Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

MODELING AN EMERGENCY OPERATIONS CENTER WITH AGENTS

Margaret L. Loper
Bart Presnell

Georgia Tech Research Institute
347 Ferst Drive
Atlanta, GA 30332, U.S.A.

ABSTRACT

The Georgia Emergency Management Agency (GEMA) operates the State Operations Center, which is the centralized hub for all communications regarding state operations, with respect to emergency response. The Center provides three primary functions: Request for Information, Request for Assistance, and Media Relations. The Center has responsibility for the entire process of servicing an emergency, from instigation to communication to execution to ending service for the emergency. To accomplish this, the Center is set up like a production facility, which follows a well-defined process articulated in a plan. In this project, we worked with GEMA to develop a simulation of the Center. The simulation, called GEMASim, models the information flow among people working within the Center, as well as the flow of information coming into and leaving the Center. Data is presented on the initial experiments with GEMASim using two different scenarios.

1 INTRODUCTION

Many of today's current simulation models are designed to analyze individual systems and thus are unable to provide realistic, decision-quality information about the likely effects of terrorist acts on homeland security (NRC 2002). One method of model development that appears to offer significant potential for analyzing the complexities of counterterrorism applications is complex adaptive systems (CAS) and agent-based models (ABM). Complex adaptive systems involve phenomena that may be characterized by the interactions of numerous individual agents or elements, which tend to self-organize at increasingly higher levels. This process results in evolutionary, emergent, and adaptive properties that are not exhibited by the individual agents themselves (NRC 2002). A general rule for CAS is that we cannot accurately predict the actual outcomes of the actual system. However, we can create a model that accurately simulates the processes that the system will use in order to create a given output.

This research project is focused on applying CAS and ABM techniques to the Georgia Emergency Management Agency (GEMA) State Operations Center (SOC). The GEMA SOC can be described as a large-scale socio-technical system, i.e. a system consisting of a large number of entities (such as humans, machines, computer systems, etc.) interacting with each other in significant ways in order to accomplish a specific goal. The dynamic behavior of such a system, to a great extent, can be characterized by the individual behavior of entities within the system and the aggregate and emergent behavior of the entities interacting with each other.

An agent-based simulation with dynamically interacting agents was developed to evaluate both the individual behavior of the agents and the complex phenomena of the large-scale, complex socio-technical system. As the agents interact with each other, the simulation captures how their collective behaviors create the performance of the GEMA SOC.

2 BACKGROUND

GEMA's State Operations Center is a powerful, technology-driven tool that allows GEMA staff and other emergency coordinators to handle response and recovery actions in a central location (GEMA 2004a). The SOC provides GEMA personnel and other emergency coordinators with access to communication and information systems, including telephones, the Internet and e-mail, fax machines, printers, weather reports and live traffic information.

The operations support unit handles all requests for assistance—which could be from local or state government agencies—to help them handle a particular situation. When a community is overwhelmed by the effects of or is preparing to face a disaster, an emergency management agency (EMA) Director will call GEMA to request assistance in the form of personnel, equipment, or coordination efforts. Under normal circumstances, the operations support unit duty officers route any request for assistance to the appropriate state agency. If these requests increase to a level beyond the normal operating capacity, the GEMA Director will activate the SOC.

When the Director of GEMA activates the SOC, not only GEMA personnel but also emergency coordinators from other state agencies, the military, utility companies, and volunteer groups will come to the SOC, depending on the severity of the incident. These people staff the SOC 24 hours each day during an activation, allocating personnel and equipment to meet the needs of the affected areas.

3 CONCEPTUAL MODEL

The GEMA SOC is the centralized hub for all communications for state operations with respect to emergency response. The SOC can be thought of as a production facility; it has responsibility for the entire process of servicing an emergency. In this project only a subset of the SOC chain of command was modeled. These positions are described below (GEMA 2004b).

- SOC Chief – The SOC Chief oversees and manages the SOC during activation.
- Public Affairs Officer (PAO) – The PAO conducts public information activities at the state level prior to, during and following emergency/disaster situations.
- Intelligence Officer (IO) – The IO has the primary responsibility of maintaining up-to-date and timely information, known as situation reports of the emergency situation.
- Technical Expert – Technical experts assist the SOC staff with technical expertise associated with responding to an emergency request.
- Operations Chief (OC) – The OC establishes procedures for notification and response to all inquiries and incidents requiring prompt and appropriate attention.
- Operations Officer (OO) – The OO assists the SOC Chief with overseeing and managing the SOC during activation.
- Action Officer (AO) – The AO is the focal point where requests for assistance are taken, monitored and disseminated to agencies having primary responsibility for services requested, and where assistance actions are coordinated.
- State Agencies Emergency Coordinator (SA) – State agency emergency coordinators coordinate with the SOC staff and their respective agency in an expeditious manner during emergency operations.
- Communications Officer (CO) – The CO is responsible for receiving and documenting incoming calls.
- Call Takers (CT) –This position is activated when call volume for the AOs are overwhelming.

In the sections that follow, we will discuss how GEMASim was developed, including how an agent-based paradigm was used to model the SOC chain of command. Additionally, we will discuss how the processes that SOC personnel use to respond to emergencies were modeled, as well as the behaviors used to accomplish their tasks.

3.1 Agents

SOC personnel were modeled as individual agents. There were 68 agents modeled and they were divided into two types: internal to SOC and external to SOC.

The internal agents include SOC personnel that resided inside the physical structure of the operations center. In addition to personnel, the network and phone system were also represented as individual agents in the simulation. We did this in order to model the role of these systems in accomplishing activities. The external agents include SOC personnel in the field. For example, the EMA Director is a field agent that works for GEMA but resides in one of the eight areas around the state of Georgia. They are the first line of response in requesting services during emergency situations.

Each SOC agent also had an associated skill level. This is represented as a standard speed plus a variance. The standard speed has the following composition: 70% standard speed to perform activity (does not vary), 10% experience (vary), 10% speed (vary), and 10% duty time (varies based on simulation time). The skill level was used in the design of experiments in order to evaluate the effect of skill in different scenarios.

Another agent that is external to the SOC is the World. The World agent represents everything that is not related to the SOC, and can be thought of as where emergencies occur. To model this, the World is responsible for generating service requests and requests for information that are sent to the SOC. This is illustrated in Figure 1.

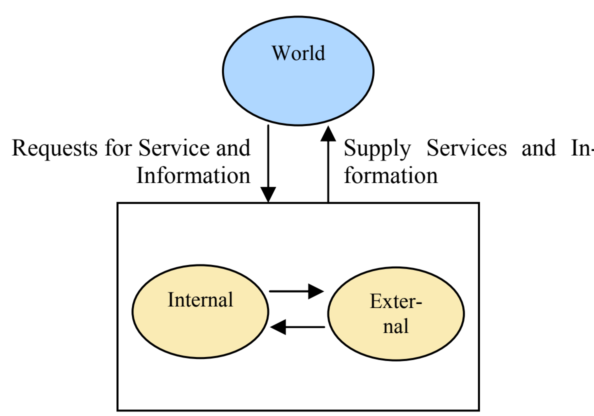

Figure 1: World Agent Representation

3.2 Activities

As mentioned previously, the SOC has the responsibility for the entire process of servicing an emergency, from taking the initial call to interacting with state agencies to supplying resources for the emergency. To service a request, SOC personnel follow defined processes which involve exchanging information among people in the SOC, as well as people external to the SOC. In the simulation, a process (or activity) is represented as a set of ordered events executed together to accomplish the activity. Activities can be simple (one event, unidirectional flow) or complex (many events, bidirectional flow – e.g., request/response). Each agent performs one or more activities and the life of the agent is defined by the set of activities it performs and the order in which the activities are performed.

Each activity was developed using GEMA processes and "stories" of past experience (Brown 2003). The activities were designed using state diagrams which depict the flow of information between individuals. Figure 2 illustrates a complex activity chart. The circles represent the agents and the lines connecting them indicate the flow of information required to accomplish the activity. Above each line is the message name used to convey information in the simulation.

3.3 Behaviors

Behavior is the process the agent goes through from the receipt of an input (sensing) to the sending of a response (acting). In GEMASim behaviors are rule-based; agents perceive their environment through incoming messages and respond by sending messages.

There were 56 different behaviors defined for the simulation. For all behaviors, agents are in one of three states when receiving an event: ready, busy, or preempted. In the Ready state, the agent is idle and ready to start an activity. In the Busy state, the agent is busy processing an event when the message is received so the incoming event is put into a queue and waits to be processed. The Preempted state is similar to the busy state in that the agent is doing some work when the message arrives. In this case the agent is waiting for a response from another agent, and therefore the incoming message is put into a queue and waits to be processed.

4 IMPLEMENTATION

4.1 Software

The basis for the GEMASim implementation was an open source agent framework called OpenCybele managed by

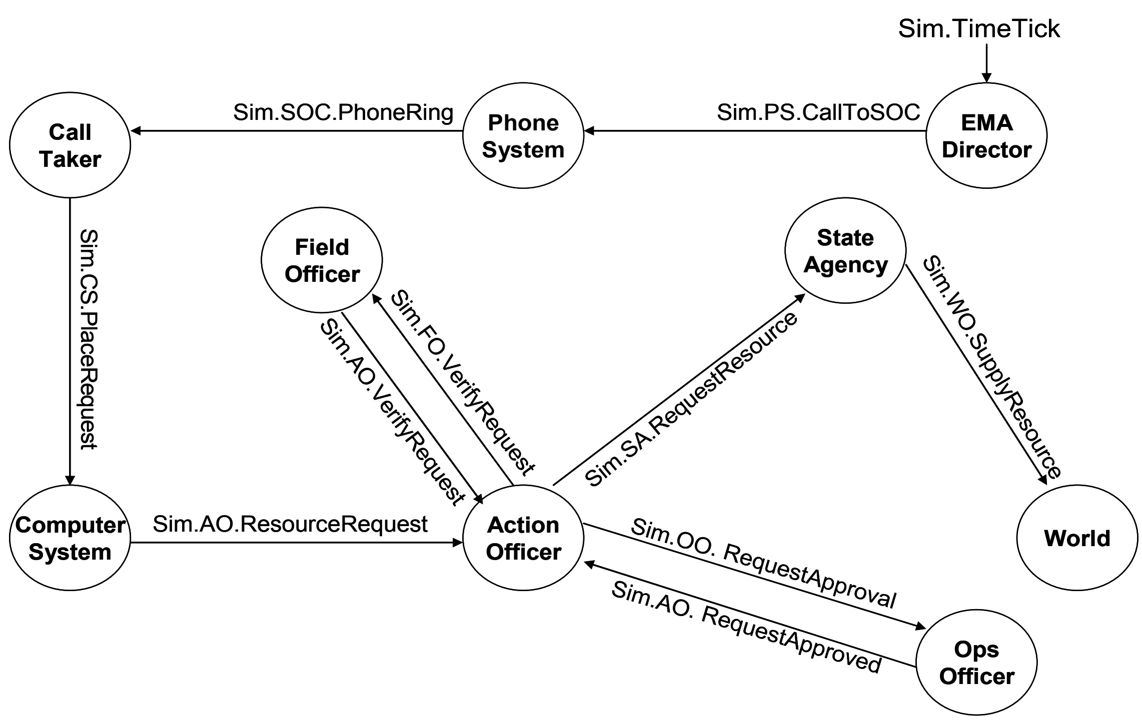

Figure 2: Activity Chart Depicting Resource Request from EMA Director

Intelligent Automation Inc (IAI) (IAI 2004). OpenCybele was created to support the paradigm of Activity Centric Programming. In this paradigm the agent is a collection of event driven activities that share data, thread and execution concurrency structure. The activities are the active element in the system. They receive events and send out new events in response. Activities have their own internal data and methods. These methods are not called directly by other activities. Each activity registers particular methods to be called when a given event occurs. This mapping of method to event can be changed dynamically.

The OpenCybele framework provided the capabilities necessary to develop an agent based simulation, an agent construct, message passing, and a time keeping mechanism. Although, OpenCybele provided many features, we did modify it in a few ways. First, we changed the communication system to be used with the Federated Developers Toolkit (FDK) (FDK 2004) instead of the communication server provided in the installation package. However, this modification was not used in the tests as only one machine was used. Also, the event management service was modified to use a time stamp to order event delivery.

4.2 Event Generator

The event generator is the agent that creates the activity in the simulation. It creates the requests that trigger all of the other activities. The Event Generator is a persistent activity. It generates a request, and decides on a time to wait. Once this time has passed it generates another request. It continues in this pattern until it receives the end of simulation message. The time between requests can be generated by three different algorithms: constant, sequence and uniform range. In the constant algorithm, the time difference is always a user requested constant. In the sequence algorithm, the time difference cycles through a user defined sequence of differences. In the uniform range the difference is chosen from a uniform random distribution range specified by the user.

The process of formulating a request is multi-stepped. First, the algorithm randomly chooses whether to submit a media, information or resource request. There is a uniform distribution between the three. If the request is a media or information request, the request is sent to the SOC. If the request is a resource request, more decisions must be made. First, the type of event for which a resource request is generated is determined. This is done by randomly selecting from the list of available event types for the particular run of the simulation. This event type is used to determine a list of appropriate resources to request. A resource is then picked based on probabilities assigned to that resource and the amount is chosen from a normal distribution for that resource. Last, it is randomly determined whether the request is sent to the local EMA or the SOC directly. The request is sent to the appropriate agent.

The probability of a resource being selected is based on the amount of the resource. The probability is simply the amount of that resource to be requested in the simulation divided by the sum of all resources to request in the simulation. The amount of resource to request is chosen from a normal distribution.

4.3 SOC Design

The SOC functions in 3 modes: normal, active, and heavy active. The first mode is normal operating procedure. In this mode, incoming calls are sent to the communication officer. The communication officer forwards the call to the appropriate person. Media requests are to the public affairs officer. Information requests are to the operations chief, and resource requests are to the operations officer. The public affairs officer and the operations chief respond immediately to requests. The operations officer will forward the request to the appropriate state agency.

In active mode, things function in a similar manner, except incoming calls are directed to the action officers. Media and information requests are forwarded to the same recipients. Resource requests are handled by the action officers. The request is assigned to a state agency. This assignment is then sent to the operations officer for approval. In heavy active mode, the flow of requests is similar, except the calls are received by the call takers. Media and press requests are still routed to the public affairs officer and operations chief. Resource requests are entered into the computer system and then forwarded to the action officer for handling. The action officer handles the request; however the information is received via the computer system instead of over the telephone.

In addition, the phone system for the SOC is modeled as a group agent. Each SOC personnel agent has one phone agent associated with it. In addition, there is an agent that acts as the entry point of all calls into the SOC, and an agent that represents the outgoing link from the SOC. The computer system is modeled by a single agent to represent the lag in the computer system.

5 EXPERIMENTS

We developed several experiments to evaluate the GEMA-Sim prototype implementation and demonstrate how the simulation could be used to assist decision-making. This section will outline the scenarios developed and the data collected during the experiments.

5.1 Scenarios

The only inputs to the model are service requests; therefore the scenario was modeled as a call volume of requests over time. We used the event generator described earlier, and varied five input parameters:

- Duration is the length of the event (in days) and any phases that may be associated with the duration. For example, an event may last 2 weeks and be divided into 2 phases, each lasting 7 days.
- Density is the rate of calls coming into the SOC for an event. The rate is measured in number of calls and can be generally classified as low, medium, and high.
- Location is the area within the state that the event is located or requests are needed. There are 8 areas defined by GEMA for Georgia.
- Resource is the type of resource requested from the SOC.
- Quantity is the amount of the resource requested.

A GEMA resource list was used as the basis for the requests. The list has the name of the resource and the name of the state agency that can provide that resource. Unfortunately, no other data was available regarding how much of the resource each state agency had or what location (area of state) the resource was located. Therefore, we created hypothetical data based on the resource list. Using the resource list, we decided how much of that resource each state agency had in supply and how many of that resource were located in each region of the state.

To drive the event generator, we then decided how much of each resource would be requested from each state agency during each phase of the scenario, as illustrated in Figure 3. The resource always came from the area closest to the event, and once those resources were depleted it would come from a neighboring area. If the resource came from a neighboring area, a time delay was added to the supply time to model transport time across the state.

Two scenarios were created using the hypothetical data. The scenarios were developed to represent two different types of events: weather-related and terrorist-related. The Hurricane scenario was modeled as a 10 day event with 3 phases and the terrorist scenario was modeled as a 26 day event with 3 phases. The scenario data is listed in Table 1 and illustrated in Figure 4 (a) and (b).

Table 1: Scenario Data

Scenario	**Location**	**Duration**	**Phases**	**Density**
Hurricane	Coast	10 days	3 days	Lo
			3 days	Hi
			4 days	Med-Lo
Terrorist	Atlanta	26 days	4 days	Hi
			10 days	Med-Hi
			12 days	Med-Lo

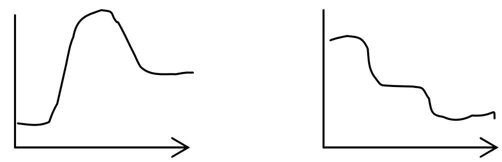

Figure 4: Call Volume Over Time for (a) Hurricane Scenario and (b) Terrorist Scenario

5.2 Experimental Results

5.2.1 Design of Experiments

Seven runs were designed which varied the experience level of SOC personnel as well as the quality of the phone and network systems. The details of the runs are shown below:

- Run 1: Skill is High for all Agents and no Phone or Network Problems
- Run 2: Skill is Medium for all Agents
- Run 3: Skill is Low for all Agents
- Run 4: Skill is Low for 3 CT and 3 AO and 50% of SA, all other Agents have High skill
- Run 5: Skill is Low for OO and OC, all other Agents have High skill
- Run 6: Skill is High for all Agents, add Phone Problems

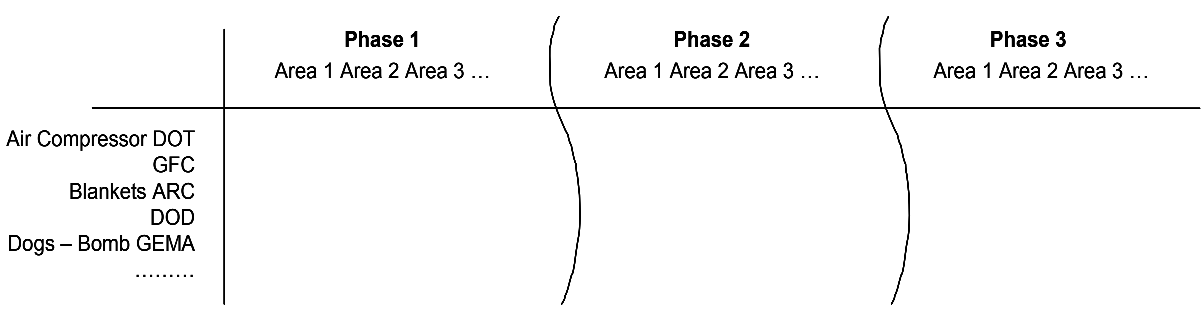

Figure 3: Resource Request Specification

- Run 7: Skill is High for all Agents, add Network Problems

In these runs, Run 1 is considered to be the baseline. While optimistic, it gives us a standard or "typical" amount of time from which to assess performance. The results of the other runs are then compared to the baseline in order to assess how varying different parameters affects the performance of the SOC.

Additionally, the hurricane and terrorist runs were also compared to each other. Since the existing SOC processes were developed based primarily on weather-related events, we wanted to evaluate how those processes work under "terrorist" conditions.

5.2.2 Metrics

Output data was collected per agent, activity, and resource. Metrics were designed for each category to help evaluate the performance of the SOC. For Agents, the metrics included time agent spent actively working on the activity; the number of times agent did the activity; and average time each instance of the activity took for the agent. For Activities, the metrics included time any agent spent actively working on the activity; average time each instance of the activity took for any agent; and the average total time plus queue time for any agent. For Resources, the metrics included amount requested; amount eventually delivered; and the time it took to deliver the resource to world.

5.2.3 Results

The data showed in this section is for the weather and terrorist scenarios. Additional results for both scenarios and all metrics can be found in (Loper 2004). The first metric is performance. Performance is measured by the amount of time required to perform all of the activities in the scenario. For example, in order to supply a resource that is requested, several activities occur including a call into the SOC, verification of need, interaction with one or more state agencies, and finally delivery of the resource to the world. The run time of the scenario is then all the activities and processes that occurred over the course of the simulation. Figures 5 and 6 show the run time for the Hurricane and Terrorist scenarios, respectively. The results are in time steps.

Run time increases for all runs as compared to the baseline (run 1). As expected, the run time increases as skill level decreases (runs 2 and 3) and as phone and network problems are introduced (runs 6 and 7). Run time also increases for the mixed skill level (runs 3 and 4), although not as severely as when the total skill level changes. The increase in run time is attributed to an increase in the time it takes agents to perform their activities, and an increase in the time activities sit in queues waiting to be processed.

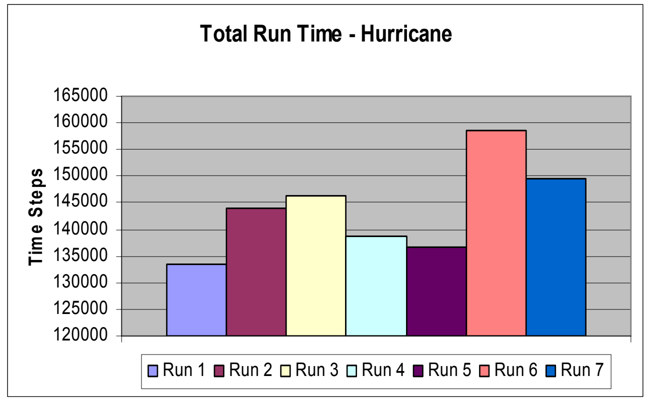

Figure 5: Run Time for Hurricane Scenario

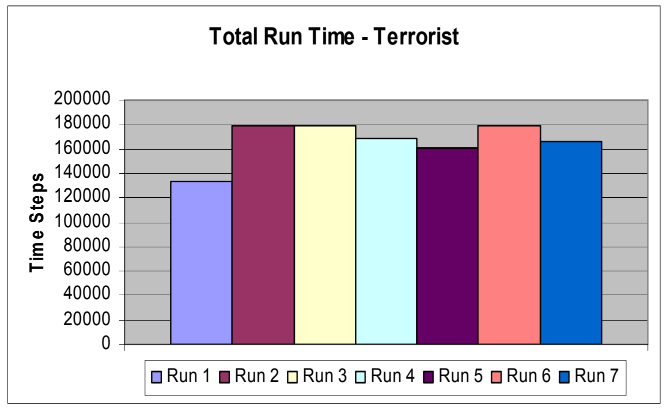

Figure 6: Run Time for Terrorist Scenario

The next metric is wait time. Wait time is the time the agent spent waiting on replies from others, as well as interruptions, to respond to other events. Figure 7 shows the total wait time (for all agents) for the Hurricane scenario.

The wait time for the Hurricane scenario have a similar shape to total run time shown in Figure 5. The wait time increases for all runs as compared to the baseline with the largest increase occurring when there are phone problems. From this graph, it would appear that phone communication is critical for the agents being able to complete their activities. When the phone system is impaired, agents wait for an additional 39 hours (over 10 days) to complete their tasks.

The next metric examines the agents that took the most time to execute their activities over the course of the simulation. Figures 8, 9, and 10 show the agent times for runs 1 (baseline), 2 (medium skill) and 7 (computer problems), respectively. They illustrate which agents are central to the SOC and the effect of varying skill level and computer performance has on run time.

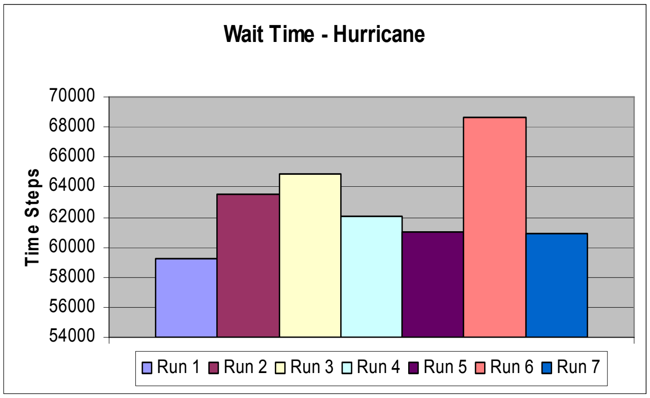

Figure 7: Wait Time for Hurricane Scenario

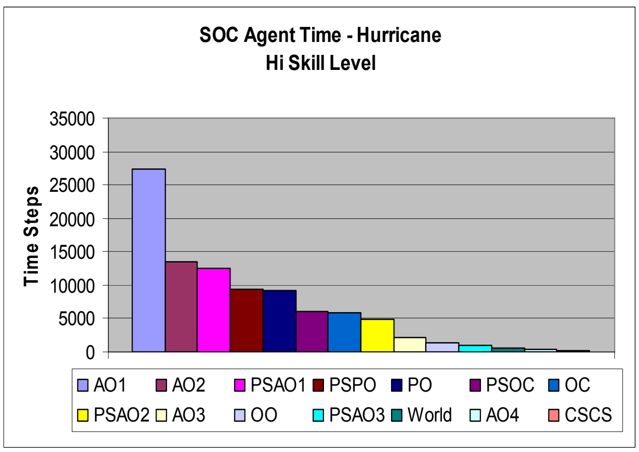

Figure 8: Agent Time for Run 1 of Hurricane Scenario

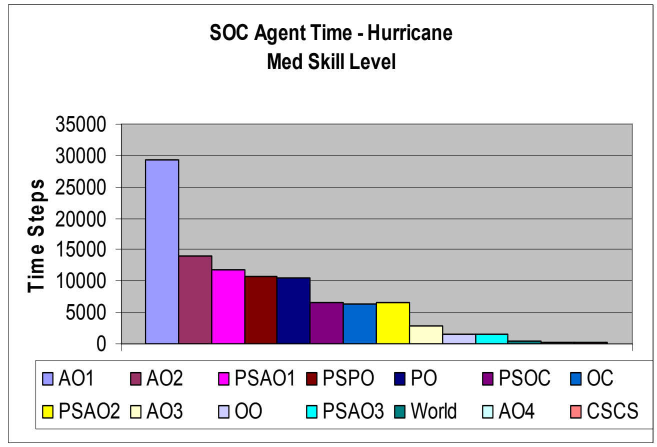

Figure 9: Agent Time for Run 2 of Hurricane Scenario

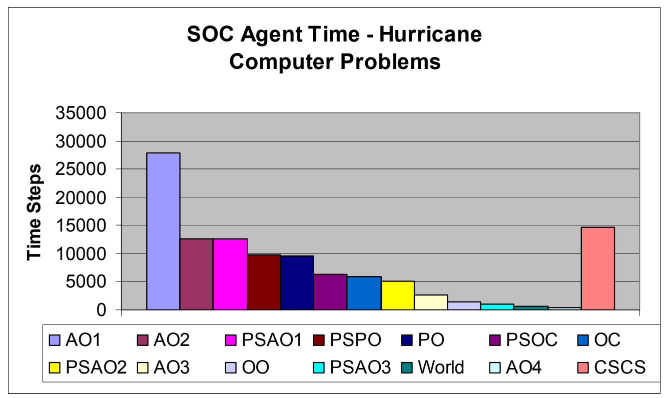

Figure 10: Agent Time for Run 7 of Hurricane Scenario

As can been seen in the figures, the shape of the curve representing which agents take the most time is similar in all cases. The most noticeable increase is in Figure 10 when the time for the computer system (CS) agent increases significantly. The information contained in these types of charts give an indication which agents are central to the execution of the SOC and which ones could become bottlenecks in the activities. This type of chart could be used as a means to evaluate process change, automation or more manpower. The charts of the Terrorist scenario look similar to the Hurricane and therefore will not be presented in this paper.

The next metric is queue time, which is the time an event waited in an agent's queue before it was processed. For example, when a request comes into the SOC, it is entered into the computer system and the AO is notified. The AO will then retrieve the request and start processing it, which includes communicating with field agents, technical experts, and internal SOC management. If the AO is busy processing another request, the new request will sit in the "in-box" until the AO has time to start work on it. Therefore, backups at queues (i.e., large queue times) represent bottlenecks in the system and can be attributed to things such as limited agent resources or process problems. In other words, adding more resources (e.g., more AOs) or changing the process an AO goes through to process the request, may lead to smaller queue times. Figures 11 and 12 show the queue time for the Hurricane and Terrorist scenarios, respectively.

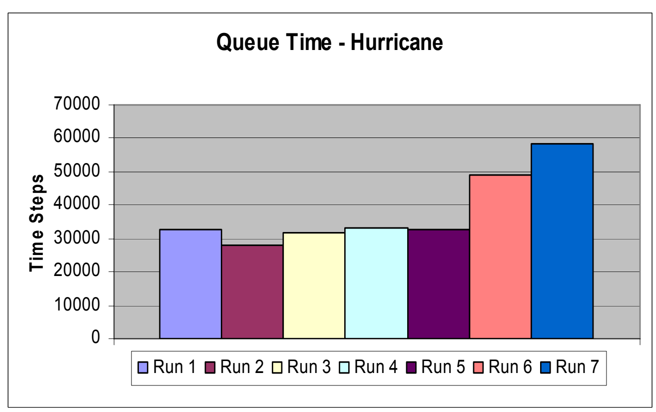

Figure 11: Queue Time for Hurricane Scenario

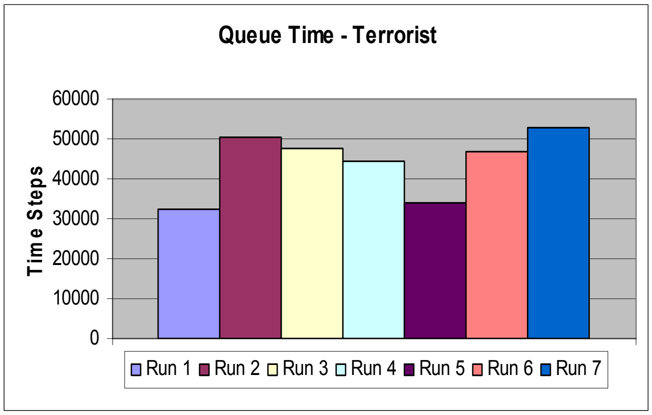

Figure 12: Queue Time for Terrorist Scenario

It is interesting to note that queue time changes very little in the Hurricane scenario as skill level is varied. However introducing phone and network problems does have an affect on how long events wait in queues. This would indicate that the processes currently used by the SOC are resilient and work well regardless of skill level. The reason queue time decreases as skill level is varied is a result of doing only one run.

The queue time data can also be graphed per run based on the agents that had the largest queue times. For example, Figures 13 and 14 show detailed queue data for runs 1 and 2 of the Terrorist scenario.

As seen in the charts, the largest queue times are attributed to the action officers and their activities. The curves have quite different shapes when skill level is decreased. The biggest increases in queue time are for AO1 to clarify requests and for the DOT and DOD to process requests. This may indicate there is a problem with how the state agencies and AOs communicate when skill level is decreased.

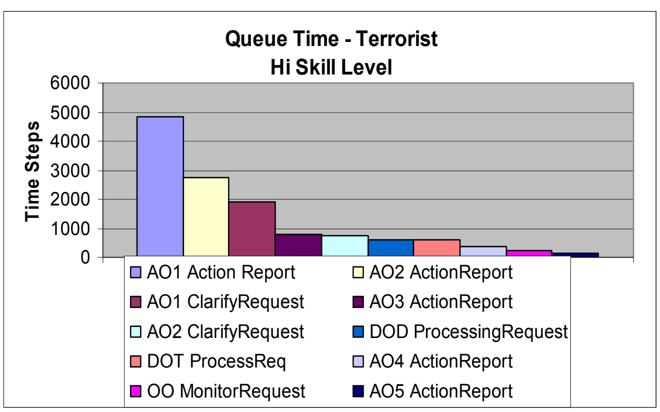

Figure 13: Queue Data for Run 1 of Terrorist Scenario

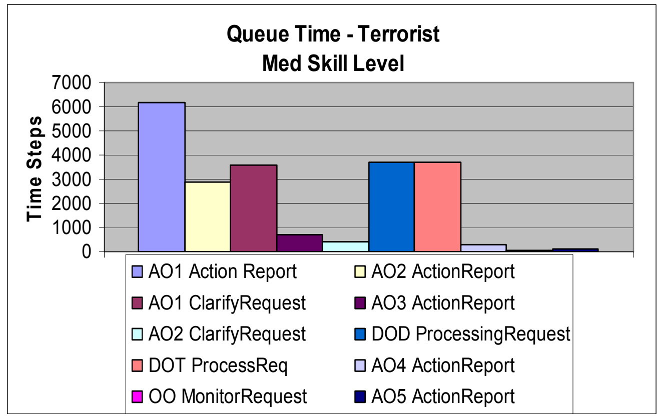

Figure 14: Queue Data for Run 2 of Terrorist Scenario

6 CONCLUSIONS AND FUTURE WORK

The data presented in the previous section are examples of metrics that GEMASim can produce for evaluating the GEMA SOC. Since the data used in the two scenarios was hypothetical and the activities implemented were our best effort at summarizing the actual process, the data should not been assumed to be valid. Instead, more work is required to collect real data and refine the processes. Further, more data is needed on actual behaviors of the agents, so more realistic representation of the SOC and its personnel are modeled.

From the data collected in the experiments, one can speculate that decreasing skill level affects performance as does introducing phone and network problems. Further, there also seems to be some noticeable change when using the processes designed for weather-related events in terrorist-related emergencies. This needs to be investigated further.

As for future work, more research is needed in the area of data collection and synthesis. There is little existing data on GEMA processes, lessons learned, and related events. In fact, most of the information we gathered came from translating "stories" into activities and scenarios.

ACKNOWLEDGMENTS

Funding for this research was provided by the Georgia Tech Research Institute Fellows Council.

REFERENCES

FDK. 2004. Georgia Tech College of Computing, Parallel and Distributed Simulation (PADS) Group, federated simulations development kit [online]. Available via http://www.cc.gatech.edu/computing/pads/fdk/ [accessed June 10, 2004].

GEMAa. 2004. Georgia Emergency Management Agency [online]. Available via http://gema.georgia.gov/ [accessed June 10, 2004].

GEMAb. 2004. Georgia Emergency Management Agency (GEMA) state operations center protocols (Draft).
GEMA. 2003 Personal communications with Dan Brown and Bobby Dockery, spring.
IAI. 2004. Intelligent Automation Inc. OpenCybele agent infrastructure [online]. Available via http://www.opencybele.org/ [accessed June 10, 2004].
Loper, M. L. 2004. Agent based simulation for homeland security, GTRI IRAD Final Report, June 2004.
NRC. 2002. *Making the nation safer: the role of science and technology in countering terrorism*, National Research Council, The National Academies Press, Washington, D.C.

AUTHOR BIOGRAPHIES

MARGARET LOPER is a Senior Research Scientist at the Georgia Tech Research Institute (GTRI) at the Georgia Institute of Technology. She earned a B.S. in Electrical Engineering from Clemson University in 1985, an M.S. in Computer Engineering from the University of Central Florida in 1991, and a Ph.D. in Computer Science from the Georgia Institute of Technology in 2002. Her current research interests include synchronization algorithms, temporal uncertainty, parallel and distributed systems, and theoretical aspects of modeling. Her research has been funded by DARPA, the Defense Modeling and Simulation Office, and the U.S. Army STRICOM. Her e-mail address is margaret.loper@gtri.gatech.edu.

BART PRESNELL is currently a software engineer at Electronic Arts. He earned a B.S. in Computer Science from Brown University in 1996 and a M.S. in Computer Science from he Georgia Institute of Technology in 2004. His current research interests are coordination in multi-agent systems, automated AI system tuning, and machine learning in electronic gaming. His email address is bart.presnell@gmail.com.