Agent Based Models for Logistics in Wargaming

Jeffrey Stavash  
Bipin Chadha  
Janet Wedgwood  
John Welsh  
Lockheed Martin Advanced Technology Laboratories  
3 Executive Campus  
Cherry Hill, NJ 08002  
jstavash@atl.lmco.com, bchadha@atl.lmco.com, jwedgwoo@atl.lmco.com, jwelsh@atl.lmco.com

Miles Parker  
Dan Teitelbaum  
NuTech Solutions, Inc.  
1100 North Glebe Road  
Arlington, VA 22201  
miles.parker@nutechsolutions.com, dan.teitelbaum@nutechsolutions.com

Keywords:  
Agent-Based Modeling, Agent-Based Logistics Simulation, Ascape, Complexity Science

ABSTRACT: Logistics plays a critical role in any war, however, until recently, its role in the wargaming arena has been limited. Lockheed Martin Advanced Technology Laboratories (ATL), in conjunction with Nutech Solutions, Inc., has developed an approach to designing and analyzing the impact of logistics on wargaming scenarios. This approach leverages concepts emerging from the complexity sciences. This paper describes our approach and its application to an expeditionary logistics scenario.

The goal of our project is to provide a robust M&S environment that can supply enough logistics detail in simulations without overwhelming the user (or the computer) with extensive inputs and scenario dependencies. A particular scenario may need to account for the existence of distributed platforms, supply levels, availability and reliability of equipment, availability of personnel, weather, terrain, enemy activity, sea/surf/road conditions, etc. Delays and losses must be considered. Logistics supply chains look more like a dense web of relationships than a narrow distribution pipeline. As the density of the web increases, there is a corresponding change in the performance of the systems that operate in these complex topologies. In complex adaptive systems, patterns of behavior emerge that are often unpredictable and have unexpected outcomes that cannot be understood by an examination of the parts. One technique that leverages these insights is agent-based modeling (ABM).

ABM is an approach to solving problems in the complexity science arena. In ABM, real-world systems are modeled as collections of autonomous decision-making entities called agents. Each agent individually assesses its situation and makes decisions based on its own set of rules. ABM methods are used to understand the behaviors that emerge from the interactions among agents and their environments making them well suited to providing future projections for logistics planning.

This paper describes an agent-based model of an expeditionary logistics scenario that models detailed logistics issues in a wargaming environment. The simulation models the effects of critical supply items on the effectiveness of expeditionary units. It also models the effects of communications, noise, and uncertainties in the environment. This paper also describes some of the results and areas for future research.
1. Introduction

The US military faces many challenges in the area of logistics. Military forces are no longer dedicated solely to deterring aggression but must respond to and support a variety of combat and humanitarian missions. From peacekeeping, to feeding starving nations, to conducting counter-drug operations, the military must continue to adapt to evolving missions and working with a broad range of allies or coalition partners [1]. Some of the worst logistics experiences in military history resulted from a chain of unexpected events that individually could have been overcome, but combined, had serious impacts on the operational readiness of the forces. Currently, logistics simulation technology is limited by its inability to represent such ad hoc, unexpected behaviors by commanders, forces and threats, and the environment.

Our goal is to directly support the logistics systems acquisition community by providing a credible, detailed logistics modeling capability that can be used as the basis for developing and evaluating new and legacy logistics systems, processes, and procedures.

The model is targeted to provide sufficient capability for component commanders, operational forces, and training commands to conduct logistics wargaming exercises and provide other types of logistics training within an operational context. Our agent-based modeling (ABM) approach provides sufficient accuracy and representation while achieving the necessary throughput to support time critical operational analysis requirements. This approach, combined with evolutionary optimization techniques, is amenable to implementation in a variety of parallel computing architectures, taking advantage of this emerging lower cost technology.

ATL has been working in the area of complex agents for two years. We have successfully developed agent-based simulations that exhibit various characteristics in the logistics domain using such agent frameworks as Ascape and Repast. We increased the speed that Ascape agents can be developed by designing a GUI development environment that created agent templates as well as enabling teaming structures for the agents. We have interfaced Ascape with a genetic algorithm tool that enables us to develop the rules required to achieve desired behaviors.

2. Agent-Based Modeling

ABM is an approach to solving problems in the complexity science area that models systems as collections of autonomous decision-making entities called agents. Each agent individually assesses its situation and makes decisions based upon its own set of rules. The rules can evolve based on time and environmental factors such as spatial relationships. The ABM technique allows for unlimited “what if” changes, deletions or additions to the rules with near real-time viewing of impact. ABM results in a realistic simulation of a system because it emulates the manner in which the world really operates. Even a simple model can exhibit complex behavior patterns and provide valuable information about the dynamics of the real world. Agents are capable of evolving, allowing unanticipated behaviors to emerge. Agent-based models often are the only realistic method of capturing and predicting non-aggregate behavior. ABM methods are used to understand the behaviors that emerge from the interactions between agents and their environments and thus are well suited for providing future projections and course of action development for dynamic environments. ABM solutions are robust and able to work even with missing or inaccurate data. Such models can be used in decision-making, and course of action support, as well as to develop flexible, adaptive strategies in dynamic environments [2].

3. Agent-Based Logistics Simulation

Agent-based logistics simulation (ABLS) is an adaptive and dynamic logistics Modeling and Simulation (M&S) capability developed by Lockheed Martin and NuTech Solutions, Inc. (formerly BiosGroup) for the Office of Naval Research (ONR) to conduct naval logistics analysis, wargaming, and training exercises in an unpredictable environment. Many different frameworks are available for developing complex agent behaviors. We built this application using NuTech’s Ascape agent framework. Ascape provides the user environment and software framework to support the development, visualization and exploration of agent-based models [3].

The objective of this effort was to develop an innovative, strategic, and operational logistics M&S tool suite, founded on agent-based modeling approaches, that satisfies a multitude of communities, including system acquisition, wargaming and training, and operations. In particular, the application enables the Logistics Commander and/or Operational Commander to create a logistics scenario or refine an existing one, run the scenario and gather and display various graphs and metrics (see Figure 3.1).

A model of a warfighting mission can be instantiated in the ABLS. The user can create ships, combat units, vehicles, equipment, air transport etc. and assemble them into different fighting forces. The types of activities that
can occur are: move, consume supplies, request supplies, request service, rearm, refuel, etc.

The simulator, which has complete VCR-like controls, runs and collects user-defined statistics that can be graphed as the simulator is running. At any time during the simulation run, parameters of the ships, combat units, air transport, etc. can be changed to run on-the-fly "what-if" scenarios. The effect of the change (if any) will be seen as the simulation continues to run. The simulator can be run in two modes: forward and backward — "Forward" to explain and understand phenomena and — "Backward" to find the parameters that lead to desired objectives [4].

By providing high-level abstractions, an agent based modeling framework allows users to define complex models of systems and their environment using relatively simple constructs. The framework provides strong user-level tools that allow direct interaction with the model environment, agents, and agent behavior. Models developed in the framework can be explored extensively at run-time and can be deployed to the web. Many features of the models can be modified without any programming, avoiding laborious edit-debug, compile-run cycles. The models are cleanly separated from visualization and control, providing design features that allow for straightforward composition of clear hierarchical models. This ensures that models are maintainable and coherent, allowing complex, dynamic interchange among agents without sacrificing repeatability and comprehension. Agents have rules that dictate their behavior within the constraints of their environment and capabilities. These rules can be selected at run-time. Parameters, such as those shown in Figure 3.1, can be changed at run-time, allowing a natural and efficient anecdotal exploration of model dynamics. Statistics collection and easily customized charts for depicting numerical data are built in.

3.1 Model

Using the ABLS environment, we developed a model to simulate a hypothetical strike by a Marine Expeditionary Unit (MEU) as depicted in Figure 3.2. Figure 3.3 contains the legend. The objective is vertical sustainment of the MEU. The scenario involves an Amphibious Readiness Group 25 miles off shore and land forces 75 miles inland. There are two enemy SAM batteries and enemy forces are present. There are 12 MV-22s and 4 CH-53s available for sustaining the forces. The following elements were created for the model:

- **Units**
  - Unit Consumption
  - Consume and reorder supplies based on size and posture
  - Stochastic supply usage
  - Units have full strength, reinforcement (reorder) and critical reinforcement (safety) levels.

- **Enemy Threat**
  - Drives Attrition, Medevac, Vehicle Damage/Loss, etc…
  - Movement
  - Waypoints

- **C3**
  - Communications
    - Communications may fail
    - Limited bandwidth (user controlled)
  - Command and Control
    - Supply chain decision making
    - Request priorities
    - Parameterized threat avoidance

- **Transport**
  - Vehicles
    - ARG ships: have 15 day supply of fuel, ammo, water
    - VTOL vehicles (MV22 & CH53’s): move supplies to units at site of their engagements
  - Order mechanism
  - Loading and unloading takes parameterized times
3.2 Scalability

One of the major areas of interest during the development of this simulation was how well it would scale. We address scalability and model flexibility through the library concept where we choose the level of fidelity for each part of the model. The scalability and flexibility strategy also periodically stresses the system to determine where the edges are and if the edges are constraining (requiring rework). It was found that the simulation has very few limiting parameters. The simulation runs quickly, even under a large graphic load (i.e. large numbers of transport vehicles in motion). The parameters that have the greatest effect on the simulation’s speed are number of enemy units, number of enemy SAMs, number of MV-22’s, number of CH-53’s, and number of friendly units. In the current implementation, enemy assets, be they enemy units or SAMs, attack in the following manner. They search the list for agents they can attack (aircraft and friendly units for enemy units, and aircraft alone for SAMs) to find those that are within attack range. Of those within range, a random draw is made and a hit occurs with some probability. As the number of possible targets in the model increases, the time necessary to determine which ones are in range should increase linearly. The time required to run the simulation should grow linearly with the number of MV-22’s, CH-53’s, friendly units, enemy units, or enemy SAMs.

A parameter sweep was run for one thousand iterations, with three parameters—number of MV-22’s, number of enemy units, and number of SAMs—swept from zero to five hundred in increments of one hundred, and with all other parameters fixed at their default values. For the purposes of the test, the number of CH-53’s was set to zero and the number of friendly units was kept fixed at its default value, but the effects of these parameters on run-time scaling should be similar to those of the number of MV-22’s.

In Figure 3.4 we show a set of plots using a color scale (giving the run time in seconds) to summarize the run-times of all the runs. In particular, the lower right plot (500 MV-22’s) shows very predictable run-time growth as a function of number of enemy SAMs and number of enemy units. In contrast, in the upper left plot (0 MV-22’s) the run-time grows with the number of enemy units but stays relatively constant when the number of enemy SAMs is varied. This is because the enemy units have potential targets (i.e. the friendly units), but the enemy SAMs have no potential targets since the number of MV-22’s is zero.
We have observed that run-time grows approximately linearly with the model parameters, indicating that the simulation scales well to large numbers of agents. What happens when multiple parameters are increased together? Since enemy units interact with friendly units and aircraft and enemy SAMs interact with friendly aircraft, run-time scales roughly as the product of the number of friendly agents times the number of enemy agents. In the worst case, where the parameters are increased equally, run time scales as the total number of agents squared. For example, when the number of agents was increased by a factor of 5 (from 100 MV-22’s, enemy units, and enemy SAMs to 500), the run time increased by a factor of 17: somewhat better than $N^2$, since $5^2 = 25$. Models with $N^2$ run time scaling are considered computationally tractable; however, if even better run time scaling is needed in the future, an implementation using quadtrees to store agents in a spatially indexed manner could be used to improve run time scaling from $N^2$ to $N \log(N)$.

### 3.3 Analysis Results

The model has almost 90 user-controlled parameters that affect the dynamics of the scenario. The model was used to analyze the following areas:

- Resupply curves – number of sorties vs. intensity
- Error in inventory assessment vs. time to run out of supplies
- Effect of Medevac on Delays in Resupply
- Optimal route planning

The following are some of the results obtained. These are some of the typical analysis a logistics commander would like to perform as part of the planning.

#### 3.3.1 Resupply Curves: Number of Sorties vs. Intensity

This experiment examined the effect of varying the proportion of total time that the MEUs are in an assault posture on the number of sorties. The expected result: the higher the intensity of the battle, the more sorties that were required, serves as a check on the basic correctness of the model.

#### 3.3.2 Effects of Inventory Error

This experiment investigated the effects of biased inventory error on logistics effectiveness as measured by total time units below safety levels. Various results can be anticipated depending upon the direction of the bias in the
error. For example, if a positive bias holds, unit commanders will think they have more supplies than they actually do, so there will be a tendency to run out unexpectedly. In the opposite direction, with negative bias, commanders will think they have fewer supplies on hand than actually is so. Consequently, they will tend to order unnecessarily, and this can put extra strains on the logistics process.

The key results of Analysis Plan #2 are depicted in Figures 3.5 and 3.6 shown below.

![Figure 3.5](image1.png)

**Figure 3.5. Effects When Perceived Inventory Levels Exceed Actual Levels – Observed Time Units Below Safety Levels**

![Figure 3.6](image2.png)

**Figure 3.6 Effects of Unbiased Inventory Perception Error – No Observed Time Units Below Safety Levels**

Under condition 1 (positive bias, large error) the results graph from the model in Figure 3.5 shows that in this run Class I inventory drops below safety levels by time step 164 and Class V drops below safety levels at time step 240. This is consistent with what we would expect to occur as a result of positive bias for inventory error. However, when the same scenario is run under condition 2 (0 bias, small error), we do not see any drops below safety levels for any class of supplies within this time period (Figure 3.6).

### 3.3.3 Effect of Medevac on Delays in Resupply

This experiment examined the effect of the trade-off between reserving vehicles for medevac and delays in resupplying. The results indicated that for every 1% increase in VTOLs reserved for medevac, the time the MEUs spend below their reorder threshold increases by more than 30 minutes. The linearity of this result was another model validating data point.

### 3.3.4 Optimal Route Planning

This investigation demonstrated the existence of a trade-off between VTOL’s taking shorter but riskier vs. longer but safer flight paths. By choosing to fly a longer, but safer route around the SAM batteries 90% of the time, and taking the riskier path 10% of the time, logistics effectiveness was maximized at near minimal cost. If the VTOLs always took the longer path or took the longer path less than 90% of the time, the MEUs spent more time below safety and reorder levels.

### 4. Conclusions and Recommendations

The team successfully built and demonstrated a proof-of-concept model for the ABLS. Four Analysis Plans were developed, studies were performed, and meaningful behavior was observed. For example, it was observed that inventory bias can have a large impact on inventory shortages and larger safety stocks are required to maintain safer routes.

In summary, ABM is a more natural and understandable model than traditional techniques. In unknown situations where there is no historical data, or the data is missing or incomplete, ABM provides valuable analysis technique.

### 5. Acknowledgements

We would like to thank Dr. David Schrady from Naval Postgraduate School and his students as well as several members of the Lockheed Martin team who participated on this project. This work was funded by an ONR grant.

### 6. References


Author Biographies

JEFFREY STAVASH is a Lead Member of the Engineering Staff at ATL. He has been the technical lead on numerous modeling and simulation (M&S) research projects, including the Smart Product Model (SPM) for the US Navy’s DD 21 program, the Collaborative Enterprise Environment (CEE) for the US Air Force, and the CollabTech program for the US Army Tank Automotive and Armaments Command (TACOM). Jeff is currently leading the Dynamic Simulation Composition IRAD which is utilizing agent-based modeling, complexity science, and bio-inspired algorithms to compose simulation behaviors for Predictive Battlespace Awareness (PBA). Mr. Stavash holds an MS degree in computer science from the New Jersey Institute of Technology and a BS degree in computer science from Seton Hall University.

DR. BIPIN CHADHA is a staff consultant specializing in enterprise systems and complex system simulation initiatives. He was a principal investigator on the ONR’s LogMS and GLC2 programs. He has also provided technical leadership on multiple programs in interoperability among information systems, supply-chain management, simulation of complex systems, agent-based modeling, and system dynamics. He has authored and co-authored several publications in the field of simulation and design of complex engineered systems. Dr. Chadha is a member of the DMSO Collaborative Environments Investigative Committee (CEIC), the ASME’s Engineering Information Management Steering Committee, and Society for Computer Simulation. He received his Ph.D. from Georgia Institute of Technology.

JANET WEDGWOOD is a Lead Member of the Engineering Staff at ATL. She is leading the Predictive Battle Management IRAD, and is consulting on the Dynamic Compositional Simulation IRAD. Ms. Wedgwood has been involved in the development of several agent-based simulations, including the ONR LogMS program. She was the project lead on the Smart Product Model, a collaborative M&S environment developed for the Navy’s DD 21 program. Ms. Wedgwood’s experience with standards groups includes participation in the Virtual Socket Interface Alliance (VSIA) System-Level Design and the On-Chip Bus Design Working Groups. The VSIA’s charter is to “dramatically improve the productivity of System-on-Chip (SoC) development by specifying open standards and specifications that facilitate the integration of software and hardware ‘virtual components’ from multiple sources.” Ms. Wedgwood introduced the taxonomy and Model Year Architecture approach developed under the DARPA Rapid Prototyping of Digital Signal Processors (RASSP). These were used as the basis for two VSIA standards. Ms. Wedgwood has an MS degree in electrical engineering from Stanford University and a BS degree in electrical engineering from Rensselaer Polytechnic Institute.

JOHN WELSH is the manager of the Enterprise Technology group at ATL, which develops composable simulation technologies, and collaborative-engineering environments for mission-level analyses, logistics supply-chain improvement, and other major system projects. These projects include development of the logistics simulation model for the ONR Expeditionary Logistics program, development of the Smart Product Model for the DD 21 program, implementation of a collaborative-engineering environment for the Air Force, implementation of an enterprise strategy for the Tri-Service program for the Rapid Prototyping of Application-Specific Signal Processors, and numerous other related programs. Mr. Welsh has an MS in systems engineering from the University of Pennsylvania and a BS in electrical engineering from Villanova University.

MILES PARKER is a Senior Software Architect at NuTech Solutions responsible for the development of ABMs and tools. He has a BS degree in political science from the University of Utah and is currently pursuing an MS in computer science at John Hopkins University. Prior to joining NuTech, Mr. Parker spent three years at the Brookings Institution where he authored Ascape, a framework to design and explore agent models. He was also a principal designer and officer at a company that developed innovative, enterprise-level, rapid-application-development tools in Java, and spent ten years developing software applications for business.

DAN TEITELBAUM is a Senior Scientist at NuTech Solutions working in the areas of technology adoption, marketing and consumer behavior, and operations and supply chain management. He has a BS degree in Electrical Engineering from the University of Illinois and a Ph.D. in Engineering & Public Policy from Carnegie Mellon University. Prior to joining NuTech, Mr. Teitelbaum co-founded Sensory Computing, a computer consulting firm specializing in multi-media and computer-based training.