Training for emergency response with *RimSim:Response!*

Bruce D. Campbell[a], Konrad E. Schroder[a]
[a]Pacific Rim Regional Visualization and Analytics Center, Box 352650, MEB,
University of Washington, Seattle, WA, USA 98195

ABSTRACT

Since developing and promoting a Pacific Rim community emergency response simulation software platform called RimSim, the PARVAC team at the University of Washington has developed a variety of first responder agents who can participate within a response simulation. Agents implement response heuristics and communications strategies in conjunction with live players trying to develop their own heuristics and communications strategies to participate in a successful community response crisis. The effort is facilitated by shared visualization of the affected geographical extent. We present initial findings from interacting with a wide variety of mixed agent simulation sessions and make the software available for others to perform their own experiments.

Keywords: Simulation, emergency response, cognition, situation awareness

1. INTRODUCTION

Crisis management planning for emergency response teams has received increased focus in recent years as large crisis events like Katrina and 9-11 have provided benchmarks for considering wide-area readiness for both natural and man-made crisis phenomena. Counties, cities, and towns have been updating and reviewing mutual aide agreements between jurisdictions in the attempt to cement cross-boundary relationships that can be so useful in effective emergency response coordination. The creation of mutual aid documents forces individuals to get to know each other better and that may very well be the most important step in preparing for larger, regional emergency crisis events. But, mutual aid documents are quite often written with very general specifications on what the mutual aid actually entails because there is a sense of not knowing what the crisis is going to look like when it comes.

We commend this acknowledgment that the nature of crisis is chaotic and more often than not outside the realm of effective planning, especially when considering that the cost of planning for something that has very little chance of occurring can't often be justified by budgets concerning taxpayer revenue. The cost of cross-jurisdictional drills is prohibitive, and as so, physical drills are performed only for more likely or eminent scenarios. Instead, workshops and table top exercises are held for those senior personnel who are held accountable for their jurisdictions or who have a considerable economic investment that could be rendered useless by a scenario being discussed. The workshops and tabletop exercises are not inclusive of all first responders who would have to act at the time of the crisis, but instead just those key personnel who would need to communicate with each other and coordinate teams within their command.

The Pacific Rim Visualization and Analytics Center coordinated at the University of Washington, known as the PARVAC to other regional centers, has been working on developing improved shared artifacts for communicating during community-wide first response activities. We have come to realize the magnitude of the cognitive load involved in maintaining situation awareness for effective first responder behavior. We have studied successful distributed cognition methods for embedding mission-critical knowledge in shared visualization tools. We have studied the embodied mind model that suggests humans need a wide arrange of external thinking tools in order to embody an internal model for effective action. And, we have studied the emerging model for augmented cognition – a model that looks to improve social cognition between humans given solid shared artifacts and computation support with which to share situation awareness.

Through our journey to understanding emergency response, emergency operation centers, and first responders, we have seen a wonderful opportunity for developing emergency response scenario simulators with software that can be used by anybody who would participate in a cross-jurisdictional crisis management scenario. Although we couldn't justify pursuing a process whereby every emergency response role play a simulation simultaneously, we could justify pursuing a process whereby crisis management roles were embedded in software with the help of software-based agents that could perform the roles of those first responders who were not able to play their role synchronously with others. The result is our *RimSim:Response!* software that is available for download at http://www.oworld.org/parvac/rsr.html.

1.1 Motivation and Rationale

Recognition-primed decision-making suggests humans make quick decisions in emergency situations based on recognizing a situation as relevant to a situation they responded to successfully in the past [1]. When making decisions based on recognition, humans have been observed to be extremely calm and competent in performing their duties – their cognitive load is reduced so that their brain can process sensory information clearly without mental clutter. With that peace of mind, humans can process new sensory information clearly when clues in the environment suggest their initial plan is no longer appropriate. In this state, we humans compare new evidence to our current mental plan and search and correct for a new solution that can compensate for newly encountered conditions.

If the theory of recognition-primed decision-making is as valid as observed and documented so far in the literature, the theory suggests we should be pursuing a path whereby roles in society gain more exposure to potential conditions whereby recognition-based decision-making could kick in under the intense time pressures of many emergency response scenarios. Community-wide crises are inherently dangerous and costly to simulate in the physical world. We believe they don't have to be dangerous nor prohibitively costly in the virtual world of simulation software. In fact, we suggest software simulators could be extremely cost beneficial considering what is at stake during an emergency – human life, property, and a community's recovery cost associated with the long return road to normalcy.

Mental pattern recognition is not something neurologists, cognitive scientists, or computer scientists yet understand entirely, but is something that is documented in study after study of memory or learning experiments. Embodied mind theorists make a strong argument that we should not spend so much time figuring out how pattern-recognition works in the brain, but instead spend more time figuring out how to incorporate our powerful skill in pattern-recognition into the cognitive activities we need to perform in our daily life [2]. We believe that first responders, if not community residents in general, should be given comprehensive tools for representing emergency response scenarios in software artifacts they can interact with and replay in order to mentally build a history of experience they can use for recognition-based decision-making should emergency response scenarios occur in the physical world.

1.2 RSR Overview

Our investigations into emergency response training tools have continued via the development of a software-based simulation framework in which we can test out hypotheses. We call our newest software tool *RimSim:Response* (RSR) and continue to build new features to support our effective emergency and response training and planning goals based on distributed cognition, embodied mind, and situation awareness concepts. We have stumbled upon a software agent-based feature that has facilitated our understanding of emergency response and given us the opportunity to encapsulate first responder behavior in agents that can be coordinated for watching simulating first response efforts given the physical constraints of known community-wide emergency scenarios. These agents have made a huge difference in helping us develop and debug our tool while at the same time seeing new opportunities for including agents in useful simulation tools for first responders to use at their leisure – a time when their use could be less expensive to the taxpayer.

1.3 RSR Software Architecture

As described previously in [3], we have modularized the software support facilities of RimSim in order to allow parallel development of code bases needed to implement software-based emergency response simulations – along the augmented cognition model suggested by Stu Card [4]. We believe an iterative development model allows us to test out useful ideas rapidly and gain immediate feedback on the design. Our RimSim architecture, as shown in Figure 1, supported the development of a single-user medical supply logician's role training and planning simulation tool with which we integrated an optimization model into our computation services [4].

The same RimSim architecture also supports the work highlighted in this paper – the development of computer-mediated software agents with which to plan and test first response strategies. By encoding first response heuristics into software-based agents, we can use them in emergency response simulations in order to gain insights into team cognition and communication patterns while at the same time investigating possible team behaviors in response to various scenarios.

We implement computer-mediated agents by adding agent-driven communication services and agent-driven dynamic visualization interactions into our software. Our central communications server, relational database services and machine algorithm components did not need to change to support the experiments we discuss below.

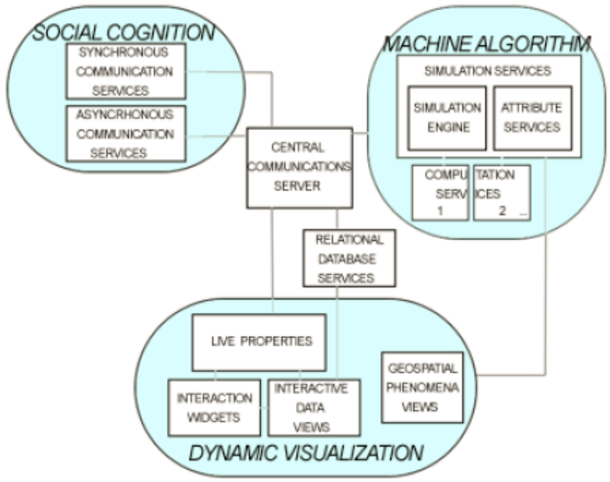

Fig. 1. RimSim software architecture diagram

The following two sections build upon our previous publications by describing a computer-mediated agents implementation that incorporates agent-driven communication services and agent-driven dynamic visualization interactions into the RimSim software to create the RSR version.

2. COMPUTER-BASED RESPONSE AGENTS

In order to simulate multiple first responders within a response scenario, we added an agent class and an agent sub-classing mechanism into the dynamic visualization services of RimSim. Software-driven agents allow emergency responders to train their pattern-recognition cognition through visualizing the actions of many first responders implementing heuristics encapsulated in agent subclass code. In order to test emergency response scenario designs without requiring help from human-based players, and to give a trainee as many repetitions as possible to gain experience through simulated scenarios, we run our visualization services often in a mode where all or most roles can be played by computer-operated agents, which play according to encoded strategies in agent sub-class types. We wish to encode realistic agent heuristics into computer-mediated agents in order to provide more useful visualizations of emergency response efforts and expect to continue to work with first response organizations to iteratively encode their procedures and policies through prolonged ethnographic study. We also invite other emergency response researchers to encode agents within our software. In the meantime, we have been able to look at a sensitivity analysis of our software agents approach with an assortment of heuristic types as described below.

Because human neurons fire much slower than computing transistors, and human players can only attend to a subset of visual interface features at one time, we chose to limit the agents to assign one emergency response resource every turn – with turn-taking time set by a simulation designer or agreed upon by the participating players before the simulation begins.

In order to fine-tune emergency response visualization through our agent-based approach, we varied the emergency response behavior of sub-classed agent types by varying three parameters that affect agent behavior: (1) their policy on sharing resources with neighboring jurisdictions, (2) their strategy for determining which active unresolved incident should be attended to next, and (3) their strategy for determining which available resource to assign the targeted incident.

Each agent takes a turn in effecting the dynamic visualization until the scheduled simulation time ends or all incidents have been handled and no more are scheduled to occur. A turn is calculated as follows:

1. An agent figures out its resource surplus (how many resources it has free, minus the number of unmet resource demands from incidents in its jurisdiction). We call this number **R** (note: R is occasionally negative).

2. We then interrogate the agent's R-level willingness to help other jurisdictions. We call this number **HiWat**. If R > HiWat, the agent looks at the requests for help that have been made by other players (human or agent). If there is an open request and the R > HiWat, the agent chooses a resource according to its Resource Selection Strategy (RSS) and send it to the incident specified in the request. *This ends a turn*.
3. If instead R <= HiWat, we interrogate the agent's R-level willingness to ask for help from other jurisdictions. We call this number **LoWat**. If R < LoWat, the agent chooses the last incident on our priority list (if there is one; priority as determined by our ISS) and broadcasts a request for help on that incident. *This ends a turn*.
4. If instead R >= LoWat, the agent chooses an incident according to its Incident Selection Strategy (ISS) subject to its Resource Sharing Policy (RSP). This RSP provides a primary ordering of the incidents according to jurisdiction, the ISS provides a secondary ordering in case of ties (and usually we find there are many ties). If there are no incidents to choose, *this ends a turn*.
5. The agent chooses a resource to send to the incident it has chosen by following its embedded heuristics, which in this case are according to an RSS. The agent sends the resource to the incident it chose in step 4. *This ends a turn.*

For the study we document in this paper, the agent behavior-defining parameters we chose to test include:

- **HiWat** - an integer representing willingness to give help when asked.
- **LoWat** - an integer representing willingness to ask for help
- **Resource Sharing Policy** (RSP) - one of 5 alternatives representing willingness to volunteer help.
- **Incident Selection Strategy** (ISS) - one of 4 alternatives representing heuristic for choosing the next incident to apply resources.
- **Resource Selection Strategy** (RSS) - one of 2 alternatives representing the heuristic for choosing the next resource to assign.

2.1 Resource-Sharing Policy

An agent's RSS policy describes under what conditions it will voluntarily respond to incidents outside its given geographic region. Five policies are defined and implemented in agent sub-classes. We refer to the policies by name in order to assist reading comprehension when describing our experiments in a later section:

- Sociopath - Never volunteer aid to another region.
- Selfish - Prefers incidents in its own region, but will volunteer aid to another region if there are no active incidents in its own region.
- Equalitarian - Does not take geographic region into account when determining which incident to handle next.
- Selfless - Prefers to volunteer for incidents in another region, but will handle incidents in its own region if there are no outside incidents to handle.
- Altruist - Never handles its own incidents, but will always volunteer for incidents outside its region.

2.2 Incident Selection Strategy

Within the broader resource-sharing policy, there is still the question of which incident to handle first, since there might be many active incidents within a single geographic region at any given time. Which incident is selected in the end depends first on policy and then on incident selection strategy. We implemented four representative incident selection strategies for our experiments:

- First Fit - Chooses the incident with the lowest incident number regardless of other considerations. Computationally, this is far the simplest of the strategies.

- Round Robin - Chooses the incident that has been active the longest.
- Lottery - Gives each incident a number of tickets equal to the total number of resources it requires, and chooses a ticket at random. The incident holding the winning ticket is selected.
- Greedy - Considers the resources that would have to be applied to each incident, and chooses the incident that could be handled most quickly (that is, on the basis of the furthest required resource).

2.3 Resource Selection Strategy

Once an incident has been identified, an agent must choose resources to assign to that incident in order to assist in incident resolution. There are likewise many possible strategies for choosing between resources to assign. For the demonstration purposes of this paper, we encoded two resource selection strategies:

- First Fit - Chooses the free resource with the lowest resource number.
- Closest - Chooses the free resources closest to the incident.

2.4 Agent Types

The agent types that we use to demonstrate our software use are a combination of the elements above, particularly the two selection strategies, since the policy can be supplied as a parameter for every agent type:

- LocalAgent: First Fit incident selection, First Fit resource selection.
- RoundRobinAgent: Round Robin incident, Closest resource.
- LotteryAgent: Lottery incident, Closest resource.
- GreedyAgent: Greedy incident, Closest resource.

2.5 Communications Model

Agents that communicate with live players or other computer-based agents do so following a communication model that can be modified to simulate communication channels that are likely to be available at the time a scenario were played out in the real world. For our demonstration, we use ideal communications channels that could be used such that every message that is intended to be sent by a player or agent is received by the intended recipient with full fidelity of the sent message. To add noise and breakdowns to communication channels would only add complexity to our demonstration that is intended to show an example of using agents to learn about an emergency response scenario across jurisdictions.

2.6 Visualization Interface

To verify that our agents worked as intended, we seeded emergency response scenarios repeatedly with four computer-mediated first response agents and watched those agents behave using our dynamic visualization interface. An example of the visualization appears in Figure 2.

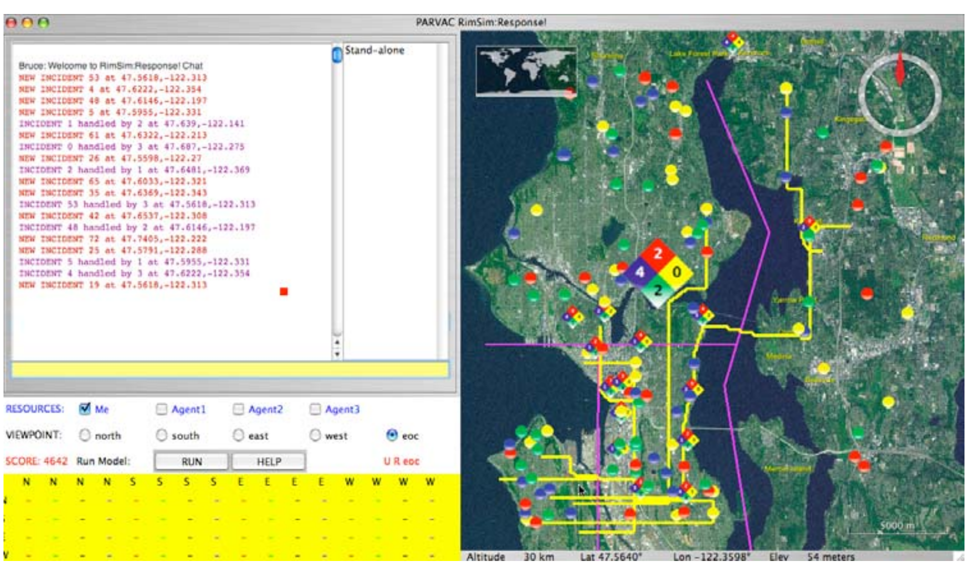

Fig. 2. RimSim visualization of agents in action

In Figure 2, messages between agents appear in the upper left corner. The player list is empty with the word *Stand-alone* appearing to confirm the visualization is running without live players (as opposed to showing player names in the live interactive player case). Resource layers can be toggled on and off to focus on resource types in isolation. Current viewpoint can be changed to the geographical midpoint of any jurisdiction participating in the scenario. A score appears in red based on the current implemented scoring algorithm (outside the scope of this paper). And, to the right, a large visualization of all incidents, resources, and paths of resources in transit to incidents appears for visually inspecting the agent run as the simulation unfolds.

We inspect current incident demand levels by hovering our pointing device (usually a mouse cursor) over the diamond-shaped icons that signify active incidents (which then enlarge for easier inspection). Numbers appear as to existing demand for resources by type. Each resource is defined by color with its matching incident demand for that colored resource visually shown on an incident. Each incident appears as an icon at the location where the incident command is managed for that incident. As a development team, we run weekly agent visualizations to inspect for the proper behavior of coded agents.

3. DEMONSTRATION OF AGENT USE

We visualize emergency response scenarios repetitively with each combination of agent behavior characteristics for each geographic jurisdiction. Given that we have five incident sharing policies, four incident selection strategies, and two resource selection strategies for each of four base agent types, we have 160 (5 x 4 x 2 x 4) possible combinations of characteristics for each of the four geographic jurisdictions.

3.1 Scenarios

To test the usefulness of visualizing computer-based agents in training for situation recognition for emergency response, we developed three scenarios of concern – all set in the Seattle, Washington, USA metropolitan area – to visualize how the different types of sub-classed agents and their encoded heuristics perform. The incident patterns were designed to represent three distinct and representative large-scale events. The following incident patterns are shown in Figure 3:

1. Bioweapon scenario: Incidents are clustered closely together, with a spatial distribution approximating a Gaussian distribution with a 1-mile half-width. Because the incidents are so closely set, two of the agents have no incidents in their regions at all. The incidents are timed to follow a slightly smoothed Poisson distribution with λ=1 in units of 15 seconds, so most of the incidents activate immediately, the rate of incident activation dropping off quickly until the last incident appears 48 seconds from the start of the simulation.
2. Earthquake scenario: Incidents are distributed uniformly over the playing field – representing a shallow source of disturbance on unstable soil – so that each region has a number of incidents proportional to its geographic size. The temporal distribution is the same as in the bioweapon case.
3. Tsunami scenario: Incidents are placed by hand along the shoreline of Puget Sound but with the same number of incidents as the earthquake scenario. The temporal distribution of incidents is the same as in the bioweapon and earthquake cases.

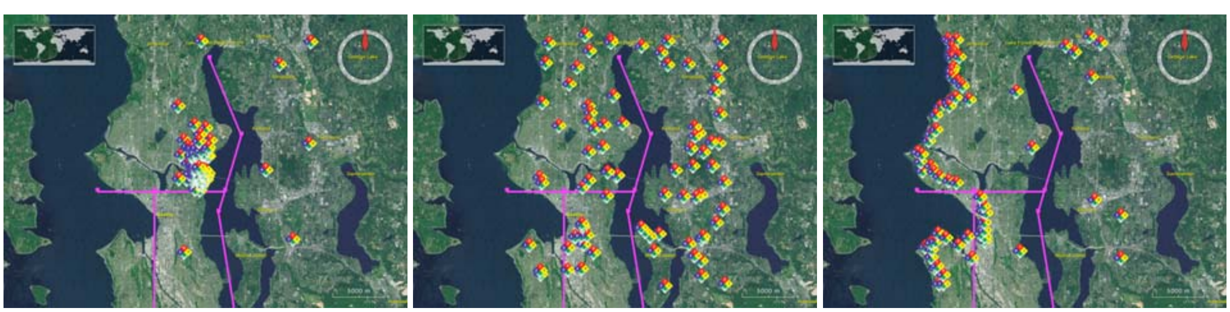

Fig 3. Incident distributions for the bioweapon, earthquake, and tsunami scenarios.

3.1 Emergency Response Resources

Within each simulation run, we randomly distribute resources under each agent's control geographically. The resources are available at the inception of the simulation continuously until the last incident is handled and no more occur in the future. So, the resources have the same probability of starting location irrespective of scenario. Each agent has forty resources (ten each of four colors), each of which can be redeployed as soon as an incident is completely handled such that no resource demand is left to satisfy.

3.2 Simulation Runs

Each of the scenarios is divided into the same four geographic areas to represent jurisdictions for the emergency response agents. We varied all possible 160 agent characteristic mixes across all four geographic regions to create 640 runs for each scenario. We watched each of the 640 simulations take place for each scenario and noticed many interesting patterns of resource allocations including obvious inefficient motions for inferior characteristic sets. The response activity across the two major bridges was especially interesting to watch as were the clustering of movements at times between jurisdictions.

4. DEMONSTRATION RESULTS

Figure 4 looks at the time to completion for each simulation across all three scenarios. Scenario completion times include the area from the top of each scenario area down to 0 on the x-axis. In all but twelve of the 640 runs, the tsunami scenario took less time to mitigate than the others (in twelve runs the completion time was identical to the earthquake scenario). The earthquake scenario took less time than the bioweapon scenario in all cases (but was much closer in the most efficient cases). The distribution of response times gives a sense of how significant agent behavior mix was to the completion time of the scenario. The more isolated the community event to one jurisdiction, the more impact agent behavior makes on completion time.

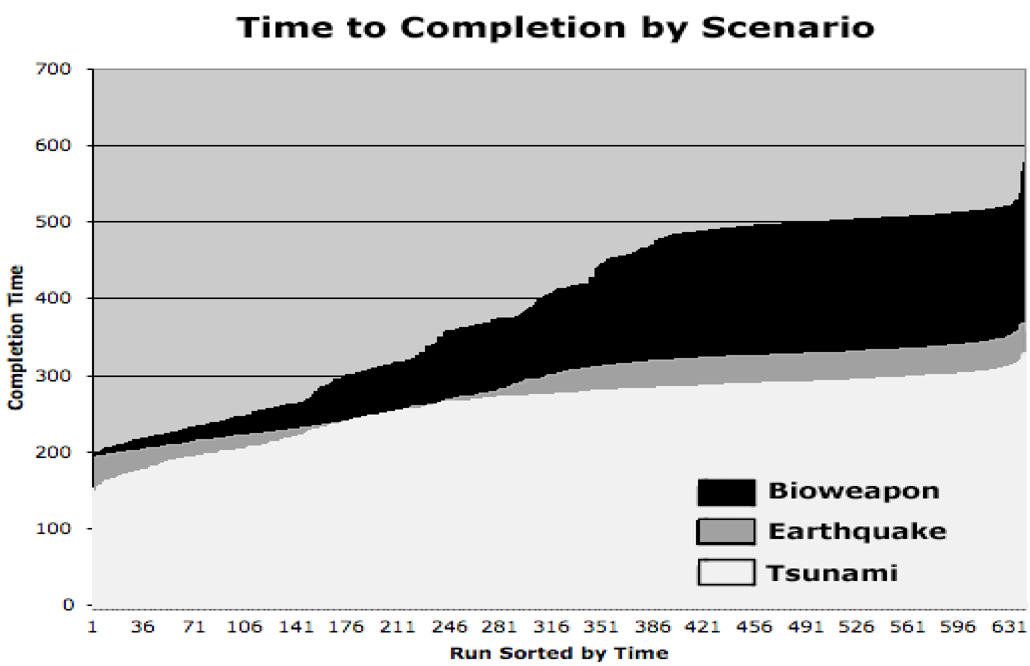

Fig 4. Seattle-based scenarios results

5. CONCLUSIONS

We believe that emergency response simulation can help train emergency responder cognition through gaining experience with important geospatial and temporal patterns associated with emergency response efforts. While performing expected first response roles through drills or a simulation may help them perform their role better, we set out to provide software with which to visualize large community emergency response scenarios so first responders could better experience the patterns of activity across all roles involved – knowing we might have to scale up to hundreds or thousands of responders eventually.

Our software provides a simulation designer the opportunity to encode agent behavior in computer-mediated agents that can then drive a visualized simulation. By visualizing the total response effort repeatedly, a first responder can get a sense of how their role interacts with others for important distributed cognition to aid their task performance – and hopefully suggest how to be a better team member in helping out with the information needs of other roles. To demonstrate the use of our software, we generated three scenarios and ran simulations of each with different mixes of agent characteristics implemented in their response behavior. The results suggest there is a useful difference in scenario completion times to consider when planning the procedures and policies each first responder should follow during emergency response.

The results shown in section 4 are provided to demonstrate the use of our software without suggesting we yet understand realistic agent behavior. We provide a simple example of how insights can be generated through the encoding and visualization of software-supported first response agents. Our paper is unable to fully demonstrate the aspects of visual pattern recognition that our simulation visualization provides. The temporal aspect of first responder movement and resource allocation cannot be experienced by a static written document. Instead, the reader can only imagine Figure 2 changing often through new transportation paths appearing, old transportation paths disappearing, resources moving toward their destination, incident demand levels changing dynamically, and incidents disappearing upon being resolved.

We believe geospatial and temporal visualization of complex emergency crisis scenarios can provide a strong communication aid for first response organizations to use in discussion with other organizations within their community. The fact that those same organizations can encode their policies and procedures into a simulation and encode the policies and procedures of organizations they want to understand better should facilitate the consideration of joint activities.

Please contact the authors to get a copy of the latest software and support in implementing simulations of interest.

ACKNOWLEDGMENTS

We thank the Pacific Rim Visualization and Analysis Center (PARVAC), the National Visualization and Analytics Center (NVAC), and the Department of Homeland Security (DHS) for providing funding and support for the *RimSim* project.

REFERENCES

[1] Klein, G.A. (1998) Sources of Power: How People Make Decisions. MIT Press: Cambridge, Massachusetts.

[2] Varela, Francisco J., Thompson, Evan T., and Rosch, Eleanor. (1992). The Embodied Mind: Cognitive Science and Human Experience. Cambridge, MA: The MIT Press. ISBN

[3] Campbell, B., Mete, O., Furness, T., Weghorst, S., Zabinsky, Z. (2008), Emergency Response Planning and Training through Interactive Simulation and Visualization with Decision Support, In *Proceedings for the 2008 IEEE International Conference on Technologies for Homeland Security*.

[4] Card, S. (2007), Augmented Cognition, *Lifetime Achievement Award Acceptance Speech*, Visualization 2007 Conference, October 29, Sacramento, CA.