Migration of the Maritime Simulation Model 2.0 into a Force-on-Force Federated Simulation Architecture

Michael Schneider, Allen S. Harvey Jr., Nicholas Livas

Engility Corporation

Lorton, VA

michael.schneider@engilitycorp.com, allen.harvey@engilitycorp.com, nicholas.livas@engilitycorp.com

ABSTRACT

The population of small maritime vessels within and around the United States greatly outnumbers the number of law enforcement vessels available to police them. Lawrence Livermore National Laboratory developed the Maritime Simulation Model (MSM) under an interagency agreement with the Department of Homeland Security (DHS) to understand concepts of operations used by law enforcement to encounter small vessel maritime traffic. MSM was implemented as an agent based model in Repast Simphony with limited behaviors and physics. DHS wanted to enhance the physics and behaviors without a significant rewrite or redevelopment of the original code. Under contract with DHS, Engility Corporation migrated MSM into the Defense Threat Reduction Agency's federated modeling and simulation architecture to improve physics, enhance artificial intelligence, and federate with other detection models. Improving the fidelity of simulations required overcoming hardware resource limitations, terrain correlation issues, and assumptions made based on how agents behave in one model framework versus another.

ABOUT THE AUTHORS

Michael Schneider is a Software Engineer at Domain X Technologies (supporting Engility Corporation). He obtained his Bachelor's degree in Computer Engineering at Virginia Tech. He has 14 years of varied experience including working with data analysis and databases, software development with C++, Java and other languages, DOD Acquisition, transportation logistics, Chemical Biological Radiological Nuclear (CBRN) and Command and Control (C2) systems, and Modeling and Simulation.

Allen Harvey is a Computational Scientist and Associate Technical Fellow at Engility Corporation. He obtained his Bachelor's degrees in Mathematics, Computer Science, and Physics at SUNY Brockport and his Masters in Computational Science at George Mason University. He is currently a PhD candidate in Computational Physics at George Mason University and a certified Project Management Professional by the Project Management Institute.

Nicholas Livas is a Modeling and Simulation Engineer at Engility Corporation. He obtained both his Bachelor's degree in Chemical Engineering and Master's degree in Systems Engineering at the University of Virginia. Mr. Livas is currently registered as a Certified Systems Engineering Professional by the International Council of Systems Engineers.

Migration of the Maritime Simulation Model 2.0 into a Force-on-Force Federated Simulation Architecture

Michael Schneider, Allen S. Harvey Jr., Nicholas Livas
Engility Corporation
Lorton, VA
michael.schneider@engilitycorp.com, allen.harvey@engilitycorp.com, nicholas.livas@engilitycorp.com

INTRODUCTION

The number of law enforcement entities is always outnumbered by the population they must police, on and off water. The Department of Homeland Security has invested in modeling and simulation to help better understand law enforcement capabilities of differing force structures and Concepts of Operations (CONOPS). In the maritime arena, Lawrence Livermore National Laboratory developed the Maritime Simulation Model (MSM) under an interagency agreement with the Department of Homeland Security (DHS) to understand CONOPS used by law enforcement to encounter small vessel maritime traffic. Although MSM and later MSM 2.0 was successful in simulating very large regions with thousands of small vessels, the approximations made for physics and behaviors introduced sufficient uncertainty in some scenarios of interest. To reduce these uncertainties, DHS asked Engility Corporation to replace the physics and artificial intelligence engines with federates from Defense Threat Reduction Agency's (DTRA) federated modeling and simulation architecture while avoiding a total code rewrite.

MSM 2.0 Background

MSM 2.0 is an agent-based simulation tool developed to evaluate the effectiveness of law enforcement detecting a threat onboard a small boat within a population of otherwise benign boat traffic. It is implemented within the Repast Simphony, an agent based JAVA modeling environment. Law enforcement systems can perform detection operations as a mission ancillary to their normal safety and regulatory duties, or in a heightened alert mode in response to intelligence about a particular threat. MSM can be used to evaluate the effectiveness of such a surveillance capability under a variety of scenarios and conditions, as well as to determine factors important for increasing system effectiveness. The software reads a map of common boat routes throughout a region of interest and patrol zones for law enforcement. During a simulation, MSM stochastically places boats along these routes and within regions while applying Monte Carlo simulation to determine average behavior for tradeoff studies and sensitivity analysis.

MSM 3.0 Background

MSM 3.0, also known as MSM Force on Force Evaluation and Analysis of Key Performance Parameters (FREAK), represents the migration of the existing MSM 2.0 code into DTRA's FREAK architecture. The DTRA FREAK architecture contains multiple modeling and simulation tools which together form a comprehensive solution set that supports training, security analysis, and investment decision support related to weapons of mass destruction. The two tools used to constitute the new MSM FREAK are Joint Semi-Automated Forces (JSAF) and Constellation of Intelligent Reasoning for Constructive Analytic Simulation (CIRCAS).

Joint Semi-Automated Forces

JSAF is a discrete event, three dimensional, agent-based force-on-force modeling and simulation engine. JSAF generates, controls, and displays joint service military, opposition forces, and civilian platforms (vehicles, people, and systems) that operate within and respond to a synthetic environment. JSAF was designed to support virtual and constructive simulations under the Defense Advanced Research Projects Agency's Synthetic Theater of War Advanced Concept Technology Demonstration program in the 1990's. JSAF was adopted primarily by DTRA and the Navy Warfare Development Center (NWDC), with both organizations maintaining their own version of the code. During the time of development, DTRA's version of JSAF was chosen to be incorporated into MSM 3.0.

Since initial development of MSM 3.0, both versions of JSAF are now merged and efforts are ongoing to incorporate the new release of JSAF into MSM 3.0.

Constellation of Intelligent Reasoning for Constructive Analytic Simulation

CIRCAS was developed as an artificial intelligence framework using Repast Simphony and is provided as an Application Programmer Interface for developers to model agent behaviors. Repast Simphony is an agent based modeling and simulation framework utilizing the JAVA programing language. Repast is and has been developed and maintained by Argonne National Labs for the past 14 years. DTRA sponsored the development of CIRCAS to replace the need for operator interactions with force-on-force federated simulation engines such as JSAF. The developer monitors reports from the federated simulation engine, updates the agent model states, and issues new orders to the agents in the simulation engine. The orders CIRCAS can issue are relatively high level; Halt, Move, Follow, Fire (as in fire a weapon), and Mount (as in mount a vehicle) are examples. CIRCAS implements the Context object required by Repast in order to perform the initial configuration, connect to the federation, and discover the federated objects. Since this object is required for CIRCAS to run it does pose a limitation to the ability for developers to build additional functionality into underlying Repast scenario such as adding different types of projections to display or metadata objects external to CIRCAS agents.

INTEGRATION APPROACH

The objective of MSM FREAK was to integrate MSM 2.0 into the DTRA FREAK architecture while reusing as much of the MSM 2.0 code as possible. Since CIRCAS and MSM 2.0 are both based on the Repast Framework it was believed that the bulk of the integration effort would be made to the way movement was handled in the models. For example, rather than agents updating their position on the Repast Cartesian grid, orders would be issued to the federate to move to a location.

The integration was broken into several phases. The initial focus was of integrating/implementing MSM 2.0 functionality into MSM FREAK prior to the inclusion of any new behaviors.

- Phase 1: Code review of MSM 2.0
- Phase 2: Set up the required scenario parameters in JSAF and CIRCAS and the initial compilation
- Phase 3: Green/Neutral force behaviors
- Phase 4: Blue force behaviors
- Phase 5: Red force behaviors
- Phase 6: New features and enhancements

Initial delivery of MSM FREAK consisted of Phases 1 through 4 while Phase 5 is currently ongoing.

Phase 1 – Code Review of MSM 2.0

Originally planned to be one of the shorter phases of the MSM FREAK integration, Phase 1turned out to be one of the longest durations. This was primarily due to obtaining access to an official release of original source code. While the code was written in JAVA and tools are available to decompile JAVA class and jar files, the code provided little insight into the original developer's intentions. When recovering source code from JAVA class and jar files, significant context is lost as developer comments do not appear and spacing/tabbing changed. In the case of the products used, line numbers were added to every line of code which had to be stripped out before recompiling. During this decompiling and review process, an official code release was requested from the developer.

With the official code obtained, the walkthrough and review accelerated by making UML designs of code. Umlet, a free and open source software UML tool was used to auto-generate UML diagrams of each class. Another tool and eclipse plugin, CodePro AnalytiX by Google**,** was used to perform a dependency analysis on the code. Both of these tools were very helpful in understanding the code at a high level, which allowed for a more targeted and organized line-by-line walk through of the code.

Phase 2 – Set Up of Scenario Files and Initial Compile of the Source Code

Setting up the JSAF and CIRCAS scenario and parameter files are fairly straight forward for anyone with experience working with those federates. One of the challenges with the DTRA FREAK architecture was that all agents that may be needed in the scenario must be declared prior to execution. This meant setting up generic scenarios for both CIRCAS and JSAF with the hundreds of agents that would be required for a typical scenario run. This is a limitation that the CIRCAS and JSAF developers are actively trying to alleviate in a future release of those tools.

Successfully compiling the MSM FREAK code required making changes to MSM 2.0's context object. Repast requires that users set up exactly one context object. Within Repast, the context object is primarily a collection of all the agents in the scenario. For MSM 2.0 the context also included data storage objects, geography objects for displaying the agents on a geo-located grid, simulation variables, and scenario terrain boundaries. CIRCAS implements its own context object that should not be modified. The MSM 2.0 context object was refactored to be a controller object which performed the same initialization functionality as before but removed the agent and geospatial projection of data. Fortunately, the JSAF federate would provide the display functionality without any further development required.

One of the key components from MSM 2.0 that was desired to be reused was the ability to process Google Earth Keyhole Markup Language (KML) files. This allowed MSM FREAK to be backward and forward compatible with MSM 2.0 scenarios. However, particular attention needed to be paid to making sure that any terrain used by JSAF for MSM FREAK scenarios was correlated with the features identified by Google Earth.

Phase 3 – Green Force Behaviors

After examining the code it was determined that green force (i.e. the benign boat traffic) behavior would be the easiest of the three forces to integrate. Green traffic follows routes specified in the scenario and always responds to simple orders from law enforcement/blue forces. Most of the changes made to green forces were to enable agents to issue move and halt orders to the simulation engine and maintain awareness of its location and progress along the route.

Phase 4 – Blue Force Behaviors

The bulk of the code changes would have to happen in the blue forces, as that's where most of the CONOPS are contained. The approach taken to integrate blue agents was to integrate one state at a time. There were 14 states in MSM 2.0. An additional state was added to support the initialization of blue agents, bringing the total to 15 states. The majority of the states and transitions are shown in the figure below. The scope of the project for the initial delivery did not include red forces. Therefore, the integration of response asset states has not yet been completed.

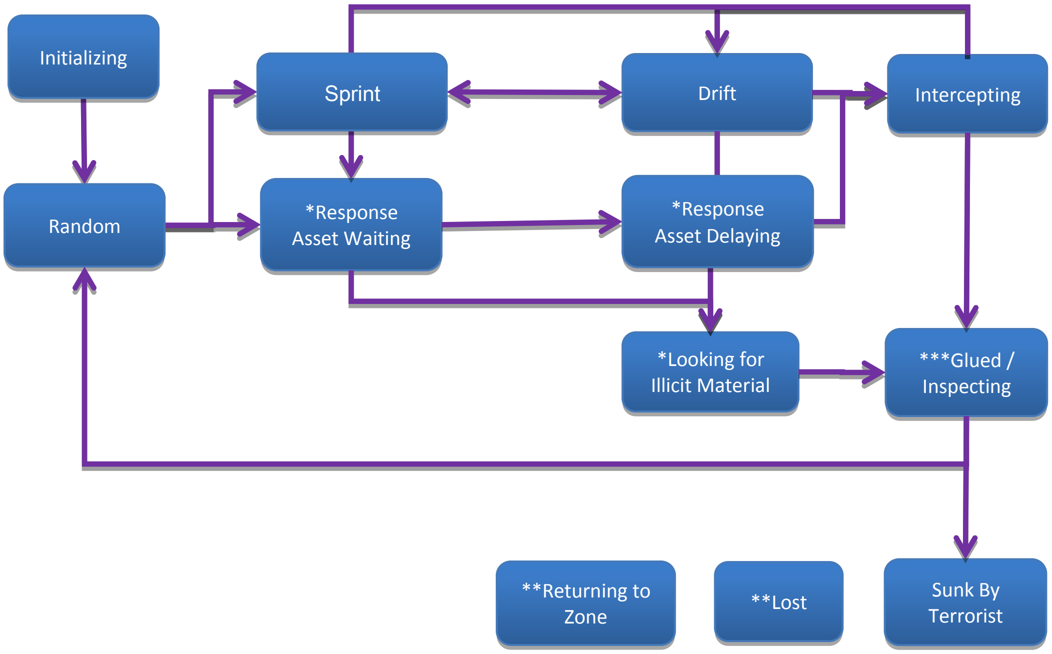


Figure 1. MSM State Transitions

Phase 5 – Red Force Behaviors

The initial delivery of MSM FREAK only included Phases 1 through 4. The work on Phase 5 has begun but has not yet been completed. Most of the red force behavior is completed as part of the green force behavior as they are nearly identical, with the difference being that red forces are trying to go undetected and therefore will behave like green forces until an encounter with law enforcement. States for red forces will need to be integrated to allow for additional red agent responses, such as attempt to flee or fight law enforcement. Blue forces will also have to be modified to include states to fight (with either a win or lose outcome) and to chase down a fleeing threat.

Phase 6 – New features and Additional Enhancements

This phase is still to be determined once total integration of MSM 2.0 is complete. However, there are some proposed enhancements to blue and green forces, particularly through expanded sets of behaviors that CIRCAS makes easy to implement. One such item includes crowded harbor scenarios where green boats are not always traveling but reach a dense and steady population with boats occasionally arriving and leaving the area. Another enhancement includes the ability for blue forces to not inspect every boat they can but rather selectively choosing boats that are perceived to be violating a law (speeding, drinking, or other unsafe behavior).

INTEGRATION CHALLENGES

There were both expected and unexpected challenges while integrating MSM 2.0 into the DTRA FREAK architecture. These challenges included modifying the synchronous behavior of the existing MSM 2.0 code to allow for asynchronous behavior in CIRCAS which enables the code to execute tasks before preceding tasks complete. The intercept algorithm, which is the logic that determines how the law enforcement boats trail and ultimately proceed towards benign boats, had to be modified in order to function properly with JSAF and CIRCAS. Finally, terrain correlation between Google Earth (the terrain used in MSM 2.0) and JSAF led to issues that displayed unexpected behavior in the code.

Synchronous vs Asynchronous MSM

MSM 2.0 was designed as a synchronous simulation. At each tick, every agent in the scenario was moved, its internal parameters updated, and states changed as necessary. This allowed for very easy and orderly state transitions and scenario progress. MSM FREAK by necessity must be an asynchronous simulation. MSM FREAK sends orders to the simulation engine and waits for responses. It must also wait for status reports from the engine regarding an agent's current location and the entities that can be seen. These events can happen at any time and in any order. Events that occurred after a specific period of time in MSM 2.0 (after a specific number of ticks) had to be changed to be scheduled to occur at a specific time. The scheduling of events and responding to external messages lead to multithreaded code. This became apparent during testing and extremely strange behaviors began to occur. An example was when two events causing a state transition occurred at similar times, such as when a law enforcement vessel's time to patrol a particular area has elapsed at approximately the same time it is trying to begin an inspection. This required the addition of semaphores to ensure blocks of code that affected state transitions could not interrupted. This did have a small impact on performance, but because MSM FREAK is limited to running real time it is negligible.

Intercept Algorithm

The ability of a law enforcement vessel to choose another boat to intercept for the purpose of further investigation had multiple issues to resolve to make an intercept realistic and feasible. In the MSM 2.0 software, boats experienced instantaneous acceleration and deceleration, or in other words they only traveled at full speed and could stop on a dime. In reality, it takes appreciable time for a stationary law enforcement vessel to speed up to intercept and it must also slow down when approaching the target vessel so that both are moving at approximately the same speed during their encounter. When determining the intercept location in MSM 2.0, a flat earth approximation can be used and law enforcement knows exactly where their target is moving to and at what rate of speed. In the federated world of MSM 3.0, law enforcement cannot cheat but must observe their target and perceive its velocity and dead reckon its position. With that information, a curved earth mathematical model was needed to determine generally how close the intercept could be. However, in determining this intercept location, the distance the target is predicted to travel (assuming it does not change its course or reach its own destination first), is dead reckoned and additional 10% to account for the time needed for the both vessels to slow down to a safe encounter speed.

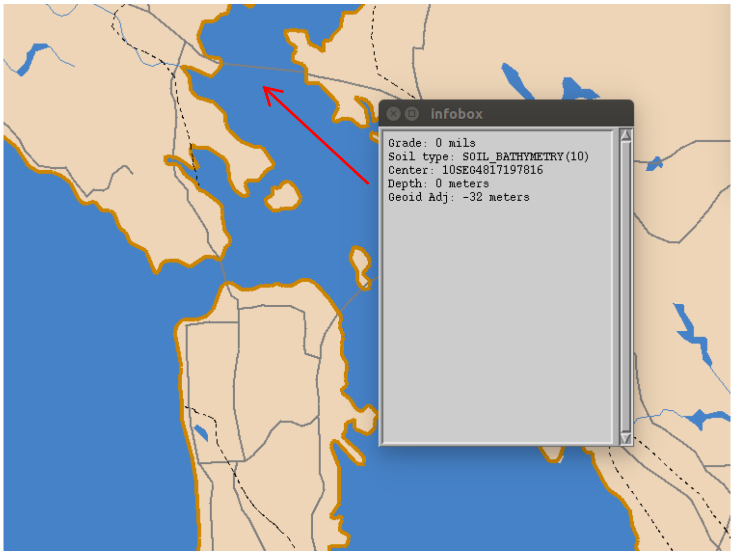
Another small issue to resolve was the frequency of scheduling intercept target locations. Whereas MSM 2.0 applies a greedy algorithm to reevaluate all potential intercept locations each simulation clock tick to always be heading to the closest target, MSM 3.0 cannot do this. Not only is this not realistic behavior, but there is processing time overhead in scheduling new move orders in a federated approach. As such, MSM 3.0 only revaluates targets every 5.0 seconds and will only issue a revised move order if the change in intercept distance varies by 10%. This does maintain the possibility that the blue vessel could pick a new target, but its likelihood is much less than in the MSM 2.0 approach.

Terrain

MSM 2.0 utilizes a KML file that defines the scenario boundary, destinations, travel routes, obstacles in the waterway, and patrol zones. In this way, the terrain in MSM 2.0 is flat and boats move wherever routes are defined or patrol regions identified, regardless if water of sufficient water depth exists. MSM FREAK uses JSAF and requires the use of a Compact Terrain Data Base (CTDB) format terrain. In addition, JSAF boats can only move on water where there is sufficient water depth. JSAF comes bundled with a low resolution world terrain and specific terrains that are more detailed for particular regions. Additional terrain files can be obtained various places; for example the United States Army Geospatial Center.

In certain areas, there were occasions when the terrain from Google Earth was not properly correlated with the CTDB terrain that was used in JSAF, which ultimately led to erratic agent behavior during the simulation. Examples of this poor correlation include times when the CTDB terrain included land (above sea level) that extended many meters into an area that Google Earth showed as navigable water. On other occasions there were areas displayed as water while being defined as having no depth (0 meters; see the figure below), negative depth (meaning depth above sea level), or being areas being displayed as water but with a soil type that was not water.

Terrain is generally built for a specific purpose and for a specific geolocated area. Like any engineering effort building to a list of requirements, terrain built for a land simulation that does not require high resolution, or just needs to "look about right" may not have implemented fully other terrain features (like navigable waterways) considered unnecessary for the purpose originally designed. These issues with terrain ultimately took significant time to debug and understand the odd behaviors seen during the scenario, as well as the locating other CTDB files that had much better defined terrain for our regions of interest.



```
Grade: 0 mils
Soil type: SOIL_BATHYMETRY(10)
Center: 10SEG4817197816
Depth: 0 meters
Geoid Adj: -32 meters
```

Figure 2. Example of Poor Terrain Correlation in JSAF's default CTDB

OBSERVATIONS

While maintaining most of the original functionality of MSM 2.0, there are significant differences in MSM FREAK compared to MSM 2.0. The scenario display, the simulation speed, and the use of a federated simulation engine are factors that make MSM FREAK different than MSM 2.0.

Scenario Display

MSM 2.0 displays the running scenario in the Repast Graphical User Interface (GUI). Due to limitations in CIRCAS, MSM FREAK does not have the capability to display the scenario in the Repast GUI. Instead, CIRCAS relies on its federated simulation engine to display the running scenario. MSM FREAK currently uses JSAF as its federated simulation engine.

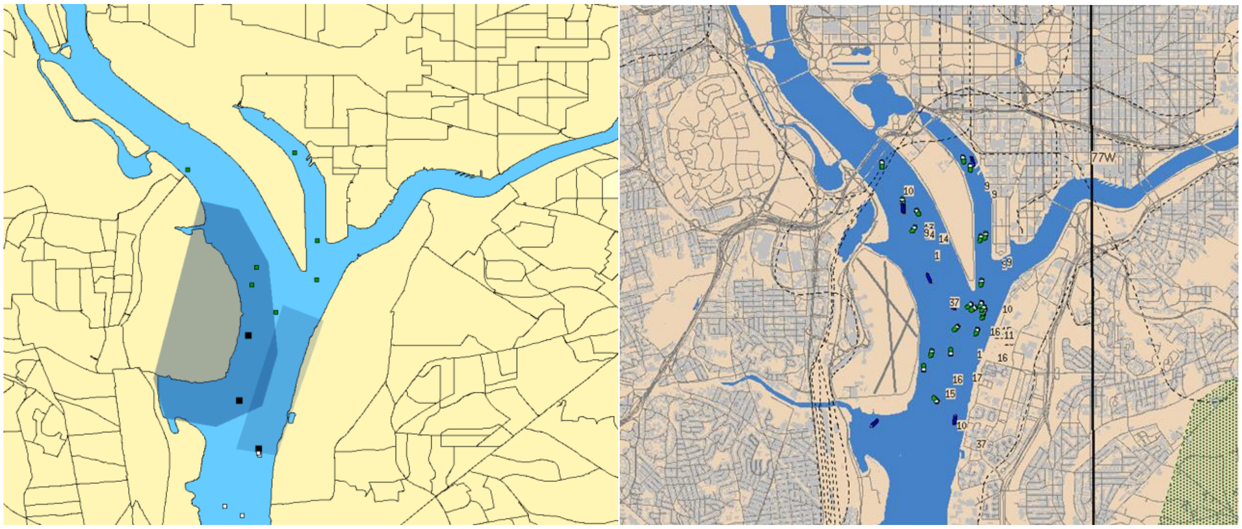


Figure 3. Comparison of Displays between MSM 2.0 (Left) and MSM FREAK (Right)

Simulation Speed

MSM 2.0 is capable of running many times greater than real time. Each simulation tick in MSM 2.0 is one millisecond of wall clock time and represents 6 seconds of simulation time, thus running at a theoretical maximum speed of 6000 times real time. MSM FREAK runs in real time. Running faster than real time in MSM FREAK is not practical except for very small scenarios. This limitation is due to the enhanced physics fidelity provided by JSAF (such as boat movements) and the resources required to run the physics and federation overhead. The simulation can be accelerated slightly if the user chooses to not display any GUI at all and only cares about metrics reported upon simulation completion.

Federated Simulation Engine

MSM FREAK requires the use of a federated simulation engine with its own terrain. Paths, zones, boundaries, obstacles, and points defined in the Google Earth KML file must correlate with the terrain being used by the simulation engine. MSM FREAK currently makes use of JSAF as the engine to run the agents being simulated. However, other federated simulation engines already within the DTRA FREAK architecture or simulation engines that have the capability to federate with the DTRA FREAK tools can be used as well.

SUMMARY

MSM 2.0 was successfully integrated into the DTRA FREAK architecture, expanding the usage and capabilities of the DTRA FREAK tools. As a result there are more realistic behaviors from boats, such as boats accelerating/decelerating and using a turning radius based on its speed. However, MSM FREAK is not a replacement for MSM 2.0 but rather a different approach with different capabilities. The choice between using MSM 2.0 or MSM FREAK is that of a standard tradeoff in the modeling and simulation arena; that of speed vs fidelity. MSM 2.0's ability to run much faster than real time it has the benefits of being able to analyze larger areas over a longer period of time than MSM FREAK. For more focused studies MSM FREAK has the advantage of being more accurate due to it enhanced physics fidelity over MSM 2.0.

Development of MSM FREAK is continuing. At the time of this report, DTRA JSAF and NWDC JSAF baselines have been merged and released by DTRA. Efforts are underway to resolve any changes introduced with this newer version of JSAF and to incorporate the remainder of red force behaviors. Future efforts include incorporating new blue and green force behaviors to allow studying a wider range of law enforcement CONOPS.

ACKNOWLEDGEMENTS

This work has been supported by the US Department of Homeland Security, Domestic Nuclear Detection Office, under competitively awarded contract/IAA HSHQDC-12-A-00004. This support does not constitute an express or implied endorsement on the part of the Government. Thanks also go out to Daniel Faissol of Lawrence Livermore National Laboratory (LLNL) (faissol1@llnl.gov) and his team as the original developers of Maritime Simulation Model. Additional thanks goes out to Argonne National Laboratory's Repast Simphony development team including (but not limited to) Michael North, Nicholson Collier, and Eric Tatara and the Defense Threat Reduction Agency's JSAF and CIRCAS development teams, including (but not limited to) Tim Leong, Dave McKeeby, Eric Granlund, David Callaway, and Laura Dunleavy.

REFERENCES

Argonne National Labs (2013). *The Repast Suite*, Retrieved December 30 2014, from http://repast.sourceforge.net/

Collier, N. & North, M. *Repast Java Getting Started*, Retrieved December 30, 2014 from http://repast.sourceforge.net/docs/RepastJavaGettingStarted.pdf

Ozik, J. (2014). *ReLogo Getting Started Guide*, Retrieved December 30 2014, from http://repast.sourceforge.net/docs/ReLogoGettingStarted.pdf

Edmunds, T. & Faissol, D. & Ni, K. & Wheeler, R (2011). *Maritime Simulation Model: User's Guide*. Lawrence Livermore National Laboratory LLNL-SM-516131.