Agent-Based Modeling for Systems of Systems

Ankur Mour, C. Robert Kenley, Navindran Davendralingam, and Daniel DeLaurentis
Purdue University
School of Aeronautics and Astronautics
Center for Integrated Systems in Aerospace
Neil Armstrong Hall of Engineering
701 W. Stadium Ave., West Lafayette, IN, 47907-2045
davendra, ddelaure, kenley, and amour at purdue.edu

Abstract. Agent-based modeling is an important tool for the engineering of systems of systems. This paper briefly reviews the historical development of agent-based modeling and system of systems concepts, compares agent-based modeling to other approaches, and describes the Purdue Discrete Agent Framework for agent-based modeling. An application of the Discrete Agent Framework to a system of systems is described and illustrates the ability of agent-based modeling to capture non-intuitive behaviors that may arise due to the complex dynamics that occur in interconnected systems of agents that follow a behavioral set of rules. The paper concludes by looking back to the past to understand the potential for applying agent-based modeling to support the ongoing engineering and operations of an evolving system of systems.

Introduction

Agent-based modeling employs a collection of autonomous decision-making entities called agents imbued with rules of behavior that direct their interaction with each other and their environment. Agent functionality is quite flexible, with behavior types ranging from simply reactive (change state or take action based on fixed rules) to learning and adaptive (change state or take action after updating internal logic schema via learning).

Agent-based modeling is an important tool for the engineering of systems of systems. Using this approach, systems engineers can investigate alternative architectures and gain an understanding of the impact of the behaviors of individual systems on emergent behaviors.

The purpose of this paper is to provide an understanding of the concept of systems of systems and the application of agent-based modeling to systems of systems. We review important literature on systems of systems and agent-based modeling, and we compare agent-based modeling to other methods for analyzing systems of systems. We describe a simulation tool that provides a capability for agent-based modeling and simulation and describe the application of this tool to a system of systems. Finally, we describe a successful application of the concepts of agent-based modeling to a system of systems that has been in operation for multiple decades.

Brief Review of the Literature on Systems of Systems and Agent-Based Modeling

A working paper from the Santa Fe Institute by Marimon, McGrattan, and Sargent (1989) used an agent-based simulation to analyze money as a medium of exchange in an economic system. This paper may be the earliest publication that describes simulating agents rather than performing closed-form mathematical analysis of a system of agents.

About the same time, Eisner, Marciniak, and McMillan (1991) used the term “system of systems” in a paper that described the need for a new discipline that could provide overall management control over independently acquired systems that have different time phasing and interdependent coupling and that tend to be uni-functional individually while the systems of systems is multi-functional. In addition, optimizing the individual systems does not guarantee optimization of the overall systems of systems and the combined operation of the constituent systems is necessary to achieve an overall mission objective. They described specialized computer-based tools for developing a concept of operations, managing an integrated acquisition master schedule, analyzing network performance, prototyping user interfaces, tracking configuration baselines for constituent systems, and interface compatibility modeling and analysis. They developed their approach to manage the acquisition of a threat warning and attack assessment system of systems, and they foresaw the need to develop system-of-systems methods applicable to air transportation, digital communications, and strategic missile defense. They refer to “the extensive use of automated models and simulations in order to predict performance” (532).

Eisner, McMillan, Marciniak, and Pragluski (1992) described the Rapid Computer-Aided System of Systems Engineering (RCASSE) environment that they structured using ten elements: mission engineering, baseline architecting, performance assessment, specialty engineering, interface compatibility evaluation, software evaluation, risk management, scheduling, pre-planned product improvement, and life cycle cost evaluation. They point out that “The advance of computer networks has created a complexity that certainly did not exist with individual, non-automatically interfaced systems” (268), and they indicate that under the RCASSE process results of performance assessment from automated modeling and simulation tools feed back to the preceding step to modify, as necessary, the baseline architecture.

Maier (1996) described five principal characteristics of systems of systems that distinguish them from monolithic systems: operational independence, managerial independence, evolutionary development, emergent behavior, and geographic distribution. Maier (1998) identified the first two of the five characteristics as the necessary and sufficient properties for a collections of systems to be regarded as a systems of system. He also concluded that for systems of systems, “it is apparent that the architecture of each is defined through communications” (280) and that “Collaborative and virtual systems-of-systems will also become more common with the ubiquity of smart systems independently operated and managed” (283).

Clymer (1997) defines an agent-based system architecture as a set of agents, the activities or functions each agent performs, the interactions of each agent with the environment, and how each agent communicates with other agents. When we refer to the term “architecture” in this paper, we are using Clymer’s definition. Clymer’s methodology was intended to be applied by systems engineers to realize complex adaptive systems, which “are comprised of a collection of agents where each agent in the system communicates data, knowledge (rules and facts), and mission goals with one or more other agents” (1).

DeLaurentis (2005) expanded Maier’s five characteristics of a system of systems to include networks, heterogeneity, and trans-domain. By trans-domain, DeLaurentis means that effective study of systems of systems requires unifying knowledge across several fields of study: engineering, economics, policy, and operations. He evaluated the four system views of Rouse (2003) (hierarchical mapping, state equations, nonlinear mechanism, and autonomous agents) for applicability to systems of systems and concluded that the modeling of a system of systems as autonomous agents is well suited for capturing the emergent behavior that derives from complex interactions of the other six characteristics of systems of systems. He also described how object-oriented methods could be effective and efficient for implementing agent-based models. Hsu and Butterfield (2007) defined four principles of emergence and proposed that agent-based modeling to measure the existence, type, of level of emergent behavior of systems of systems and the initiation mechanisms for the emergent behavior.

The recent increase in availability of software packages for agent-based simulation and the increased understanding that agent-based modeling is well suited for systems of systems has resulted in many applications of agent-based modeling to systems of systems. Kilicay-Ergin and Dagli (2008) describe the use of AnyLogic agent-based simulation software to model the behavior of alternative system-of-system architectures for financial markets. Hsu, Price, Clymer, Garcia, and Gonzalez (2009) describe using OpEMCSS software to simulate the behavior of a system of systems that they call the World Model. Giachetti, Marcelli, Cifuentes, and Rojas describe a simulation that uses Java-based CybelePro software to model the performance of a human-robot team as an agent-based system of systems.

Comparison of Agent Based Modeling to Other Approaches

Various methods and tools for simulation of complex processes exist; however, they primarily fall into the main categories of equation-based system dynamics, discrete event simulation, and agent-based modeling. The following section provides comparative discussion on the differences between agent-based models and the other methods.

System dynamics is defined as the "the study of information-feedback characteristics of industrial activity to show how organizational structure, amplification (in policies), and time delays (in decisions and actions) interact to influence the success of the enterprise" (Forrester 1958, 38). The system dynamics methodology uses mathematical equations to represent feedback loops of stock flows throughout a process network. An important assumption for system dynamics is that individual stock items are indistinguishable and that the resulting feedback loop generated is an adequate representation of each individual stock flow's aggregate behavior.

Typically, system dynamics represents an aggregate level of performance as continuous differential equations. Analysts employ these models support strategic-level decision-making and to develop an overarching view of long-term trends in the dynamics of an enterprise. System dynamics has been widely used across a range of applications that range from socio-economic to engineering systems, and aims to reduce complex behaviors to their most aggregate forms assuming that adequate, structured representations of the behaviors exist.

Agent-based modeling is a natural choice for analyzing systems of systems because (1) it can capture emergent phenomena, (2) it provides a natural description of the system, and (3) it is flexible (Bonabeau 2002). Schieritz and Milling (2003) and Borshchev and Filippov (2004) provide a comprehensive comparison of simulations that use systems dynamics versus simulations that use agent-based modeling. System dynamics focuses on a top-down, aggregate modeling that typically uses continuous-form representations (feedback loops) of system processes; agent-based models are based on discrete agent-specific logic rules that take a bottom-up approach to simulation. Agent-based modeling provides a means of connecting micro-level behaviors to the macro level of a system whereas systems dynamics link system structures to system behavior (Schieritz and Milling 2003, Borshchev and Filippov 2004). The main difference is in the ability of agent-based models to capture emergent behaviors. The agent-based setting allows flexible interactions between individual agents, which results in non-intuitive dynamic modes being generated. System dynamics reduces the possibility of exploring emergent phenomena due to the natural filtering of these modes that occurs through enforcement of aggregate equations over populations of individuals within the system, and the establishment of a rigid flow structure.

Discrete Event Simulation is a method used to model real world processes as a series of interconnected discrete events that are functional processes. These processes are typically at the low- to mid-level state of abstraction in the hierarchy of interconnected systems and do not consider performance characteristics of the individual elements that execute these processes themselves. The focus of a process-centric simulation here is naturally well suited to applications where processes are the critical aspect of analysis such as in healthcare (e.g., patient flow), manufacturing (e.g., production floor processes layout), and logistics (e.g., distribution processes at a hub).

As with the system dynamics approach, the discrete event simulation approach is a top-down approach that models aggregate behaviors of processes.

A more recent method of discrete event simulation uses Petri nets, a mathematical and graphical method that employs triggers, tokens, and transitions to model interactions between entities (Murata 1989). Further developments of Petri nets have resulted in colored Petri nets where colored tokens represent the flow of different packets of information along the feasible pathways of the Petri network. Much literature has proliferated on Petri nets due to their simplicity and computational ease of use. Additionally, they bear some useful properties in comparison to other methods. For example, when contrasted to Markov chains, Petri nets do not require an increase in the number of states (and consequently state variables) with an increase in the number of tokens used in the model. This preserves the computational complexity of the underlying simulation and scales well to larger problems. In addition, Petri nets easily handle serial and concurrent execution of process events.

While discrete methods use a powerful and intuitive representation of processes in a system, they are mainly intended to model and represent finite interactions where the underlying structure of the process is already known. They share the focus with systems dynamics of modeling top-down characteristics of a system and assume pre-defined structures and aggregations of macro behaviors. In contrast, agent-based models are able to more generally represent individual entities that drive the discrete events and allow for possible emergent behaviors that are not otherwise apparent from the aggregated discrete dynamics of a system.

DAF Approach to Agent-Based Modeling

Purdue University developed the Discrete Agent Framework (DAF) for agent-based modeling in 2010 to enable easy application in multiple domains. Developed in object-oriented MATLAB, this engine provides the foundation to build agent-based simulation models to explore various architecture configurations for systems of systems and evaluate their performance. DAF also enables coordinated development, verification, and validation of the system of systems architecture through selective failure simulation.

DAF allows the modeling effort to focus on the systems of systems itself and not the logistical “dirty work” of getting a runnable simulation from a blank slate.

The first major application of DAF was through a sponsored research initiative of the Missile Defense Agency of the US Department of Defense. The purpose of this research was to examine and model a Ballistic Missile Defense System (BMDS) as a system of systems by simulating it as a collection of functions (executed by agents) that could be distributed in a myriad of ways. For example, a ground-up development effort to model a BMDS would require coding not only agent behavior, but also routing of communication between agents. Development in DAF reduces the routing into a single command, allowing the modeler to focus on agent behavior. The engine provides the means and a head start to addressing top-level objectives, developing mathematical models and agent behavior algorithms, and defining the architectural design space.

DAF views a system of systems architecture as a collection of agents that are connected by communication links. In practice, each agent in DAF is an in-code application of a formal model developed from research and of communication links that emulate real or proposed communication standards (Chow, Braun, and Fry 2012). This approach allows a DAF user to follow Maier’s communication-centric architecting approach (Maier 1998) by using the different possibilities of linking these agents as a means to distinguish one architecture from another. DAF can used to generate and evaluate a wide variety of architectures by defining the functional capabilities and behaviors of agents and the communication links between the agents. A representative implementation of DAF involves generating architecture alternatives, then simulating them to identify the configuration that provided the best balance of efficiency and reliability.

There are certainly many *agent-based modeling packages* available and each provides many combinations of capability, ease-of-use, and availability. For example, SWARM is an open source Objective C/Java-driven simulation system for modeling complex systems through discrete event simulation. NetLogo is another Java-based package that provides multi-platform complex system simulation, but also comes with a large database of sample models and implementations. Initial development work on the SWARM system indicates that an object-oriented development environment is ideal for building agent-based simulations. The primary differentiator between DAF and other packages is that DAF is MATLAB-based, and as a result, any DAF application can utilize the many mathematical, statistical, and visualization tools built into MATLAB or the many supported and third-party toolboxes associated with MATLAB. Additionally, the widespread use of MATLAB in research and industry should reduce the time required to learn how to use DAF and the time to apply it to a particular project.

Results of Applying DAF to A Littoral Operations Scenario

In order to illustrate the capabilities of agent-based modeling using DAF, we developed a simulation model to capture the performance of a Littoral Combat Ship (LCS) squadron in a scenario involving multiple threats. The LCS is a frigate-sized, modular platform optimized for operating in the littorals, or coastal areas. The LCS is distinctive for its modular "plug-and-fight" mission packages. Rather than being a standalone, multi-mission ship, the ship's mission orientation can be changed by changing its mission packages. (O'Rourke 2012). The LCS without any mission packages is referred to as the LCS seaframe. The seaframe forms the core of the LCS and provides basic self-defense capability through sensors, weapons, and speed while the mission packages form the bulk of the war fighting capability of LCS. The seaframe is augmented by mission packages that are focused in one of three mission areas: Surface Warfare (SUW), Anti-Submarine Warfare (ASW), or Mine Counter-Measures (MCM).

The SUW mission package adds a MH-60R helicopter armed with Hellfire missiles, the Non-Line of Sight (NLOS) missile system, and a Vertical Take-Off Unmanned Aerial Vehicle (UAV). The SUW mission package provides maritime security and prosecution of small boat threats in littorals. The ASW mission package uses off-board technology to detect, classify, localize, and prosecute threat submarines. The package includes Unmanned Surface Vehicles (USVs), Remote Manned Vehicles (RMVs), and the MH-60R helicopter that both employ a dipping sonar for the detection of sub-surface targets. The ASW LCS does not have an anti-submarine weapon and is dependent on its MH-60R helicopter to deliver anti-submarine weapons. Finally, the MCM mission package is dependent on its helicopter for neutralization of detected mines. The USVs and Remote Mine-hunting Systems (RMS) in the MCM mission package use towed bodies to detect mines.

Figure 1 shows a hierarchical view of the system of systems for the littoral operations scenario using a lexicon developed by DeLaurentis and Callaway (2004). The collection of entities at the lowest level of the lexicon (indicated by α in the figure) and their connectivity determines the construct of a β-level collection of LCS Mission Packages, Surface Threats, Sub-Surface Threats, and Merchants. The collection for a littoral region (γ-level) is a collection of β-level entities, and the δ-level is a set of National and International Institutions to which the γ entities belong. DeLaurentis and Callaway (2004) also define four categories (resources, operations, economics, and policy) that are used in Table 1 to describe the breadth of the SoS problem as well as to help guide modeling and simulation of the SoS.

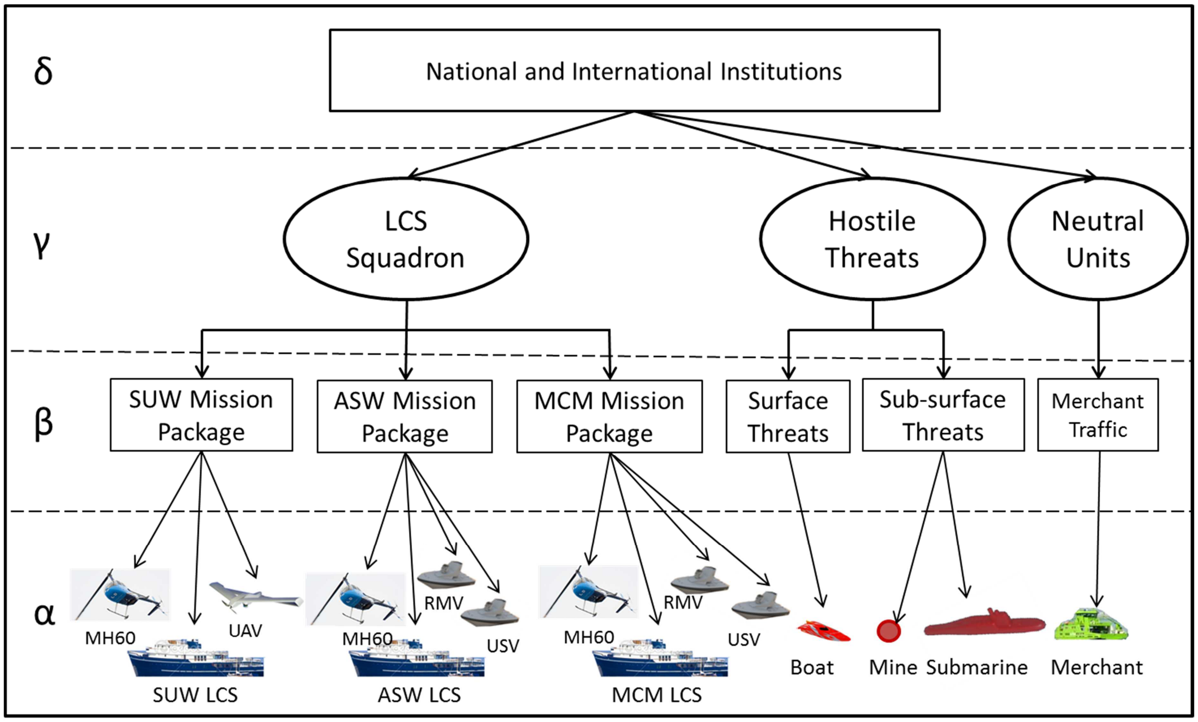

Figure 1. The SoS Hierarchy for Littoral Operations

Table 1. Description of SoS Categories for Littoral Operations

Level	Resources	Operations	Economics	Policy
α	LCS, MH60, UAV, USV, RMV, Boat, Mine, Submarine, Merchant Ship	Operating a single resource like LCS, Submarine, etc.	Economics of acquisition and operation of single entity during a patrol or voyage	Policies regarding single-resource usage: crew make-up, roles, and responsibilities; weapon loadout and firing doctrine, etc.
β	SUW / ASW / MCM Mission Package, Surface Threats, Sub-surface Threats, Merchant Traffic	Mission package operations, threat operations, merchant ship operations	Economics of military patrols and merchant voyages	Policies regarding LCS package, threat, and merchant operations: communications protocols, engagement tactics, evasion tactics, etc.
γ	LCS Squadron, Hostile Threats, and Neutral Units	Squadron, antagonist fleet, and merchant fleet operations	Economics of navies, pirates, and merchant fleets	Policies regarding naval doctrine, naval campaign operations, merchant voyages, etc.
δ	Coastal Radio Stations Global Navigation Satellite Systems	National and Global operations in political institutions	Economics of international commerce	International maritime law Technical standards for communication and navigation systems

Agent-Based Simulation Modeling

Our DAF instantiation for the littoral operations scenario uses models described in three masters' theses from the US Naval Postgraduate School (Abbot 2008, Ozdemir 2009, and Jacobson 2010). Military and industry experts reviewed the theses prior to final approval and verified their descriptions of the functionality and performance parameters of the individual components of the system of systems.

These master's theses focused on various aspects of the system of systems architecture that are important for determining an acquisition strategy. Abbot (2008) simulated 41,195 littoral operations to address the best mix of a squadron of multiple LCS platforms with a mix of focused mission packages to ensure operational success across the broad range of challenges associated with littoral warfare. Abbot concluded that a squadron size of six to ten LCS produces the best results, and that a compositional rule of thumb of five LCS for the primary threat and two LCS for the secondary threat applies to each warfare area. Ozdemir (2009) compared the US Freedom class LCS with five similar international frigates and corvettes against small boat and submarine threats in confined waters and showed that the LCS is the most combat-effective performer, but that its cost detracts from its operational advantages. Jacobson (2010) concluded that the NLOS system is the best surface-to-surface missile system for the LCS while the Advanced Precision Kill Weapon System (APKWS) and Low-cost Guided Imaging Rocket (LOGIR) were the ideal air-to-surface missile system.

Our simulation focused on the operational capability of the SUW, ASW, and MCM mission packages. Antagonist threats include missile boats, mines, and submarines; friendly units include the LCS, MH-60R, UAV, RMV, and USV; and neutral units include merchant ships. Both friendly and threat units have a probability of terminating opposing units associated with their weapon systems while all units including neutral units have a probability of detecting other units in the simulation that are within a maximum detection range.

Agent Behaviors

Threat missile boats can perform basic surface search with a detection and classification range of 20 nm and 12 nm respectively. They transit at a speed of 8 knots, attack at 40 knots, and can travel at 15 knots when damaged. Missile boats transit and attack as a group for safety and cumulative strength. When a missile boat detects a friendly unit, it will pursue; they also evade when taking fire.

Threat submarines proceed on a patrol path and engage with torpedoes once an antagonist is detected. If they are fired upon, they commence evasion procedures by taking randomly drawn courses away from friendly forces.

Threat mines are static in nature and are dispersed over the simulation map. The mines simply detonate whenever an agent comes within a specified range of 1 nm, and destroys that agent in the process.

The neutral merchant agents can detect and classify targets at 20 nm, while travelling at a speed of 20 knots. The adding of merchants provides realism to the scenarios in that they add to the surface clutter for both friendly and threat sensors. Merchants do not carry any weapons, and the threat and friendly units are not interested in investigating or engaging the merchants.

The Littoral Combat ship agent with the surface warfare mission package (SUW LCS) can detect other units that are within 50 nm. LCS has a transit speed of 20 knots, and an attack speed of 40 knots. If damaged, LCS will be able to travel at its transit speed. The SUW LCS is equipped with two different types of weapon systems – the Non-Line-of-Sight Missiles and the Rolling Airframe Missile (RAM) air defense missile system. Upon detection of a missile boat, the LCS will pursue the detected threat and attempt to terminate it.

The SUW MH-60R acts as a scout for early detection of missile boats. It has a detection range of 75 nm and an operational speed of 144 knots. Once the MH-60R detects missile boat, it will pursue but will maintain a standoff distance of 20 nm due to the short reach of its weaponry and wait for the SUW LCS to close. Once the LCS has closed on the MH-60R, the MH-60R will approach the missile boat with the LCS. The MH-60R is assigned to an SUW LCS and will not pursue or attack anything other than a surface threat. The SUW MH-60R needs to refuel every 3.5 hours, and the refueling waypoint is located near a home base. When the SUW MH-60R is damaged, it returns to its LCS platform and is replaced by the SUW UAV.

The SUW UAV (Unmanned Aerial Vehicle) becomes active if the SUW MH-60R has returned to the LCS platform after being damaged. The UAV has a sensor range of 20 nm and transit at a speed of 80 knots. It replaces the MH-60R as a scout to detect surface targets for the SUW LCS but the UAV has no weapon systems to engage with a detected missile boat.

The Littoral Combat ship agent with the anti-submarine warfare mission package (ASW LCS) has the same detection radius, probability of detection and speed as the SUW LCS. Once a threat is detected, it will pursue. ASW LCS has weaponry to engage both surface and subsurface contacts, and it will engage submarines with a priority over threat missile boats. The ASW LCS maintains a 10 nm standoff from a detected submarine, while the MH-60R engages the submarine.

The ASW MH-60R is both a scout and pouncer for submarines. It has a detection range of 22 nm and an operational speed of 20 knots. The ASW MH-60R uses a dipping sonar to detect submarines; a tactic known as “sprint and drift.” In order to depict this tactic, an effective search rate results in an aggregate speed of 20 knots. The ASW MH-60R needs to refuel every 3 hours because of its search tactics. The ASW MH-60R only has three torpedoes, and once its torpedoes are expended, it transits to ASW LCS where it is given three more torpedoes and is able to re-engage submarines.

The ASW USV (Unmanned Surface Vehicle) has a detection range of 5 nm and a probability associated with detection. The USV has a speed of advance of 25 knots as they operate much like the ASW MH-60R to employ a dipping sonar. Once a submarine is detected by any of the ASW agents, the ASW USV will close to help localize the submarine, and pass the information to the ASW LCS for prosecution.

The ASW RMV (Remotely Manned Vehicle) has the same parameter values and operates on the same principle as the ASW USV. The only difference is that the RMV has a slower advance speed of 12 knots as compared to the ASW USV.

The Littoral Combat ship agent with the mine-countermeasure warfare mission package (MCM LCS) has detection radius of 50 nm and the same probability of detection and speed as the SUW LCS. While the MCM LCS has weaponry to engage both surface and subsurface contacts, it will engage mines with a priority over threat missile boats. The ASW LCS maintains a standoff from a detected mine and passes on the mine position information to the other LCSs in the squadron, while the MH-60R neutralizes the mine.

The MCM MH-60R is the only clearance platform available in the scenario. It has a sensor range of 5 nm and an advance speed of 20 knots. Unlike the ASW MH-60R, all the MCM supporting vehicles use towed body to counter mines. Once a mine is detected by or the position information is passed on to the MCM MH-60R, it will pursue and engage. It has to refuel every 3 hours and can only carry three clearance missiles at a time.

The MCM USV (Unmanned Surface Vehicle) and MCM RMV (Remotely Manned Vehicle) have the same parameter values as their ASW counterpart. None of the unmanned off board vehicles carries weapons, and is limited to pursuing the enemy and passing this detection to their respective LCS.

Simulation Scenario

We simulated a scenario that occurs in a 100nm by 150nm region, proceeds in time steps that equal 30 seconds, and that last no longer than 8 hours.

All friendly units start from their home base that is located in one diagonal corner of simulation map (100, 0), while the threat units have their home base in the opposite diagonal corner (0, 150). The threat force consists of five missile boats, five mines, and one submarine while the friendly fleet consists of all three LCS platforms. Three neutral merchant ships are dispersed randomly throughout the simulation map to provide realism to the scenario.

The SUW LCS starts the mission with its MH-60R airborne, and the ASW and MCM LCS have their MH-60R, RMV and USV deployed. The mission of the LCS fleet is to clear the waters of any missile boat, mines and submarine threats.

Results

Multiple simulation runs were carried out and in order to analyze the battle scenario and the agent behavior. Table 2 depicts the results from a typical simulation run.

Table 2. Results from a Typical Simulation Run

Agent Terminated	Agent Location	Time Step	Terminator	Terminator Location	Weapon Used
Mine 5	(75,40)	38	MCM MH60	(78.85,32.18)	Clearance
Mine 4	(65,50)	200	MCM MH60	(68.7, 44.9)	Clearance
Mine 3	(55,60)	272	MCM MH60	(58.70,59.89)	Clearance
Mine 2	(45,65)	621	MCM MH60	(40.58,65.77)	Clearance
Missile Boat 1	(17.8,111.18)	623	SUW LCS	(30.88,96.62)	NLOS
Missile Boat 2	(17.96,110.09)	635	SUW LCS	(28.25,99.63)	NLOS
Missile Boat 3	(18.65,112.08)	642	SUW LCS	(26.72,101.39)	NLOS
Missile Boat 4	(19.2,112.56)	651	SUW LCS	(24.75,103.66)	NLOS
Missile Boat 5	(18.10,108.88)	659	SUW LCS	(22.98,105.66)	NLOS
Mine 1	(20,105)	665	Self-Detonation	(20.32,105.52)	-N.A.-
SUW LCS	(20.32,105.52)	665	Mine1	(20,105)	-N.A.-
ASW LCS	(44.72,79.73)	763	Submarine	(44.69, 79.69)	Torpedo

The simulation results reveal an expected outcome for missile boats and most of the mines and non-intuitive outcomes for the submarine and one of the mines due to emergent properties. The SUW MH-60R helicopter detects missile boats early in the scenario because of its high velocity and detection radius. The SUW MH-60R maintains a standoff distance of 20 nautical miles from the detected missile boats and passes on the position information to the SUW LCS. Guided by the threat's position information, the SUW LCS was able to terminate each of the missile boats using its Non Line of Sight Launch System from distances ranging from 6 nm to 20 nm without suffering any fire in return. The missile boats stood no chance, as they were not able to get close enough to use their weapons, which had a very short range of 1 nm.

The MCM LCS detects most of the mines and passes on the position information to the MCM MH-60 and the other LCSs, so that they could avoid colliding with the mines. The MCM LCS maintains a standoff distance of 3 nautical miles from the detected mines while the MCM MH-60, guided by the antagonist's position information, terminates each of the mines using its Clearance Missiles. However, the MCM LCS was not able to detect Mine1 quickly enough, and since the SUW LCS has no means of detecting the mines on its own, it inadvertently collided with Mine1, which resulted in its termination. We can deconstruct the chain of actions leading to this mine explosion. After Mine 3 was terminated, the MCM MH-60R was almost out of fuel and it proceeded to the refueling point. During this refueling, which lasts 45 minutes, the MCM LCS was maintaining a standoff distance from Mine2 and waiting for the MCM MH-60R to engage the Mine2. Since Mine1 was located at a distance greater than 50 nm from Mine2, the MCM LCS was not able to detect it from its vantage point. By the time the MCM MH-60R terminated Mine2 and the MCM LCS resumed its search for other mines, it was too late and the SUW LCS unknowingly strayed within Mine1's proximity radius and was blown up.

Termination of the ASW LCS agent at the hands of the submarine was another emergent and non-intuitive result. The ASW LCS does not have the means to terminate submarines and was restricted to maintain a standoff distance of 10 nautical miles from a detected submarine and to pass on the submarine's location to an accompanying MH-60R. Once the detected submarine was within the weapon range of the ASW MH-60R, it fired torpedo missiles on the submarine. Because of the limited number of torpedo missiles onboard the ASW MH-60R and the probability of termination associated with the torpedo missiles, the ASW MH-60R either missed its target or could only injure the submarine, before it ran out of ammunition. The submarine was then able to terminate the ASW LCS before the ASW MH-60R could reload its torpedo missiles. Once the ASW LCS was terminated, the ASW MH-60R could no longer reload and the submarine was able to escape.

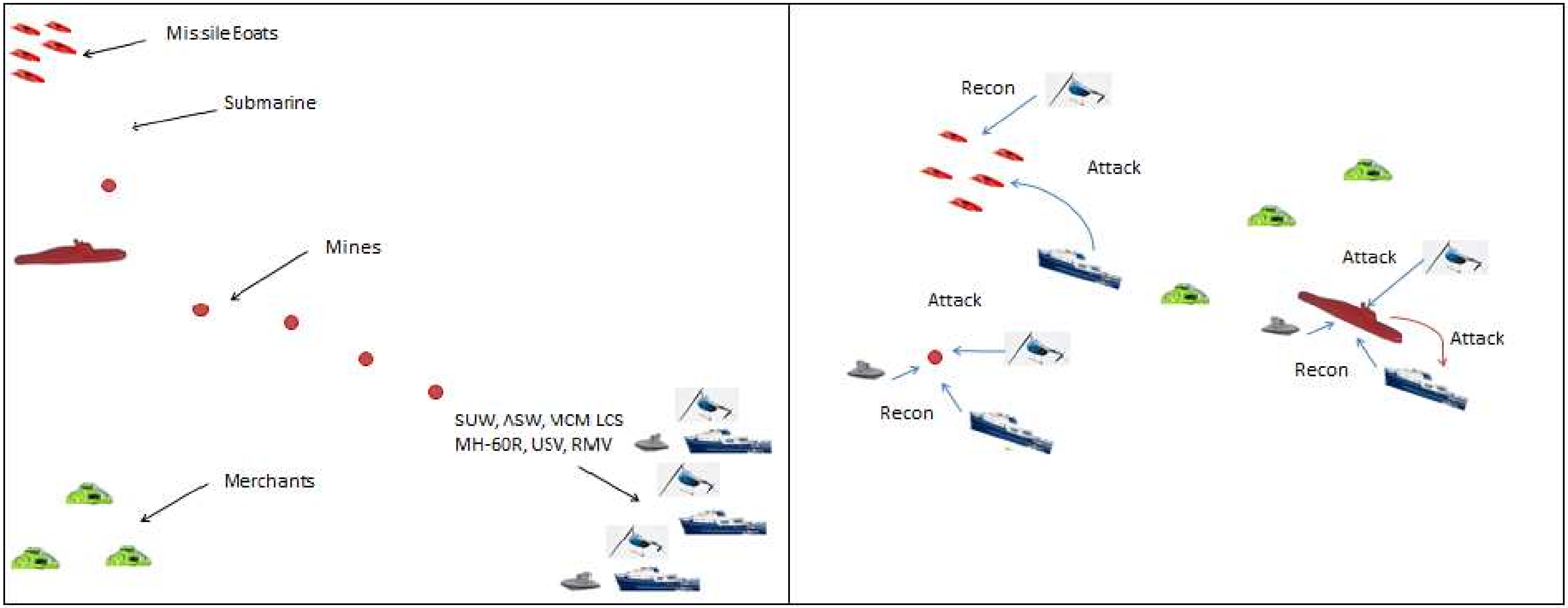

Figure 2. Starting Scenario and Emergent Scenario

Strategies

Given the previous result, we sought to implement a littoral operations strategy to tackle the threat units to minimize the friendly casualties suffered. The naval strategy was inspired by Admiral Lord Nelson's ploy in the famous Battle of Trafalgar which involved engaging an enemy fleet into a column directly perpendicular to the incoming enemy fleet. The LCS squadron approaches the hostile threats in a V-shaped pattern with the MCM LCS at the helm and the ASW LCS and SUW LCS bringing up the rear. From the analysis of the past results, it was clear one of the primary reasons for the losses suffered by the protagonist units was the fact that the LCSs moved ahead of the MH-60s when the latter ran out of fuel and had to fly to the refueling point. So in this instance the LCS squadron comes to a halt when the MH-60s are refueling and only start advancing when the MH-60s take off from their refueling point.

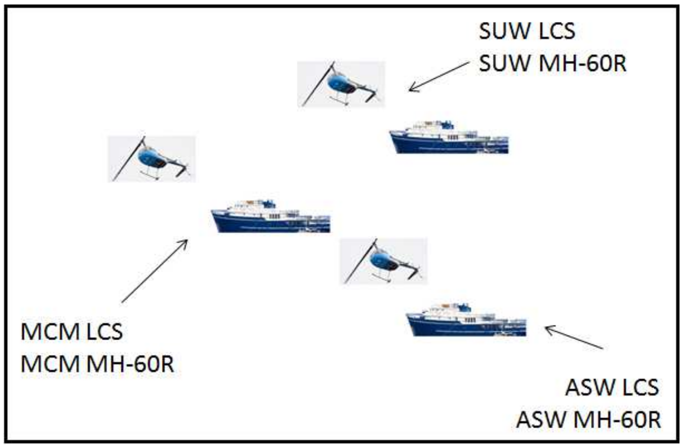

Figure 4. LCS Squadron Strategy

The simulation results reveal a much more favorable outcome for the protagonist units. The strategy adopted by the LCS squadron ensured that the ASW and SUW LCS stayed behind the MCM LCS and avoided any inadvertent collision with the mines. This time around, the ASW MH-60 was also able to terminate the submarine successfully. However the average mission time increased as the LCSs came to a halt when the helicopters were refueling. This is manifested in the extra 100 time steps (~50 minutes) it took for the SUW LCS to engage and destroy the missile boats.

Table 3 Results from a Strategic Simulation Run

Agent Terminated	Agent Location	Time Step	Terminator	Terminator Location	Weapon Used
Mine 5	(75,40)	59	MCM MH60	(74.03,39.98)	Clearance
Mine 4	(65,50)	102	MCM MH60	(67.24, 45.88)	Clearance
Mine 3	(55,60)	207	MCM MH60	(51.04,61.88)	Clearance
Mine 2	(45,65)	559	MCM MH60	(45.50,64.77)	Clearance
Submarine	(35.88,99.95)	612	ASW MH60	(40.76,96.90)	Torpedo
Mine 1	(20,105)	671	MCM MH60	(26.30,101.60)	Clearance
Missile Boat 1	(43.42,109.21)	739	SUW LCS	(49.82,91.10)	NLOS
Missile Boat 2	(39.45,107.85)	750	SUW LCS	(48.63,93.63)	NLOS
Missile Boat 3	(39.50,107.78)	751	SUW LCS	(48.49,93.90)	NLOS
Missile Boat 4	(39.56,107.81)	752	SUW LCS	(48.39,94.10)	NLOS
Missile Boat 5	(39.12,106.82)	757	SUW LCS	(47.98,95.06)	NLOS

Conclusions

Our agent-based LCS model provides a good insight into the workings of agent-based modeling and illustrates the systems-of-system nature of the problem at hand. A key characteristic of SoS problems is the unfolding of emergent behavior and properties during operations, because of the collective interaction among the system components. An agent model lucidly captures this aspect, and it provides inroads in capturing the operational capabilities and the shortcomings of the LCS.

Our simulation of the LCS combat scenario illustrated the ability of agent-based modeling to capture non-intuitive behaviors that may arise due to the complex dynamics that occur in interconnected systems of agents that follow a behavioral set of rules. Additionally, the modeling of individual agents permits the combination of discrete and continuous sets of rules within each agent – an attractive feature for hybrid systems that have elements of both components in operation. Because the rules for agents are defined at an individual entity level, there is no direct need to define aggregate modeling beyond the agents' individual behaviors and information connectivity between the agents.

The Past as Prologue

The planning guide for NASA's Systems Engineering Simulator describes "a real-time, crew-in-the-loop engineering simulator for the space station and advanced programs. It provides the ability to test changes to existing space vehicles and flight software, test the interaction of a new vehicle system with existing systems, create models of new vehicles (that may or may not exist yet) for engineering analysis, and evaluate display and control concepts and modifications. All of these functions are performed in a controlled, yet flexible, development environment. Models and capabilities developed for one customer can be used by other customers" (NASA 2011, 3). St. John, Moorman, and Brown (1987) describe how this simulator evolved in conjunction with the evolution of the NASA manned space flight systems of systems. It began in 1969 with analog computer-based hybrid simulations that were replaced with all-digital computer simulations and at the time provided the capability for large, multifunction Space Shuttle simulations designed for long-term operations support the Shuttle Program. Over the years, NASA has added several systems to the mix of systems such as the International Space Station, the Space Stations Remote Manipulator System, the Multimission Space Exploration Vehicle, the Orion Crew Vehicle, and the SpaceX Dragon.

The Systems Engineering Simulator originated prior to the emergence of systems of systems and agent-based simulation as important concepts in systems engineering. Nonetheless, the simulator is a modular, agent-based simulation that includes hardware mockups, human operators, and software agents, and it has been vital to the successful engineering and operations of a system of systems for over 40 years. We certainly hope that the agent-based modeling that is in our discharge today will have the same lasting impact:

> Whereof what's past is prologue, what to come
> In yours and my discharge. (Shakespeare, *The Tempest*, Act 2, Scene 1)

References

Abbot, B.P. 2008. "Littoral Combat Ship (LCS) Mission Packages: Determining the Best Mix." Master's thesis. Naval Postgraduate School.

Bonabeau, E. 2002. "Agent-based Modeling: Methods and Techniques for Simulating Human Systems." *Proceedings of the National Academy of Sciences of the United States of America* 99(suppl. 3): 7280–7287.

Borshchev, A., and A. Filippov. 2004. "From SD and Discrete Event to Practical AB Modeling: Reasons, Techniques, Tools." Paper presented at the 22nd International System Dynamics Conference, Oxford, UK, 25–29 July.

Chow, R., D. Braun, and D. Fry. 2012. *DAF: Discrete Agent Framework Programmer's Manual*. West Lafayette, US-IN: Purdue University.

Clymer, J. R. 1997. “Engineering of Intelligent, Complex Adaptive Systems Using an Agent-Based Architecture.” Paper presented at the Seventh Annual International Symposium of INCOSE, Los Angeles, US-CA, 3–7 August.

DeLaurentis, Dan, and Robert K. “Cab” Callaway. 2004. "A System-of-Systems Perspective for Public Policy Decisions." *Review of Policy Research* 21 (6): 829-837.

DeLaurentis. 2005. “Understanding Transportation as System-of-Systems Design Problem.” Paper presented at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, US-NV, 10–13 January.

Eisner, H., J. Marciniak, and R. McMillan. 1991. “Computer-Aided System of Systems (S2) Engineering.” Paper presented at IEEE / SMC International Conference on System, Man, and Cybernetics, Charlottesville, US-VA, 13–16 October.

Eisner, H., J. Marciniak, R. McMillan, and W. Pragluski. 1992. “RCASSE: Rapid Computer-Aided System of Systems (S2) Engineering.” Paper presented at Second Annual International Symposium of INCOSE, Seattle, US-WA, 20–22 July.

Forrester, Jay. 1958. “Industrial Dynamics: A Major Breakthrough for Decision Makers.” *Harvard Business Review* 36(4): 37–66.

Giachetti, R., V. Marcelli, J. Cifuentes, and J, Rojas. 2012. “An Agent-Based Simulation Model of Human-Robot Team Performance in Military Environments.” *Systems Engineering,* DOI: 10.1002/sys.21216.

Hsu, J. C., and M. Butterfield. 2007. “Modeling Emergent Behavior for System-of-Systems.” Paper presented at the 17th Annual International Symposium of INCOSE, San Diego, US-CA. 24–28 June.

Hsu, J. C., M. Price, J. R. Clymer, and E. Gonzalez. 2009. “Agent-Based Modeling the Emergent Behavior of a System-of-Systems.” Paper presented at the 19th Annual International Symposium of INCOSE, Singapore, SG. 20–23 July.

Jacobson, K.R. 2010. “Littoral Combat Ship (LCS), The Littoral Combat Ship (LCS) Surface Warfare (SUW) Module: Determining the Best Mix of Surface-to-Surface and Air-to-Surface Missiles.” Master’s thesis, Naval Postgraduate School.

Kilicay-Ergin, N., and C. Dagli. 2008. “Executable Modeling for System of Systems Architecting: An Artificial Life Framework.” Paper presented at the SysCon 2008 IEEE International Systems Conference, Montreal, CA, 7–10 April.

Maier, M. 1996. “Architecting Principles for Systems-of-Systems.” Paper presented at Sixth Annual International Symposium of INCOSE, Boston, US-MA, 7–11 July.

Maier, M. 1998. “Architecting Principles for Systems-of-Systems.” *Systems Engineering* 1 (4): 267–284.

Marimon, R., E. McGrattan, and T. Sargent. 1989. “Money as a Medium of Exchange in an Economy of Artificially Intelligent Agents.” Santa Fe Institute: Santa Fe, US-NM.

Ozdemir, O. 2009. “Evaluation and Comparison of Freedom Class LCS And Other Frigates/Corvettes Against Small Boat, FPB And Submarine Threats in Confined Waters.” Master’s thesis. Naval Postgraduate School.

O’Rourke, R. 2012. Navy Littoral Combat Ship (LCS) Program: Background, Issues, and Options for Congress. Retrieved from http://www.fas.org/sgp/crs/weapons/RL33741.pdf

Rouse, W.B. 2003. “Engineering Complex Systems: Implications for Research in Systems Engineering.” *IEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews* 33(2): 154–156.

Schieritz, Nadine and Peter M. Milling, 2003. “Modeling the Forest or Modeling the Trees Comparison of SD and AB Simulation.” Paper presented at the 21st International System Dynamics Conference, New York, US-NY, 20–24 July.

Biographies

Ankur Mour is a Graduate Research Assistant in Purdue's School of Aeronautics and Astronautics in West Lafayette, IN (US). He is a recent graduate from the Indian Institute of Technology Bombay (India) where he completed his Bachelor's in Aerospace Engineering with a minor in Management Studies. He is currently a member of the Center for Integrated Systems in Aerospace and the System-of-Systems Laboratory. His primary research interests are in the areas of agent-based modeling, discrete-event simulation, multidisciplinary optimization, and computational fluid dynamics.

C. Robert Kenley is a Research Scientist in Purdue's School of Aeronautics and Astronautics in West Lafayette, IN (US). He has over thirty years' experience in industry, academia, and government as a practitioner, consultant, and researcher in systems engineering. He has published papers on systems requirements, technology readiness assessment and forecasting, Bayes nets, applied meteorology, and the impacts of nuclear power plants on employment.

Navindran Davendralingam is post-doctoral researcher at Purdue's School of Aeronautics and Astronautics. He works as part of the System-of-Systems Laboratory led by Dr. DeLaurentis and received his Ph.D. in Aerospace Engineering from the Purdue University in 2011. Dr. Davendralingam currently works on projects that are funded by the DoD Systems Engineering Research Center UARC and the Naval Postgraduate School. His primary research includes the use robust optimization methods from financial engineering and operations research in dealing with uncertainties in systems and system of systems engineering problems. He is a member of the American Institute of Aeronautics and Astronautics and the Institute for Operations Research and Management Sciences.

Daniel A. DeLaurentis is an Associate Professor in Purdue's School of Aeronautics and Astronautics in West Lafayette, IN (US). He is Director of the Center for Integrated Systems in Aerospace, which is home to over 20 additional faculty and staff and pursues advancements in methods and tools for optimal integration of systems and data across aerospace and defense applications. He also leads the System-of-Systems Laboratory. His primary research interests are in the areas of problem formulation, modeling and system analysis methods for aerospace systems and systems of systems, with particular focus on network analysis and agent-driven models.