 (36.0%)	1125 (36.0%)	1000 (32.0%)
Outcome 3	3082 (24.7%)	757 (24.2%)	829 (28.5%)
Outcome 4	0 (0%)	377 (12.1%)	243 (7.8%)
Outcome 5	3650 (29.2%)	552 (17.7%)	360 (11.5%)
Total Blackout	3650 (29.2%)	552 (17.7%)	360 (11.5%)
Total Non-Blackout	8850 (70.8%)	2573 (82.3%)	2765 (88.5%)

This new approach did indeed improve the Total Non-Blackout metric to a statistically significant amount, with a p-value<0.0003 on a simulation of 3125 iterations; the data is shown in the Delegation 2 column of Table 2. Note also the increase in the best possible outcome of Outcome 1, and the decrease in the two worst Non-Blackout outcomes of Outcomes 4 and 5.

5. DISCUSSION

5.1 Future Work

Adapting this model to our domain of interest involved making a number of assumptions. For example, after accepting the validity of Crowder et al.'s "general framework," we modified the Agent Model as described in Section 3.2.2, removing or merging several components and changing their equations to remove domain-specific constants. Although we did so methodically, our connection to the empirically-derived foundations of Crowder et al.'s original work nevertheless has become weaker. We are therefore interested in further examining these assumptions and validating them where necessary, especially in the impact of these assumptions when adapting the general framework to another domain and time scale.

Crowder et al.'s framework is general enough to formally specify the causes of a system failure, as is shown in Section 2.2 and Figure 1. The baseline simulation and delegation experiments in Section 4 show that the simulation explains a range of possible outcomes and their dependence on input variables such as agent competence. The simulation also provides insight into task dynamics such as the absence of Outcome 4 described in Section 4.1, and possible ways of improving probable outcomes as described by the delegation experiments in Section 4.2. Although this model does not provide absolute proof that such policies would improve robustness of the system, it provides possibilities for further exploration. We look forward to seeking out new domains to apply this approach, and formalizing the applicability of the results.

In terms of the secondary goal described in Section 3.1, we can conclude that this level of adaptation looks promising for our research needs, which are in the area of building teams of networked user agents for use in security simulations, similar to work by Blythe et al.[2] and Wright et al.[20] In such a domain, the central aspect of validity varies based on the needs of the simulation: in some cases it may involve measures of task accomplishment, in others it may be enough to simply produce network traffic with a fidelity sufficient to a guideline. Even without performing domain-specific organizational research, we therefore find that Crowder et al's model is indeed a good basis for a future models whose parameter tuning and validation will depend on the specifics of their intended use.

5.2 Conclusion

We have adapted an agent-based model of a socio-technical system to enable the modeling of scenarios related to critical infrastructure system failures, both intentional and not. This enables us to study the complex interactions between human participants, their organizational structure, the tools being used, and the nature of the domain.

These interactions inform a methodology that allows us to provide high-fidelity models of network traffic for use during cyber range events. We have shown that changing complex user profiles can result in changing user behavior, which in the context of a cyber range tool such as LARIAT would change network traffic in a principled, empirically-grounded way.[14][5]

We implemented the model in a scenario mimicking the 2003 Northeast Blackout. Although the Blackout had purely technical aspects (such as cascade's spread through low apparent voltage), the events leading up to the cascade are best understood by modeling the people involved, the tools they used, and their organizational structure. We experimented with manipulating task delegation to increase overall performance. We identified a way of decreasing the percentage of simulations leading to a full blackout; this may be seen as increasing the probability that a given socio-technical system will successfully manage a situation that risks leading to a blackout. We plan further work on this approach to continue testing its applicability.

Acknowledgment

This work is sponsored by the Test Resource Management Center under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

REFERENCES

[1] F. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE.* Wiley, 2007.

[2] J. Blythe, A. Botello, J. Sutton, D. Mazzocco, J. Lin, M. Spraragen, and M. Zyda. Testing cyber security with simulated humans. In *Proceedings of the Twenty-Third Innovative Applications of Artificial Intelligence Conference*, 2011.

[3] L. F. Cranor. A framework for reasoning about the human in the loop. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 1:1–1:15, Berkeley, CA, USA, 2008. USENIX Association.

[4] R. M. Crowder, M. A. Robinson, H. P. Hughes, and Y.-W. Sim. The development of an agent-based modeling framework for simulating engineering team work. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 42(6):1425–1439, 2012.

[5] S. K. Damodaran and J. M. Couretas. Cyber modeling & simulation for cyber-range events. In *Proceedings of Summer Computer Simulation Conference*, 2015.

[6] J. D. Faus and F. Grimaldo. Infraworld, a multi-agent based framework to assist in civil infrastructure collaborative design. In *11th International Conference on Autonomous Agents and Multiagent Systems*, 2012.

[7] A. E. Howe, I. Ray, M. Roberts, M. Urbanska, and Z. Byrne. The psychology of security for the home computer user. In *2012 IEEE Symposium on Security and Privacy (SP)*, pages pp 209–223, 2012.

[8] Z. M. Ibrahim, L. F. de la Cruz, A. Stringaris, R. Goodman, M. Luck, and R. J. Dobson. A multi-agent platform for automating the collection of patient-provided clinical feedback. In *2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015.

[9] V. Kothari, J. Blythe, S. Smith, and R. Koppel. Agent-based modeling of user circumvention of security. In *Proceedings of the 1st International Workshop on Agents and CyberSecurity.* ACM, 2014.

[10] T. Michalak, J. Sroka, T. Rahwan, M. Wooldridge, P. McBurney, and N. R. Jennings. A distributed algorithm for anytime coalition structure generation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1, pp. 1007-1014. International Foundation for Autonomous Agents and Multiagent Systems*, 2010.

[11] J. Minkel. The 2003 Northeast Blackout - five years later. *Scientific American*, August 2008.

[12] J. Moteff and P. Parfomak. *Critical Infrastructure and Key Assets: Definition and Identification.* Congressional Research Service, October 1 2004.

[13] S. F. Railsback and V. Grimm. *Agent-Based and Individual-Based Modeling.* Princeton University Press, 2012.

[14] L. M. Rossey, R. K. Cunningham, D. J. Fried, J. C. Rabek, R. P. Lippmann, J. W. Haines, and M. A. Zissman. Lariat: Lincoln adaptable real-time information assurance testbed. In *Aerospace Conference Proceedings*, 2002.

[15] P. Shakarian, J. Shakarian, and A. Ruef. *Introduction to cyber-warfare: a multidisciplinary approach.* Elsevier, 2013.

[16] A. P. Shaw and A. R. Pritchett. Agent-based modeling and simulation of socio-technical systems. *Organizational Simulation*, pages 323–367, 2005.

[17] The Smart Grid Interoperability Panel - Cyber Security Working Group. Guidelines for smart grid cyber security: Vol. 1, Smart grid cyber security strategy, architecture, and high-level requirements. Technical report, National Institute of Standards and Technology, August 2010.

[18] US-Canada Power System Outage Task Force. Final Report on the August 14, 2003 Blackout in the United States and Canada: Causes and recommendations. Technical report, North American Electric Reliability Corporation, April 2004.

[19] K. H. van Dam, I. Nikolic, and Z. Lukszo, editors. *Agent-based modelling of socio-technical systems*, volume 9. Springer Science & Business Media, 2013.

[20] C. V. Wright, C. Connelly, T. Braje, J. C. Rabek, L. M. Rossey, and R. K. Cunningham. Generating client workloads and high-fidelity network traffic for controllable, repeatable experiments in computer security. In *Recent advances in intrusion detection*, pages 218–237. Springer, 2010.

[21] T.-F. Yen, V. Heorhiadi, A. Oprea, M. K. Reiter, and A. Juels. An epidemiological study of malware encounters in a large enterprise. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages pp 1117–1130, 2014.

[22] W. Young and N. Leveson. Systems thinking for safety and security. In *Proceedings of the 29th Annual Computer Security Applications Conference*, pages 1–8, 2013.