

Agent-Based Simulation of Cooperative Hunting with UAVs

R. Ryan McCune, Gregory R. Madey
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46656
rmccune@nd.edu, gmadey@nd.edu

Keywords: Agent-Based Simulation, Cooperative Control, UAV Swarm, Mission Planning, Swarm Intelligence

Abstract

Swarm intelligent systems are simple but robust, capable of solving complex problems that no single agent could attempt. While technological advancements have driven development of multi-agent systems across disciplines, emergent behavior inherent to swarms is a desirable yet difficult property to exploit. Solutions utilizing swarm behavior have been proposed for the Cooperative Cleaning Problem, which is applicable to UAVs cooperatively searching for evasive targets. This work proposes a new agent behavior capable of partitioning a search area, and when combined with previous swarm solutions, forms an optimization problem of how to best assign swarms to a complex topology. Agent-based simulations are developed to test swarm solutions.

1. INTRODUCTION

Swarm intelligent systems have received much attention in recent years with technological advancement and proliferation of multi-agent systems. A swarm intelligent system is often characterized as a group of agents capable of only simple behaviors, which singularly accomplish little but collectively are capable of solving complex problems. Through interactions between agents as well as the environment, the system exhibits an emergent behavior greater than the sum of its parts. Common examples may be observed in nature, like birds adjusting flight parameters relative to neighbors to yield flock formations, or ants depositing pheromone trails that facilitate efficient discovery of optimal food routes.

1.1. Applications

The swarm paradigm is highly desired in many non-naturally occurring systems. Swarms are decentralized and extremely fault-tolerant, where the failure of any single agent is of little or no consequence to aggregate behavior. In contrast, scale-free networks, while also naturally occurring, achieve fault tolerance only through low probability of hub node random failure. But the power of swarms is founded in emergent behavior, how many simple behaviors and interactions can give rise to a self-organization capable of solving

complex problems. Problem solving strategies found in natural swarms have developed through years of evolution. The challenge for artificial systems is what behaviors will give rise to an organization capable of solving a provided problem. Agent-based simulation is the ideal tool for modeling swarms.

Governments and other public services are investigating how swarms of unmanned aerial vehicles (UAVs), or drones, may be utilized for complex problems, such as search and reconnaissance under adverse conditions. Cooperating UAVs can complete more complex missions, however at expense of complexity increase due to communication and computational demands [1]. Swarm intelligent UAVs offer an attractive solution of structures that are robust, adaptive, scalable, decentralized, and require little communication [2]. Much research has characterized frameworks for UAV swarms but few behaviors or solvable problems have been outlined.

1.2. Cooperative Cleaning

The dynamic cooperative cleaners problem characterizes the task of agents cleaning a grid containing a number of dirty tiles, with dirtiness spreading to neighboring tiles at an established rate. The cooperative hunters problem describes agents searching for intelligently evasive targets, where agents must search an area in such a manner so as to be certain where targets may no longer exist. Research in [3] demonstrates that any cleaning protocol may be used as a hunting protocol.

Two decentralized cooperative search algorithms, the Parallel Path Search [4] and the SWEEP Protocol [5], utilize swarm intelligence. Parallel Path Search, also known as the UCLA algorithm, organizes agents side-by-side into a line that moves back-and-forth parallel to the boundary region, with a certain amount of overlap to account for dirty tile expansion. In contrast, the SWEEP Protocol, or the HUNT-II algorithm, arranges agents around the perimeter of the dirty tiles, and repeatedly cleans along the perimeter as long as a newly-cleaned tile doesn't disconnect the remaining dirty region. The two methods possess relative strengths and weaknesses, namely, search with the UCLA algorithm is restricted to predetermined rectangles, while the HUNT-II algorithm can clean unknown (non-rectangular) areas, but requires knowledge of the entire grid to determine disconnect-ness.

By adding a third behavior, that of a “sentry“ UAV, search maps can be partitioned into sub-regions, where each sub-region can be cleaned by a selected number of UAVs with an algorithm of choice. Consider a stationary UAV capable of powering a stronger, larger sensor because of the power conserved by lack of movement. Placing these sentries at selected points on a map, e.g. physical bottlenecks like a tunnel or mountain pass, divides the map into more manageable sub-components, as targets may no longer move between sub-components. Partitioning the map into subcomponents forms an optimization problem of how to best partition the map based on available UAVs, search time, and costs of communication, coordination, and computation. Agent-based simulation are developed to test and experiment with different swarm cleaning algorithms, with the ultimate goal of allowing a user to assign both sentry UAVs and cooperative cleaning UAV swarms for a search map and simulate cleaning.

The paper is organized as follows: section 2 discusses related work, section 3 details the cooperative hunters problem and swarm-intelligent solutions, as well as the newly proposed sentry behavior. Section 4 examines the agent-based simulations of cooperative hunting, and section 5 closes with conclusions and further work.

2. RELATED WORK

The application of swarm intelligence and emergent behavior to computational problems is an active area of research. The Particle Swarm Optimization [6] was one of the first successful applications, utilizing agents traveling through a search space based on local results to efficiently find global optima. Ant Colony Optimization [7] also successfully applies principles of swarm intelligence found in nature to optimization problems in machine learning.

Methods to control swarms of UAV have also been investigated. Agent-based models have explored UAV control using digital pheromones [1] and emergent phenomena [8]. Multi-agent properties such as bidding are explored within the UAV context [9]. UAV behaviors for a target searching taxonomy is presented [10] as well as approaches for coverage in wireless sensor networks [11]. A framework for designing UAV mobility models for reconnaissance is addressed [12]. Similarly, distributed autonomous robot behaviors are presented for mapping.

Swarm solutions to the cooperative hunters problem are first proposed as dynamic cooperative cleaning algorithms algorithms, proposed by Wagner, Altshuler et al [5] [13] [14] [3] [15]. The original UCLA algorithm was presented in [4], the Hunt-II algorithm in [5], the two were compared in [13], bounds and analysis for the two approaches are explored in [3] and [15]. The proof that solutions to dynamic cooperative cleaning also solve cooperative hunters is given in [13]. Various lower level technicalities of the HUNT-II protocol were

outlined in [14]. Previous work in [16] explores an adaptive simulation approach [17] [18] [19] for UAV mission planning.

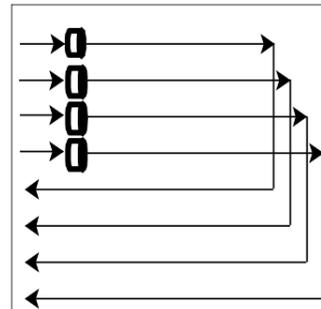
3. COOPERATIVE CLEANING

A group of UAVs may be tasked with identifying all living entities within a predetermined area with 100% confidence. Considering that some entities may be evasive or difficult to identify, the task may be considered a Cooperative Hunters problem. Similarly, the Dynamic Cooperative Cleaning problem is multiple agents tasked with cleaning a grid of dirty tiles, where dirty tiles expand at a pre-determined rate. Solutions to the Dynamic Cooperative Cleaning problem have been proven to map to Cooperative Hunters [13]. Two swarm approaches to the Cooperative Hunters problem are the Parallel Path Planning algorithm and the SWEEP Protocol.

3.1. The Parallel Path Planning Algorithm

The Parallel Path Algorithm is depicted as a homogenous group of UAVs with constant velocity and equivalent sensing capabilities [4]. The search region is a rectangle with dimensions known in advance so UAVs can properly account for dirt rate of expansion and overlap search regions if necessary when changing direction. Figure 1 from [4] depicts a flight path for vertically aligned UAVs. The shaded region in Figure 2, appearing in [15], depicts the overlap necessary for UAVs to sufficiently clean a dynamic dirty region. Refer to [4] for a formal problem definition.

Figure 1. Parallel Path line formation where agents sweep back and forth in a line [4], not unlike a group of lawnmowers



3.2. The SWEEP Protocol

The SWEEP Protocol adapts a technique from computer graphics, preserving the connectivity of the dirty region by preventing agents from cleaning critical points - tiles which would disconnect the graph if otherwise cleaned [3]. Agents are placed along the perimeter of the dirty region and move in a uniform direction synchronously along the border. Agents can detect whether a given location is dirty or not, and possess

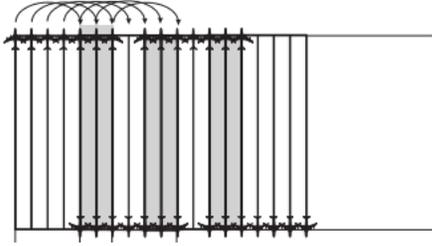


Figure 2. While agent sweep back and forth in a line, the dynamic dirty area continues to expand, so agents must overlap as shown by the shaded area [15]

a limited vision, capable of determining the dirtiness of four von Neumann neighbors. The SWEEP Protocol is versatile, requiring no previous knowledge of the topology to ensure cleanliness, permitting the cleaning of complex shapes like in Figure 3 [13].

However, determining grid connectivity for every agent on every move can become computationally expensive and requires a shared knowledge of the grid to determine connectivity. Moreover, interesting problems arise for directing a simple agent when regions become complex after rounds of cleaning and expanding. For example, rings may form which delay agent cleaning, or, as depicted in Figure 4 [3] the Swiss Cheese Trap may occur when an interior hole confuses an agent. Refer to [3] for the protocol introduction and [14] for the most detailed technical discussion.

Figure 3. The SWEEP Protocol, adapted from graphics, is capable of cleaning any connected area by placing agents along the perimeter and moving inward, cleaning as long as the tile does not disconnect the remaining area [13]

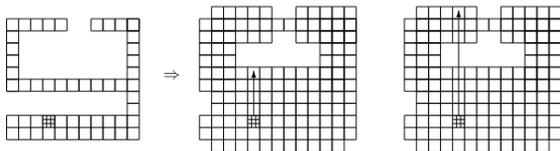
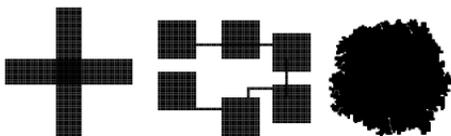


Figure 4. The SWEEP Protocol is more flexible than the Parallel Path algorithm, but is more complex because agents must know the entire map, and in some instances, make difficult and computationally expensive moves [3]

3.3. The Sentry

In addition to the two previously established cooperative cleaning algorithms, a third behavior of a “sentry” is added. A sentry is a UAV that arrives to a single provided location and sits, cleaning (or monitoring) the local tiles within the cleaning radius of a UAV. Agents in the two cleaning algorithms are only capable of cleaning the tile that the agent currently inhabits. However, due to power conserved by remaining stationary, sentry UAVs may sense and clean neighboring tiles. In current simulations, sentry agents are capable of cleaning the current tile location and all 8-neighbor tiles. The added capability of a sentry adds dramatically new capabilities for swarm cleaning. The sentry exemplifies methods to partition the search space, allowing different cleaning algorithms to be assigned to sub-regions.

For the current simulation, the sentry is centrally assigned by the mission planner, as are UAV placement and algorithm assignments. Work is under way to develop a communication and computation model for both cleaning algorithms in order to form a formal optimization problem between the algorithms. Future work will explore how sentry and algorithm assignments can be determined in a decentralized manner.

4. AGENT-BASED SIMULATION

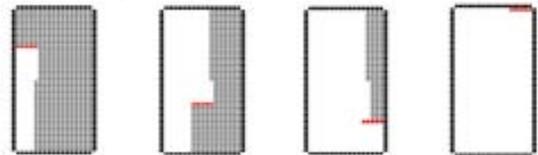
Agent-based simulations were developed to model and test solutions to the cooperative hunters problem. Simulations were built using MASON, an agent-based simulation framework from George Mason [20]. The simulations allow users to determine whether or not a swarm of UAVs can clean a given region under a given set of parameters.

4.1. Streakers

The first solution implemented was the UCLA algorithm. Topology must be rectangular, and walls are generated along the perimeter so the contaminant may not spread outside the original boundaries. The user inputs the number of agents via the console portrayed in Figure 8, as well as agent velocity, and the contaminant spread rate.

Figure 6 displays a set of vertical streakers. Agents started in the bottom left corner and moved up-and-down, left-to-right, with an overlap inputted by the user. The simulation ends after one complete sweep.

Figure 6. Vertical implementation of UCLA algorithm, agents move in parallel paths back and forth to clean the map



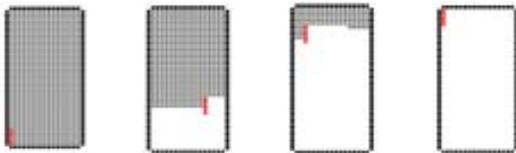
Horizontal streakers were also implemented, displayed in

Figure 5. The SWEEP Protocol cleaning a 48x48 square, where all agents begin equally spaced along the perimeter and move clockwise in unison, gradually moving inward, and adapting to the dynamic growth of the dirty tiles



Figure 7. All parameters kept equal, instances exist where one streaker implementation may clean a given topology while the other would not. The simulation may be used to test such instances. See [4] for details.

Figure 7. Horizontal Streakers can clean the same map as the Vertical Streakers in this instance, though for certain parameters and maps one orientation may clean the map while the other may not



4.2. Sweepers

The SWEEP Protocol was also implemented. For a given map, the user specifies the number of agents, agent speed, and contaminant spread rate via the console. Agents are automatically placed at equal distances around the perimeter of the contaminant. Walls are not required for Sweepers like required by Streakers.

Implementing SWEEP posed two discernible challenges. First, while agent behavior is relatively straight forward for a static contaminant, making the contaminant dynamic adds considerable complexity to agent rulesets. The Swiss cheese trap is demonstrative of such behavior. Fortunately, specific behavior and psuedo-code is provided in the relevant work.

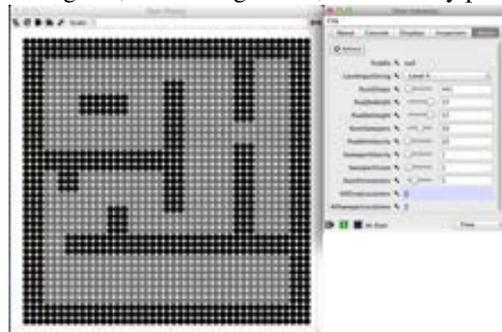
Second, the keystone of Sweeper behavior is the rule that no tile be cleaned that would disconnect the contaminant region. This requires the connectivity of the entire contaminant be tested on every step of every agent, which quickly becomes computationally expensive. One solution was to implement agent “vision“ where an agent can see in all directions from a current location (including diagonals) for a given radius. When an agent tests if a cleaning a tile would disconnect the region, the agent first tests if cleaning the tile would disconnect the region within the line-of-sight of an agent. If so, then

the entire topology is tested. Otherwise the tile is cleaned.

A simulation of Sweepers cleaning a square is depicted in Figure 5. The simulation is useful to visualize how a region is cleaned. The dynamic nature of the contaminant can produce emergent topologies and agent behaviors. In the second and third frames of Figure 8, we observe a swirl topology of the contaminant emerging from the combination of Sweeper velocity and contaminant spread rate upon the original topology. Non-random groupings of agents were also observed.

4.3. Sentry Partitioning Simulation

Figure 8. Mission Control for testing agent configurations on a new map, where the desired map can be selected, the number of agents, desired algorithm, and velocity parameters



A third simulation is under development to incorporate both Streaker and Sweeper algorithms by allowing a topology to be partitioned with Sentries. Through mouse clicks on the console, a user would select locations on the map for the sentries, and then assign a cleaning algorithm to each sub-region. This will be useful for simulating different scenarios and partition assignments according to costs associated with implementing each algorithm.

Figure 8 depicts the console with a newly instantiated map, and input console for mission parameters including number of agents, velocity, spread rate, etc.

The mission controller would review the map and select points to add Sentry agents. In Figure 9, the controller has

Figure 10. Sub-region Partitioning tests if the configuration can be cleaned, then dynamically recalculates if the map can be cleaned when an agent fails, implementing an event-driven DDDAS framework

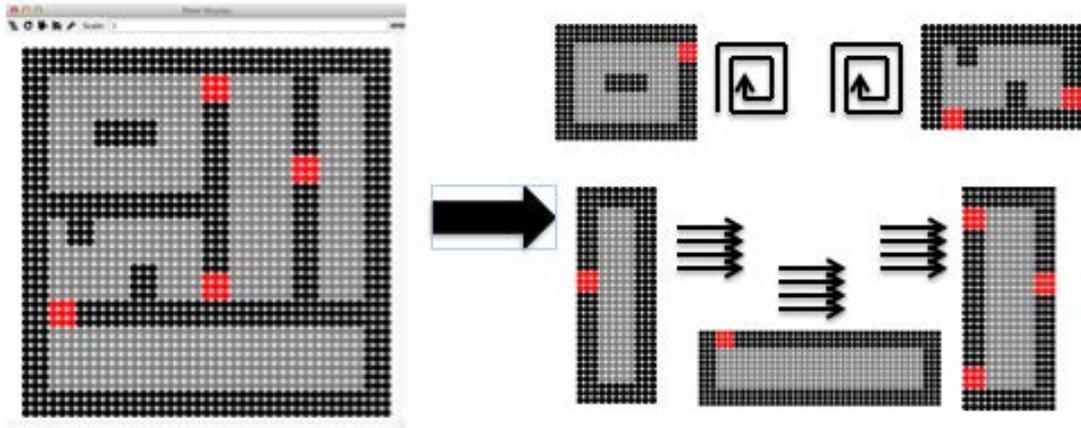
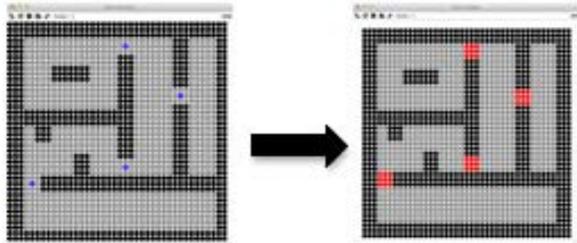


Figure 9. Assigning Sentries, through DDDAS the simulation monitors deployment and dynamically tests upon UAV failure



intelligently assigned Sentries to map locations that partition the map into less complex sub-regions. The controller is aware that in this mission, Sentries have a cleaning radius of 1, thus cleaning a 3x3 square.

Sentries can be assigned anywhere, but a given mission may have a restricted number of UAVs and the goal is to minimize total cleaning time and/or other metrics associated with the cost of cleaning, e.g. communication or computation. In Figure 9, four Sentries were assigned that partitioned the map into five sub-regions, shown in Figure 10 with their respective algorithm assignments.

The end result will allow a controller to assign sentries to locations on a map, and test different algorithms and assignments within mission context.

5. CONCLUSIONS AND FUTURE WORK

The addition of sentry UAVs facilitates spatial partitioning of a map into sub-regions, creating an optimization problem of how to best partition the map into sub-regions considering trade-offs between two swarm cleaning algorithms. The current simulation is useful to test different scenarios depending on mission contexts and determine whether or not a region can be cleaned. The simulation exposes different strategies

and trade-offs, and reveals computational considerations.

Future work includes developing a model for inter-agent communication and swarm computation. With such models, performance of a particular partitioning and assignment can be formally evaluated and compared, in addition to being tested for cleaning completeness. One observation is that Streakers can merely follow a leader that does not require much orientation, while comparatively Sweepers must share information regarding the map and avoid collisions, in addition to already complicated behavioral programming. With a communication and computational model, a formal optimization problem can be constructed.

Moreover, the role of the sentry is to act as a partition, but partitioning could be achieved through inter-swarm communication. Consider two groups of Streakers cleaning adjacent rectangles, spaces that are connected but no Sentry exists to act as a partitioner. If the two swarms were able to synchronize a pass over the region where the Sentry would normally be, then through this coordinated action the two swarms would eliminate the need for a Sentry. Agent coordination, or in this case inter-swarm communication, is a paradigm of multi-agent systems. This coordinated behavior would essentially map the behavior of groups of swarms to simply groups of agents.

Finally, the ultimate goal of swarms, including swarms of UAVs and other robots, is often to make the swarm fully autonomous. In the current form of the problem, the partitioning through assignment of Sentries is a centrally designed approach. For the swarm to be autonomous, a fully decentralized method must be explored.

Still, even if a fully distributed optimization method is discovered, the simulation proves a useful tool. Task priorities may be dynamic and a simulation allow a controller to test new configurations on the fly. Furthermore, agent-based simulations may demonstrate emergent properties of the map,

and visualizing a non-optimal approach may reveal new potential strategies.

REFERENCES

- [1] P. Gaudiano, B. Shargel, E. Bonabeau, B. Clough, Swarm intelligence: A new c2 paradigm with an application to control swarms of uavs, Tech. rep., DTIC Document (2003).
- [2] J. Rothermich, M. Ececiş, P. Gaudiano, Distributed localization and mapping with a robotic swarm, *Swarm Robotics* (2005) 58–69.
- [3] Y. Altshuler, A. Bruckstein, I. Wagner, Swarm robotics for a dynamic cleaning problem, in: *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE, 2005*, pp. 209–216.
- [4] P. Vincent, I. Rubin, A framework and analysis for cooperative search using uav swarms, in: *Proceedings of the 2004 ACM symposium on Applied computing, ACM, 2004*, pp. 79–86.
- [5] I. Wagner, A. Bruckstein, Cooperative cleaners: A study in ant-robotics.
- [6] J. Kennedy, R. Eberhart, Particle swarm optimization, in: *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Vol. 4, IEEE, 1995, pp. 1942–1948.
- [7] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *Computational Intelligence Magazine, IEEE* 1 (4) (2006) 28–39.
- [8] E. Bonabeau, C. Hunt, P. Gaudiano, Agent-based modeling for testing and designing novel decentralized command and control system paradigms, Tech. rep., DTIC Document (2003).
- [9] H. H. Brian, B. McLaughlan, M. Baker, Swarm control in unmanned aerial vehicles, in: *In Proceedings of International Conference on Artificial Intelligence (IC-AI), CSREA, Press, 2005*.
- [10] P. Gaudiano, E. Bonabeau, B. Shargel, Evolving behaviors for a swarm of unmanned air vehicles, in: *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE, IEEE, 2005*, pp. 317–324.
- [11] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, *Robotics and Automation, IEEE Transactions on* 20 (2) (2004) 243–255.
- [12] E. Kuiper, S. Nadjm-Tehrani, Mobility models for uav group reconnaissance applications, in: *Wireless and Mobile Communications, 2006. ICWMC'06. International Conference on, IEEE, 2006*, pp. 33–33.
- [13] Y. Altshuler, V. Yanovsky, I. Wagner, A. Bruckstein, The cooperative hunters-efficient cooperative search for smart targets using uav swarms, in: *Second International Conference on Informatics in Control, Automation and Robotics (ICINCO), the First International Workshop on Multi-Agent Robotic Systems (MARS), 2005*, pp. 165–170.
- [14] Y. Altshuler, V. Yanovski, Dynamic cooperative cleaners various remarks, Tech. rep., Technical report (2005).
- [15] Y. Altshuler, V. Yanovsky, I. Wagner, A. Bruckstein, Efficient cooperative search of smart targets using uav swarms, *Robotica* 26 (4) (2008) 551–557.
- [16] G. Madey, M. Blake, C. Poellabauer, H. Lu, R. McCune, Y. Wei, Applying dddas principles to command, control and mission planning for uav swarms, *Procedia Computer Science* 9 (2012) 1177–1186.
- [17] F. Darema, Dynamic data driven applications systems: A new paradigm for application simulations and measurements, *Computational Science-ICCS 2004* (2004) 662–669.
- [18] F. Darema, Dddas computational model and environments, *Journal of Algorithms & Computational Technology* 5 (4) (2011) 545–560.
- [19] C. Douglas, J. Beezley, J. Coen, D. Li, W. Li, A. Mandel, J. Mandel, G. Qin, A. Vodacek, Demonstrating the validity of a wildfire dddas, *Computational Science-ICCS 2006* (2006) 522–529.
- [20] S. Luke, C. Cioffi-Revilla, L. Panait, K. Sullivan, G. Balan, Mason: A multiagent simulation environment, *Simulation* 81 (7) (2005) 517–527.

Biography

R. Ryan McCune is a Ph.D. candidate in the Department of Computer Science and Engineering at the University of Notre Dame. His research interests include agent-based modeling, swarm intelligence, and multi-agent systems.

Gregory Madey is a Research Professor in the Department of Computer Science and Engineering at the University of Notre Dame, IN, USA. His research includes topics in bioinformatics, emergency management modeling and simulation, agent-based modeling and simulation, cyber-infrastructure, web portals for scientific collaboration, and scientific databases.