12th ICCRTS:
"Adapting C2 to the 21st Century"

Informing Joint C2 System-of-Systems Engineering with Agent-Based Modeling: An Analysis and Case Study

Topic Categorization:
Modeling & Simulation

Chuck Lutz
System of Systems Engineering
Lockheed Martin MS2
Moorestown, NJ 08057
charles.d.lutz@lmco.com
(856) 638-7234

Dr. Greg Schow
Mission Systems
Lockheed Martin MS2
Moorestown, NJ 08057
greg.schow@lmco.com
(609) 326-4114

Mitchell Kerman
System of Systems Engineering
Lockheed Martin MS2
Moorestown, NJ 08057
mitchell.c.kerman@lmco.com
(856) 638-7223

Michael DiMario
Mission Systems
Lockheed Martin MS2
Moorestown, NJ 08057
Michael.j.dimario@lmco.com
(609) 326-4102

Ambrose Kam
System of Systems Engineering
Lockheed Martin MS2
Moorestown, NJ 08057
ambrose.kam@lmco.com
(856) 638-7234

Informing Joint C2 System-of-Systems Engineering with Agent-Based Modeling: An Analysis and Case Study

Dr. Greg Schow
Michael DiMario
Mission Systems

Ambrose Kam
Mitchell Kerman
Chuck Lutz
System of Systems Engineering
Lockheed Martin MS2
Moorestown, NJ 08057

Abstract

….

This paper illustrates the benefits of "informing" the systems engineering process through the use of Agent-Based Modeling. While modeling and simulation tools are often employed in the systems engineering process, their typical use provides additional information to the decision makers. Thus, these tools are used to "inform" decision makers within the systems engineering process.

An analysis was recently conducted in order to identify unanticipated effects on Composite Combat Identification (CCID) Reasoning Algorithm (CRA) within a distributed system-of-systems (SoS) Integrated Architecture Behavior Model (IABM) environment. An Agent-Based Model (ABM) simulation approach was used to evaluate potential emergent behavior. CRA, based on a Dempster-Shafer evidential reasoning algorithm, was recently introduced in the literature. Studies of its effectiveness have been limited to a single instance of the algorithm.

In investigating distributed CRA effectiveness, we considered the behavior of sensor and CCID information sharing across the network. In particular, network delays deliver distributed sensor information to CRA nodes at different times, skewing CCID recommendations for a given track from individual nodes. Discrepancies must be resolved by some suitable arbitration scheme. We considered four schemes in our study: Weighted Bayesian, Naïve Bayesian, Majority Voting, and Maximum Belief Value.

Distributed SoS architectures famously exhibit "emergent" behavior that is practically impossible to model via fully scripted simulations. We chose to apply ABM to allow us to capture this behavior, to evaluate our distributed SoS architecture from "the bottom up". In this environment, we were able to study the effects of variations in network delay and arbitration scheme on distributed CRA performance.

….

Introduction

"System of Systems"

In recent years, within the systems engineering community, the terms "System of Systems" (SoS) and "System of Systems Engineering" (SoSE) have emerged in response to the growth in complexity and scope of technology solutions produced for use in many domains. The defense industry is a typical customer of the type of work done by systems engineers, and a recent shift in focus in military acquisitions from "systems" to "capabilities" is another factor contributing to the emergence of these terms. The United States Department of Defense (DoD) and their Joint Vision 2010 (JV2010) and 2020 (JV2020) describes the US military and its goals for the 21st century. This vision assumes and articulates the necessity for multinational and US interagency cooperation and interdependent joint command and control (JC2) for a full range of capabilities operationally, organizationally, doctrinally, technically, and intellectually [1], [2]. Information superiority (IS) is a key enabler of the Joint Vision (JV) and the evolution of JC2 as it integrates the core competencies of the multiservices into joint operations. Joint Vision is the optimal integration of all joint forces which in turn creates additional complexity to operations. It is this complexity and interdependent capabilities that will achieve full spectrum dominance through independent multinational and multiagency operations acting in concert.

Acting in concert and remaining independent requires a SoS whereby the independent agencies and multinational systems dynamically merge to form a holistic system. The result is a SoS that has greater capabilities than the sum or simple combination of the individual multinational or agency systems. JV relies on SoS to bridge between existing legacy and new systems. The literature describes several SoS definitions and attributes. The attributes common to SoS are generally agreed [3], [4], [5]. These include tendency for "emergent behavior" and composition from individual systems that can operate independently, the developments of which are managerially independent from the overall system development effort.

Systems engineering has continuously evolved to develop ever more complex systems, but faces an inability to address problems of ambiguity, multiple and diverse contexts, and nonlinear relationships with other systems [6]. These systems are described as complex adaptive systems (CAS) and are composed of agents interacting with one another and their external environment. They interact in such a way to display stimulus-response behavior [7]. Agents adapt by changing their rules as they learn producing emergent behavior. CAS holds the properties of nonlinearity, difficult or impossible long-term prediction, seemingly spontaneous formation and dissolution of groups, complex communications between groups and individual entities, and the ability of individuals to reason about their environment. The concept of evolving agent behavior and ability is a key concept in CAS - the dynamics of emergence in both new species and aggregate behavior between individuals are well-studied areas.

Agent Based Modeling (ABM) may be used for simulating these complex nonlinear systems and their interactions thus representing the real world. The emergent

behavior as a result of the learning acquired by the collective action of the agents may be observed and assessed [8], [9], [10]. Agent-based models are characterized by systems of autonomous individuals (the "agents") situated within some contextual world that provides the agents sensory input of interest to the modeler. An individual agent's behavior is usually specified via fairly simple rules for sensing what is in the environment around it and reacting to stimuli. Agents tend to act in goal-directed ways. Agents may also be programmed for learning and memory; i.e. the rules they use can change over time as the agent adapts.

The popularity of ABM has been rising in fields such as biology, finance, physics, and homeland security, to name a few – even springing forth newly identified fields such as computational sociology. Many conventional agent-based models in these applications are populated by tens, hundreds, thousands, or perhaps even millions of identical agents, though in our work at MS2, we are typically interested in smaller populations of sets of different types of agents composed into heterogeneous groups, such as naval fleets and strike groups.

The emergence of Agent Based Modeling offers a means to create a validated conceptual model of a System of Systems, or architecture. Validated in the sense that the analysis was developed systematically, viewed from many angles and discovery encouraged. In order to evaluate the relative merits of alternative solutions, a dynamic view of a system's architecture is needed in order to verify completeness of the system [11], [12], [13]. You don't know what you don't know until the system is perturbed over time [14], [15]. Identifying errors early in a system's development is crucial for successful deployment [16]. For a Systems of Systems analysis the problem is exacerbated by the increased complexity inherent in a System of Systems.

Modeling and Simulation: Conventional and Agent-Based

In our modeling and simulation (M&S) work, many of our SoS studies are performed using conventional discrete event system simulations, which have an event-driven execution model and involve the specification of scripted behavior of entities acting within the simulations, such as naval forces performing various missions. By "scripted", we mean that entities follow generally planned behavior patterns, such as moving within designated operating areas and receiving and giving orders according to predetermined command structures. Simulation "hardware" assets (ships, airplanes, unmanned vehicles, etc.) have various sensor and weapon models at varying levels of fidelity attributed to them, and "software" aspects (e.g. command and control) employ basic sets of engagement rules for achieving situational awareness and reacting to enemy forces.

Recently, we have been introducing ABM into our repertoire of M&S approaches for complex SoS studies. The payoff in applying ABM comes from the dynamics inherent in such CAS as described. From simple individual behaviors, complex macroscopic behaviors emerge, and higher-order behavioral structures are observed that would be impossible or difficult to "script" or program in a conventional simulation. Difficulty of implementation aside, the "surprise" factor of ABM simulations is

something largely unachievable via other techniques – we cannot script or program large-scale behavior patterns that we cannot imagine, expect, or predict in the first place. The information content of agent models can be quite high.

The execution model of ABM simulations is typically time-driven as opposed to event-driven models such as found in discrete-event simulations. Due to the resources required to simulate many agents' behavior in parallel, agent sensor and effecter (movement, weapons) models are typically of lesser fidelity compared to conventional effects- or physics-based simulations. However, the emphasis is on the specification of the agents' "reactive" behavior - the richness of their internal processing of sensor data and their adaptability can be greatly enhanced compared to conventional simulations.

Of course, conventional simulations as discussed above involve entities acting within some simulated world, programmed to react in certain ways to external stimuli, and thus have "agent-based" aspects. However, Agent-Based Modeling and Simulation proper, as an established, distinct sub-field of M&S, typically employs tools and analytical techniques specifically tailored to working within the agent paradigm. Agent behavior specification is mostly rule-based and reactive, and agent interactions are typically "dense" in time, as opposed to scripted behaviors within conventional simulations in which interactions between entities may be sparse in time.

Our interest is in the use of ABM in the evaluation of SoS architectures for military applications. As mentioned, one attribute of SoS upon which the systems engineering community seems to agree is that they exhibit "emergent behavior"; thus, ABM is an excellent choice for investigating the complexities inherent in transitioning from "just plain systems" to "systems of systems". Furthermore, modeling and simulation is typically employed within systems engineering studies in order to increase the amount of information available to decision makers. ABM provides a viable means of enhancing the amount of available information, thus "informing" the systems of systems engineering process.

Case Study and Problem Statement

Since the Architecture and algorithms being evaluated were still in development we decided that the only way to evaluate the System of Systems would be to take an exploratory modeling approach. In exploratory modeling, the results of a model run are viewed as the results of an experiment that tells us what the outcome would be if all the assumptions we had to make in setting up the model turned out to be true. The result will be the representation of a Probability Density Function wherein no single outcome exists. The models & simulation only need to be plausible and consistent with what is known. In effect this approach inverts the problem, like sensitivity analysis, in a vastly more complex problem space, where the bounds of where a model is valid is determined vice determining particular solutions [17].

We applied the ABM SoS engineering analysis approach to the problem of distributed Composite Combat Identification (CCID). Interest in decreasing fratricide and other problems related to misidentification of entities and objects within a battlespace has led to research in automated combat identification.

Current Naval operations in combat identification involve platform-level activities to coordinate combat identifications throughout the platform – and in some cases across platforms – to try to form a cohesive operating picture as to the nature and disposition of forces. These activities are usually manual and require reconciliation of multiple forms of input data [18].

One recent result in automating this process is the Office of Naval Research's Composite Combat Identification Reasoning Algorithm (CRA) effort, which employs a Dempster-Shafer evidential reasoning approach to automatic combat ID [19]. Given a multitude of sensory and other inputs, the algorithm makes a recommendation as to the combat ID of a track. The scope of the ONR study was limited to investigating the functioning of one CRA node in isolation – essentially the deployment of the CRA on one platform.

In addition to plans to introduce of technologies like advanced automation for combat ID [21], the US Navy is also moving to deploy enhanced communications and networking abilities to the field. This effort is in accordance with the Joint Single Integrated Air Picture (SIAP) Systems Engineering Organization (JSSEO) initiative [20], [21]. This includes development of an Integrated Architecture Behavior Model (IABM) to capture the requirements for future integrated joint architectures supporting fully networked sensors and data available across the force. Integrated automated combat ID capabilities are part of the effort.

The precursor to our study was our interest in the impact of network and other effects on coordinated, distributed CRA. Assuming an environment in which CRA nodes are deployed to various air and surface platforms across a force and within the SIAP environment, we are interested in:

- The effects of network delays on local CRA combat ID recommendations taking sensor and other input data obtained from many nodes across the force
- Resolution of combat ID recommendation "disagreements" between distributed CRA nodes processing the same logical track within the SIAP

In the first case, network delays of some kind are to be expected even under the best bandwidth and quality-of-service (QoS) conditions. In the SIAP environment of networked sensors, any given CRA node would have available to it sensor data about a particular SIAP track culled from across the force. Due to network and processing delays, some data would arrive to the node later than other data. We are interested to know what impact this has on the resulting CCID recommendation for the track in question.

In the second case, within the SIAP environment, naturally the results of CRA processing (i.e. the combat ID recommendations) would be vital information to be disseminated across the force according to the distribution capabilities captured in the IABM. When multiple conflicting recommendations are produced for a given common SIAP track, how are force nodes to collectively resolve the differences and come to a single shared conclusion as to the true identity of the track?

To proceed in building an agent-based model of this situation, we defined two concepts that capture aspects of the second area of concern:

- Consensus Group: Given N CRA-enabled nodes that have processed a SIAP track, we will be able to divide the N nodes according to their CCID recommendations into a number of groups G < N, where all the nodes in a group have produced the same recommendation (i.e. they "agree"). Ideally, G = 1 and all nodes agree.
- Arbitration Scheme: When G > 1, the CRA nodes require some way of resolving the differences and coming to agreement on a final CCID recommendation. We call the method by which they do this an *arbitration scheme*.

Next, we survey some military uses of ABM in the literature, then discuss further details of how we conducted our case study, and end with our conclusions.

Case Study

Model Design

We implemented our case study model in an ABM environment [22]. We modeled various types of air, surface, and subsurface platforms as individual agents, with simplified sensor models and simulated CRA processing. Agents could communicate "full mesh", but with communication delays based on distance.

Our agents had ground-truth combat ID attributes as follows:

- "Allegiance": One of FRIEND, FOE, or NEUTRAL
- "Nationality": Generic nation names; one of "FRNAT1", "FRNAT2" (friendly nations), "NUNAT3", "NUNAT4" (neutral nations) and "FONAT5", "FONAT6" (foe nations)
- "Type": Various known and invented-generic air, surface, and subsurface platforms, e.g. "SH-60B", "DDG", and "Red Submarine"

The real CRA uses a database of various combat ID attributes and calculates probabilistic "belief values" for the corresponding attributes of tracks under consideration, then produces its recommendations based on the resultant belief values.

Toward this end, we created eight generic organic sensor types that a platform (agent) could have and assigned each type a weighting value that would indicate its contribution to a generated belief value. Different agents were assigned mixes of these sensors, i.e. an agent could have from one to eight sensors onboard. This allowed us to model the varying effects of input from different sensors on each type of combat ID attribute. For example, IFF (Identify Friend or Foe) has greater impact on the Allegiance and Nationality attributes than on Type. For the sake of simplicity, we used one overall "on or off" detection radius for all of an agent's sensors; i.e. if another agent came within this detection radius, our first agent would be able to detect its presence.

We modeled a CCID recommendation report as having the following fields:

- Track ID
- Allegiance recommendation
- Allegiance belief value
- Nationality recommendation
- Nationality belief value
- Type recommendation
- Type belief value

The Track ID is an integer identifying the track number to which the report pertains.

Belief values in our model were values between 0 and 1, as expected, to four decimal places. The recommendation fields above have values from the enumerations described previously (e.g. “FRIEND”, “FRNAT2”, “DDG”).

We were interested also in the effects of blue vs. red force levels on the arbitration process. That is – given a fixed number of ground truth red (threat) forces, what would the effect on arbitration be of varying the number of blue (friendly) force nodes producing tracks and CRA reports? Given a fixed number of blue forces, what would the results be of varying the number of red (hostile) and white (neutral) forces?

We created two groups of heterogeneous sets of agents, varying the number of blue forces in the first group, and varying the number of red and white forces in the second group. These data sets formed one dimension of our experimentation plan’s run matrix, and are duplicated below in Table 1.

Matrix One:

Blue	Red	White	Total
2	25	25	52
5	25	25	55
10	25	25	60
15	25	25	65
20	25	25	70

Matrix Two:

Blue	Red	White	Total
10	5	5	20
10	10	10	30
10	15	15	40
10	20	20	50
10	25	25	60
10	30	30	70

Table 1. Agent Population Sets Used in Experimentation

For each simulation run, we instantiated one of the agent sets and randomly distributed them in the agent model world. Our study did not consider effects of natural obstacles, weather, etc. on communications, thus, there was no land masses or other natural objects represented; our model world simply consisted of a large area of open ocean.

Each agent was initialized with a random direction and speed. This approach allowed the agents to simply move through the space at random, with other agents approaching and receding from an agent’s detection radius.

The environment we used models time in “ticks”. Each tick constitutes one increment of processing. In modeling in this environment, one typically produces a main processing loop procedure (usually called “go”) that handles invocation of all of the agent actions for the iteration, plus any global post-action processing and metrics calculation and gathering that needs to be done.

One event loop for an agent comprised the following high-level sequence of actions:

- Detect all agents within the detection radius
- Produce organic track reports for each detected agent
- Publish organic tracks to the network
- Retrieve all tracks from the network to obtain a full set of tracks
- Fuse track reports for identical tracks to obtain a minimized set of tracks
- Generate CRA reports for each track
- Publish the CRA reports to the network
- (globally) Arbitrate on the network-wide set of CRA reports

We will not go into all details of the processing here. We will mention that all network transmission delays were modeled by assigning each agent “delay queues” for track and CRA reports, into which time-stamped track or CRA report receptions, respectively, were inserted. During iterations, agents would determine the distance to each other blue CRA-enabled platform and insert their organically produced track or CRA reports in each agent’s queue, with a timestamp indicating the future reception time scaled by distance. Each agent would then check its queue and retrieve all tracks or reports it was “allowed” to process based on the timestamp. Note that the fundamental “tick” unit of time must map to some atomic time value of meaning in the domain of the model. We wanted the time delays as a function of distance to be measured on the order of milliseconds.

A view of the agent world for our model can be seen in Figure 1. Here we have turned on an indicator of the agent ground truth for Allegiance, shown by the agent’s color. For cosmetic purposes, we assigned icons to the agents roughly indicating their type. We also turned on ring indicators showing the sensor range for each agent.

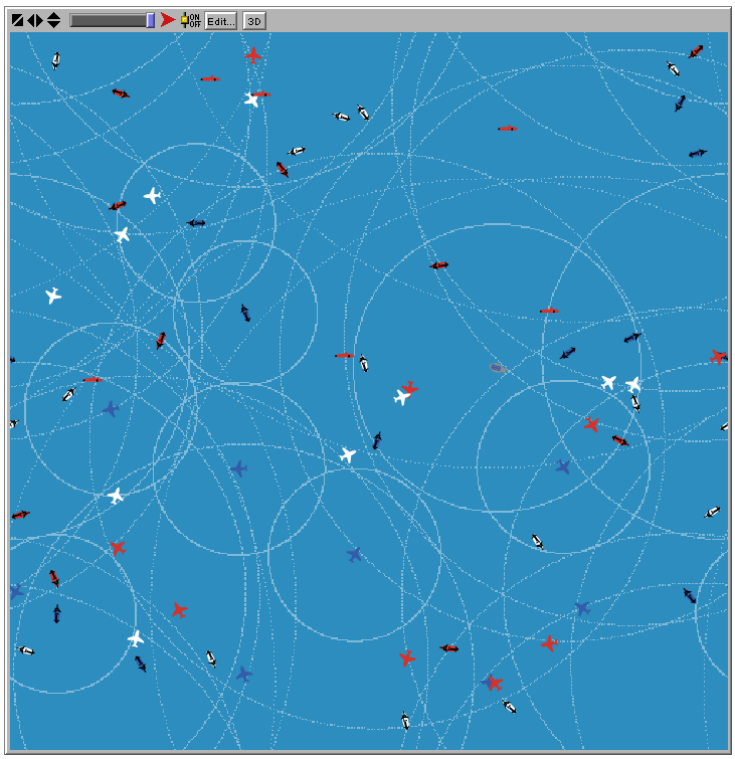

Figure 1. Agent world

We nominated four candidate arbitration schemes for trial in our experiments, expecting that during experimentation we might find differences in their performance at deriving final reports most closely matching ground truth:

- *Majority Voting:* At the end of an iteration, the number of CRA report "votes" of different kinds per track is counted, and the report with the most "votes" is chosen as the "winner".
- *Maximum:* Per track, the belief values for each CCID attribute are compared across all CRA reports; the final accepted CRA report is a hybrid, consisting of the maximum belief values per attribute type across all reports.
- *Naïve Bayesian:* The track's final CRA report is based on the median of the CRA report belief values per attribute across all reports for the track.
- *Weighted Bayesian:* The track's final report is based on a weighted combination of CRA report belief values across the report group

The choice of arbitration scheme is another dimension in our run matrix.

Hypothesis

In our experiment runs, we expected to find that as network transmission time delays increased, the ability to maintain a single global, shared track picture is diminished.

- That by using an ABM model with assumed behaviors in a model of IABM as understood at the time we would be able to verify our assumptions and identify any unanticipated emergent System Of System complications.
- With no time delay, all agents would receive the same track and CRA report information simultaneously, thus, the number of consensus groups during the entire simulation run would remain constant at one
- As communication delays increase, the average number of consensus groups would increase, and the number of consensus groups formed at the end of each iteration would depend on the particular geometry at that moment.

Experiment Plan

We ran two sets of simulation runs, one for each agent population mix matrix as described above in Table 1. The first set dealt with varying the blue forces, the other with varying the red and white forces. In addition to the agent population mix, the other dimensions in each run matrix were the maximum time delay between agents and the arbitration scheme used.

Maximum time delay was specified as 0, 1, 6, and 12 milliseconds. These values correspond to the maximum possible delay between two agents. The agent world in the modeling environment can be configured to "wrap" in both the X and Y dimensions (i.e. it forms a torus). We used this configuration with a 501x501 patch world, with each patch representing one nautical mile. In this configuration, two agents placed at opposite corners diagonally are the farthest from each other possible, or 501 nm * $\sqrt{2}$, approximately 708 nautical miles.

The four arbitration schemes were as before: Majority Voting, Maximum, Naïve Bayesian, and Weighted Bayesian.

In summary:

	Run Matrix One	**Run Matrix Two**
Number of Population Sets	5	6
Number of Maximum Time Delay Settings	4	4
Number of Arbitration Schemes	4	4
Matrix Size (Runs)	80	96

Table 2. Run matrices for the experiments

We collected the following metrics within each run:

- Average, median, and ceiling of the number of consensus groups across all tracks for the entire run
- Overall correctness (percentage) compared to ground truth track data
- Percentage of reports that had each individual attribute (Allegiance, Nationality and Type) correct
- Percentage of reports that had 0, 1, 2, or 3 out of 3 attributes correct

Each run consisted of 1000 iterations ("ticks"). At the end of an iteration, we compared each final CRA report (post-arbitration) against ground truth and scored it a 0, 1, 2, or 3, depending on how many attributes were correct. This score was recorded in a global accumulator list. At the end of a run, we summed these scores and divided by the number of reports times 3 (as there are three attributes per report). That is,

$$OverallCorrectness = \frac{\sum_{i=1}^{R} score_i}{R * 3},$$

where R is the number of final CRA recommendation reports generated throughout the run. Per-attribute correctness was calculated similarly.

We used a feature of the ABM modeling environment that facilitates automatic setup and execution of experiment run matrices. The generated metrics data can be saved to an Excel file for later analysis.

Results

Here we present a subset of our simulation result data.

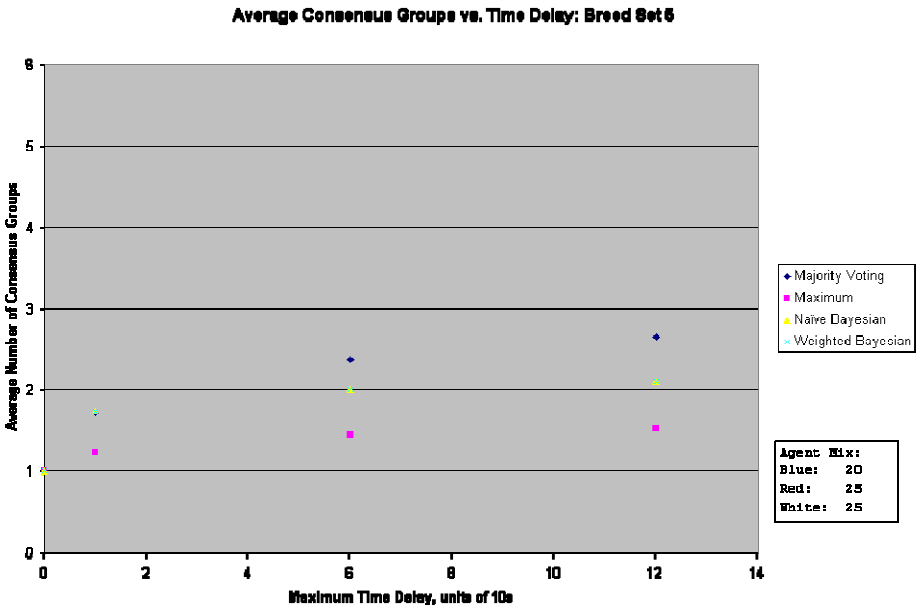

Figure 2. Average Consensus Groups vs. Time Delay for one Agent Population Set

Figure 2 shows the average number of consensus groups plotted against time delay for one of the agent populations (the mix was 20 blue agents, 25 red and 25 white). As is shown in the graph, we found that the "Maximum" arbitration scheme generally

produced the fewest consensus groups across most runs. Note that for zero delay, the number of groups is always one, as expected.

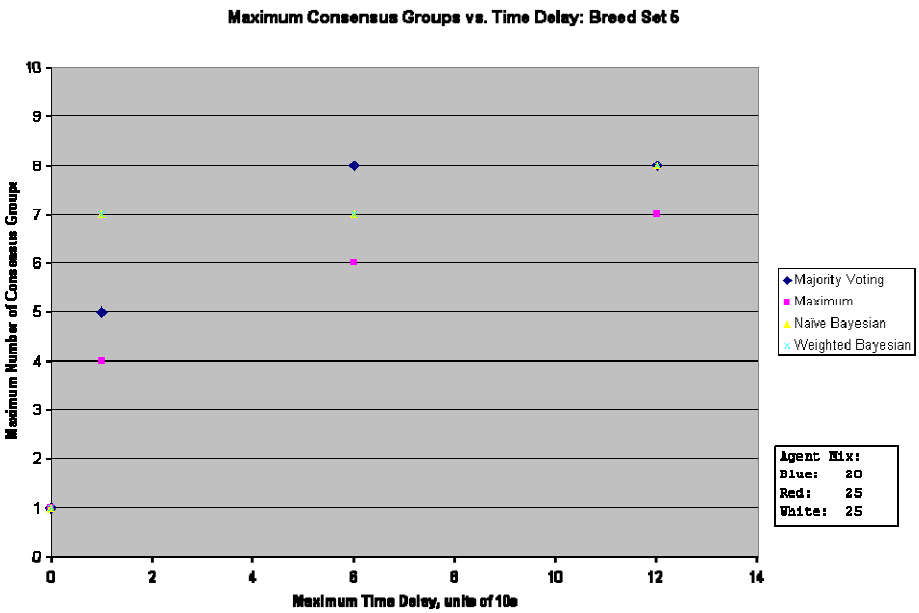

Figure 3. Maximum Consensus Groups vs. Time Delay for one Agent Population Set

Figure 3 shows the corresponding graph of maximum consensus groups encountered during the run across all tracks, per time delay and arbitration scheme. Again, the “Maximum” scheme generally produced the best results, with four, six, and seven maximum consensus groups encountered for one, six, and twelve millisecond maximum delays, respectively.

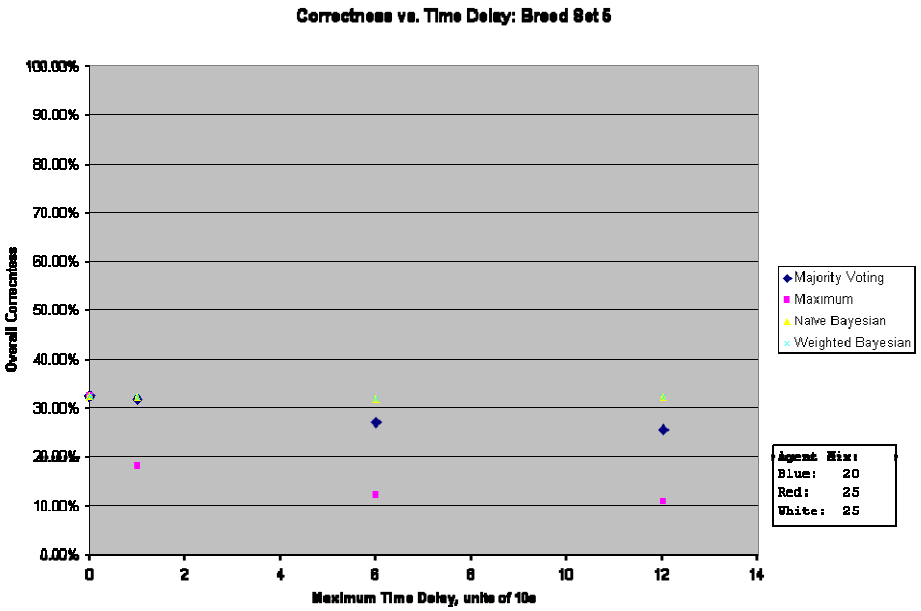

Figure 4. Correctness vs. Time Delay for one Agent Population Set

Figure 4 shows the overall correctness (calculated as described previously) plotted against time delay. In this case the Maximum arbitration scheme actually fared the worst, with a near tie between the Naïve and Weighted Bayesian schemes for the best correctness. The overall very low correctness performance is of course heavily influenced by our use of a simulated CRA function.

Figure 5 tabulates our findings for correctness, normalized over all runs. In this analysis, Naïve Bayesian fared best overall.

Normalized Correctness over All Runs		
Scheme	Overall	Rank
Majority Voting	0.996	3
Maximum	0.746	4
Naïve Bayesian	1.000	1
Weighted Bayesian	0.997	2

Figure 5. Normalized Correctness Results over All Simulation Runs

In summary, we found that the Naïve Bayesian arbitration scheme provided perfect correlation between the determined values and ground truth. By using a uniformly distributed random seed, the Naïve Bayesian and Weighted Bayesian performance was essentially the same.

All in all, we feel that our experimentation confirms the applicability of the Agent-Based Modeling approach to evaluating C2 in System-of-Systems architectures.

Conclusions and Recommendations

The Utility of the Approach

We feel that there is great utility to be had in using ABM for military applications and evaluation of systems or subsystems within SOS architectures. By specifying agent behavior at the individual level, we are able to perform "bottom up" vs. "top down" evaluations offered by conventional simulation. ABM lends itself to the investigation of distributed behavior coordination problems and for identifying inconsistencies in the distributed architecture.

SOS engineering is about identifying and understanding emergent behavior. ABM is an excellent way of revealing SOS emergent behavior that might be otherwise obscure from traditional modeling methodology. By exploring the effects of aggregate behavior, the ABM model allowed us to look into unanticipated results. Identifying trends and patterns that emerge can build confidence and insight in the SOS concept under evaluation.

Areas of Further Research

We believe that ABM approach to M&S can provide insight into the hard-to-capture qualities of SOS. In future agent models, we expect we will incorporate agent learning and adaptability and variations in the fidelity of various sensor and effector capabilities. These are some key enhancements to realizing the full potential of the technique. For example, we are looking to leveraging our new experience in ABM, and start to build basic building blocks. In the DoD industry, the search-detect-track-classify-identify-engage-kill assessment kill chain is so prevalent that we might consider building fundamental reusable building blocks that could be leveraged in various program. Our intent is to explore ways to further and enhance the use of ABM in SOS engineering, including the evolution of tools especially designed for SOS engineering trade-off analysis.

Conventional System of Systems (SOS)-level simulators are very complicated to construct, execute and maintain due to their complex nature. This issue is further exacerbated in situations where one needs to conduct Concept of Operations (CONOPS) evaluations and trade-off studies. Due to shear number of systems and their complex nature, users of SOS simulators traditionally tend to execute a large design matrix. This requires significant runtime requirement. One could leverage ABM's flexibility and ease of use to address this issue. For example, we can use ABM as a "light-weight" simulator that quickly identifies the design options which produces "main effects" within the run matrix. Along with the Taguchi Method for robust design [23], one can significantly reduce the simulation runtime, and still produce meaningful trend analysis results.

Another area that warrants research is the integration of ABM within the traditional simulation environment. One of our goals is to potentially provide a more flexible behavioral representation of command and control, planning, and tactics that can be used to help reduce the manning requirements for manned exercise and distributed

mission training (DMT) by providing better automated, computer generated forces, to provide the basis for M&S based decision aids that can be easily reconfigured to meet the needs of today, and to provide a more robust modeling of behaviors within the battlespace, especially for non-traditional threat behaviors, and Blue force responses. The main drawback, in our view, of current agent-based simulation systems is the requirement to embed the model of the physical world within the agent models. This requirement makes the development of flexible agent-based behavioral models unnecessarily complicated, especially if sophisticated models are needed to address the physical interactions that occur in combat. In essence, models of the physical world and the actors within it encapsulated in standard DoD simulations must be either extracted or recoded to fit within the agent modeling system. One possible solution is to leverage a system framework such as System for Parallel Agent Discrete Event Simulation (SPADES) [24]. SPADES is a middleware system that provides the capability to link together traditional agent systems such as COUGAAR or EMAA or any FIPA-compliant agent system, or composite agent modeling systems such as SWARM or ASCAPE, with traditional discrete event combat simulations such as the Naval Simulation System (NSS), the Extended Air Defense Simulation (EADSIM), the Joint Integrated Mission Model (JIMM), BRAWLER, JMASS, or the Air Warfare Simulation (AWSIM). SPADES provides a library and communications scheme to connect these models and simulations together, and to provide the modeling constructs within the framework such as thinking time modeling and out-of-order event processing. This library is linked into both the world model and the agent models to implement the distributed simulation functionality. In essence, SPADES allows the modeler to concentrate on the development of the C2/C4ISR behavioral model within the agent modeling frameworks, and to take advantage of existing modeling infrastructure where appropriate

References

[1] Joint Chiefs of Staff. *Joint Vision 2010*, Government Printing Office, Washington D.C., 1996.
[2] Joint Chiefs of Staff. *Joint Vision 2020*, Government Printing Office, Washington D.C., 2000.
[3] J. Boardman and B. Sauser. *System of Systems - the meaning of "of"*, Proceedings of the IEEE International Conference on System of Systems Engineering, Los Angeles, CA, April 2006.
[4] D. DeLaurentis and R. Callaway. *A System-of-Systems Perspective for Public Policy Decisions.* Review of Policy Research 21(6): pp. 829-837, 2004.
[5] M. W. Maier, *Architecting Principles for Systems-of-Systems*. Proceedings of the 6th Annual INCOSE Symposium. 1996.
[6] Correa, Y. and C. Keating. *An Approach To Model Formulation For Systems of Systems*. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 2003.
[7] Holland, J. H. *Hidden Order: How Adaptation Builds Complexity*. New York, Basic Books, 1995.

[8] Lewe, J.-H., B.-H. Ahn, et al. *An Integrated Decision-Making Method to Identify Design Requirements through Agent-Based Simulation for Personal Air Vehicle System*. AIAA's Aircraft Technology, Integration, and Operations Conference (ATIO), Los Angeles, American Institute of Aeronautics and Astronautics, 2002.
[9] C.M. Macal, M.J. North**,** *Tutorial on Agent-Based Modeling and Simulation*. Proceedings of the 2005 Winter Simulation Conference, 2005.
[10] D. A. Samuelson, C. M. Macal. *Agent-Based Simulation Comes of Age*. OR/MS Today, August 2006. http://www.lionhrtpub.com/orms/orms-8-06/agent.html (accessed 5-Feb-2007).
[11] J.R. Ellis. Objectifying Real-Time Systems, SIGS Books, Inc., 1994.
[12] I. Jacobson. *Object-Oriented Software Engineering*, Addison-Wesley Publishing Company, 1992.
[13] I. Jacobson. *Unified Software Development Process*, Addison Wesley Publishing Company, 1999.
[14] Rational University, *Methodology, Process, Education and Training for Today's Software Development Practices*. Part #810-010024-002, Rational University Press, 1998.
[15] Rational University, Methodology, Process, Education and Training for Today's Software Development Practices. Part #800-010017-002, Rational University Press, 1998.
[16] Defense Systems Management College (DSMC), Advanced Systems Planning, Research, Development and Engineering Course (SYS 301), DSMC, 1997.
[17] J.A. Dewar, S. C. Bankes, S. J. A Edwards. *Expandability of the 21st Century Army*. 1998. http://www.rand.org/pubs/monograph_reports/MR1190/ (accessed 5-Feb-2007).
[18] M.J. Liebhaber et al. *Studies of U.S. Navy Air Defense Threat Assessment: Cues, Information Order, and Impact of Conflicting Data*. Technical Report 1888, SPAWAR Systems Center San Diego, 2002.
[19] Y. Chen, B. Blyth. An Evidential Reasoning Approach to Composite Combat Identification (CCID). Proceedings of the IEEE Aerospace Conference, 2004.
[20] Brig. Gen. Rick Dinkins et al. *SIAP Integrated Architecture Development and Fielding* (presentation), INCOSE, South Maryland Chapter, April 30, 2004. http://www.incose-somd.org/ChapterMeetingMinutes/Presentations/SIAPCaptJefferyWilsonUSN.pdf, (accessed 31-Jan-2007).
[21] Young, B. W. *Future Integrated Fire Control*. 10th International Command and Control Research and Technology Symposium: The Future of C2, 2005.
[22] U. Wilensky et al. *NetLogo 3.1.2*. http://ccl.northwestern.edu/netlogo/ (accessed 5-Feb-2007).
[23] G. Taguchi. *The System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs.* Quality Resources, 1987.
[24] P. Riley, *SPADES: System for Parallel Agent Discrete Event Simulation*. http://spades-sim.sourceforge.net/ (accessed 5-Feb-2007).

