M&S JOURNAL

A DoD MODELING & SIMULATION COORDINATION OFFICE PUBLICATION | SPRING 2014

WHEN we decided on "Research" as an appropriate theme for this issue, the members of the Editorial Board asked if the focus was envisioned to be on Modeling and Simulation (M&S) employed to support research or on research used for the further development of M&S. After some pondering and talking with others in the community, the answer we came up with was "yes," in that both areas are worthy of attention. This decision is underscored by recent requests to the Modeling & Simulation Coordination Office (M&SCO) for assistance. These efforts included such issues as a request from the DoD Siting Clearinghouse (http://www.acq.osd.mil/dodsc/) to identify what models were available to simulate the effects of large wind farms on air traffic radars. Additionally, there is also the case of providing technical support to the DoD-DHS Capabilities Development Working Group, a senior level collaborative effort co-chaired by USD (AT&L).

As for examples of research in support of advancing M&S, three initiatives readily come to the forefront. First, there is the M&SCO-sponsored Cyber Working Group that seeks ways to integrate cyber effects in constructive simulations. Second is the critical participation in the NATO Study Group, which is creating a framework for the delivery of Modeling & Simulation as a Service (M&SaaS). The third example involves the activities of the Acquisition Modeling & Simulation Working Group (AMSWG), under USD (AT&L) ASD (R&E), which explore the use of M&S to aid in the material acquisition process. Given the range of endeavors, we crafted the "Call-for-Papers" to indicate these areas as topics of interest, and our contributors did not disappoint.

This issue starts with Mr. Ryan and Dr. Cummings' "Development of an Agent-Based Model for Aircraft Carrier Flight Deck Operations." The article provides an informative description of relevant M&S capabilities along with the challenges faced in attempting to validate agent-based models. From the U.S. Army Research Laboratory Simulation & Training Technology Center (STTC), Mr. Gaughan et al., provide an excellent description of a project that demonstrates the interplay between M&S and systems engineering. I had the privilege to participate in some of the design efforts described in their article, "Systems Engineering an Executable Architecture for M&S." Dr. Petty's article, "Modeling and Validation Challenges for Complex Systems," is noteworthy in that it begins as a tutorial and literature survey of complex systems, then goes on to describe ways in which models benefit the study of these systems. Dr. Petty is well-known for his contributions to the DoD M&S community, and he has a pragmatic way of communicating his ideas in a very readable style. The article by Dr. Mayberry et al., "Augmented Reality Training Application for C-130 Aircrew Training System," presents us with a practical application of how research of M&S capabilities can be utilized to support the improvement of training systems. The article describes a situation where the goal was to investigate an efficient and effective way to apply the use of augmented reality in a simulator that develops loadmaster skills in students before they train on the actual aircraft.

One aspect of working in a technology-intensive arena is the propensity to jump from one to another with little practical application. The contribution of Dr. Macedonia et al., "Cloud Simulation Infrastructure – Delivering Simulation from the Cloud," diverts from that behavior by presenting the practical aspects of developing an architecture for a Semi-Automated Forces (SAF) system delivered through a cloud computing environment. The article brings into account similar aspects of High Performance Computing (HPC) and gives the reader content that extends beyond more than just the usual buzzwords. Lastly is what I would characterize as a hybrid in that the contribution from Mr. Marrs and Dr. Heiges, "Soil Modeling for Mine Blast Simulation," begins with a detailed description of model development and validation, and then goes on to describe how that model was used in a simulation in order to better understand the effects of environmental factors on explosive mine blasts.

In closing, I am confident you will find this issue both interesting and, as always, educational. The Editorial Staff and the members of the Editorial Board take great pride in the *M&S Journal* and its contribution toward the continued advancement of M&S capabilities.

GARY W. ALLEN, PH.D.
Deputy Director
Instrumentation Training Analysis Computer Simulations and Support (ITACSS)
Joint Multinational Readiness Center, Hohenfels, Germany

Table of Contents

Guest Editorial: Modeling & Simulation in Research

According to the 2006 report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering, advances in mathematical modeling, computational algorithms, computer processing speeds, and the science and technology of data-intensive computing have prepared the way for unprecedented improvements in the health, security, productivity, and competitiveness of our nation. Additionally, and of equal significance, is the dramatic impact that effective simulation and modeling applications can have on the integrated costs associated with the development of new defense capabilities, as well as the commercialization of new technologies, thereby greatly enhancing the readiness, productivity, and competitiveness of our nation.

Modeling and simulation (M&S) draws from science, mathematics, computational methods, and engineering knowledge and methodologies. M&S represents an extension of theoretical science, and provides a powerful alternative to experimental science and observation. Driven by advancements in electronic and computational systems, M&S has evolved in complexity and sophistication to become an engineering discipline of its own. Computational capabilities are no longer simply limited to validating theories or helping to explain observed experimental results. Instead, simulations are now leading to new fundamental discoveries and extending our understanding of how natural and human-generated systems operate. This provides some motivation for how simulation is being used in research:

- For problems with no closed form solution, simulations allow human reasoning and mathematical analysis to complement each other, creating a problem-solving synergy.
- Simulation can be used for experimentation, understanding why phenomena occur, and exploring what-if possibilities. It helps expose undesired phenomena and enables a deeper understanding of the cause and effect relationships in the system.
- For many dynamic processes, simulation provides the only viable means to obtain direct and detailed observations within specified time limits. A simulation can accomplish in minutes what might require years of actual experimentation.
- In some cases, simulation may be the only reasonable approach to understand systems that do not yet exist. It can be used to explore the feasibility of new concepts, or evaluate multiple aspects of a proposed design.
- Many real-world systems are so complex that it is impossible to consider all of the interactions taking place at any given moment. Simulation allows us to better understand these interactions and gain insight into what affects the overall system.
- Simulation serves as a platform for organizing large quantities of dynamically-changing data, generating and evaluating various scenarios, and understanding what questions need to be answered.

GUEST EDITOR
Dr. Robert T. McGrath
Senior Vice President at the Georgia Institute of Technology and Director of the Georgia Tech Research Institute (GTRI)
Atlanta, GA

- It provides a framework in which to assemble a description of a system and test the completeness of the description in relation to known or desired system behavior.

This issue of the *M&S Journal* focuses on research in M&S. The papers highlight research projects across the Research and Development (R&D) community advancing two major themes. The first is how M&S is used to advance basic research and discovery, ranging from materials to medicine and from energy to economics. The second theme is research that seeks to improve the capabilities or practice of M&S and expand the ability of models or simulations to accurately represent new phenomena and at ever-increasing execution speeds.

An interesting example of how M&S is being used to enable cutting-edge research is in atomic, molecular, and chemical processes. A team led by Dr. Pratul Agarwal at Oak Ridge National Laboratory (ORNL) is improving upon nature to manufacture less expensive biofuels, detergents, and a host of other products by testing thousands of combinations of enzymes, chemistries, reaction temperatures, and wavelength-activated photochemical switches, which together can boost enzymatic functionality by 3,000 percent [1]. These discoveries are enabled by utilization of Titan, the world's fastest supercomputer, capable of operating at 27 petaflops or 27 quadrillion calculations per second [2]! Similarly exciting is an M&S application being used to develop new treatments for dissolving blood clots [3] and for design of never-before imagined semiconductor materials with novel properties [4].

An example of M&S research from my home laboratory, the Georgia Tech Research Institute (GTRI), comes from our Advanced Concepts Laboratory. Researchers are using M&S to design state-of-the-art antennas, evaluate electromagnetic signals in a complex environment, conceive new composite materials, and develop new microwave ion traps for quantum computing research. This research relies on computational electromagnetic (CEM) simulators using workhorse computational techniques such as the finite difference, finite element, and moment methods. Measuring anisotropic electromagnetic materials at low frequencies has led us to develop a new CEM-based method for inverting material properties. This technique is only possible now because fast computational algorithms are being combined with ever-increasing speeds and memory of computational hardware.

These levels of computational horsepower integrated in innovative ways with advanced electronics have also allowed the military and defense R&D communities to develop simulation environments with unprecedented levels of sophistication. These M&S tools and environments allow for a very cost-effective design and manufacture of a vast variety of defense components and systems, as well as for evaluation of their assured performance in complex and contested theaters of operation.

At GTRI and at other University Affiliated Research Centers (UARCs) across the country, we are proud to serve as trusted agents, advisors, and key contributors—providing M&S tools and solutions applicable to a broad range of Department of Defense (DoD) interests and needs. We are also pleased to be assisting in developing the next generation of Defense M&S engineers and researchers by providing hands-on, real-world work experiences to hundreds of bright and promising young students, as advocated by the *Strategic Vision for DoD Modeling & Simulation* [5].

M&S is an indispensable tool for solving a limitless range of scientific and technological problems facing our country. The M&S tools and methodologies that we teach, and the innovations that these students will envision and realize in the years ahead, will continue to revolutionize the way defense and military engineering and science is conducted in the 21st century.

References

[1] P.K. Agarwal, C. Schultz, A. Kalivretenos, B. Ghosh, and S.E. Broedel, Jr., "Engineering a Hyper-catalytic Enzyme by Photoactivated Conformation Modulation," *Physical Chemistry Letters*, vol.3, pp. 1142-1146, Apr.,2012.

[2] S. Jones, "ORNL Supercomputer Named World's Most Powerful," Oak Ridge Leadership Computing Facility, Nov., 2012. Available: http://www.olcf.ornl.gov/2012/11/12/ornl-supercomputer-named-worlds-most-powerful/)

[3] L. Vu, "Clot Busting Simulations Test Potential Stroke Treatment," NERSC, Sept., 2013. Available: http://www.nersc.gov/news-publications/news/science-news/2013/clot-busting-simulations-test-potential-stroke-treatment/

[4] A. Zakutayev, X. Zhang, A. Nagaraja, L. Yu, S. Lany, T. Mason, D. Ginley, and A. Zunger, "Theoretical Prediction and Experimental Realization of New Stable Inorganic Materials Using the Inverse Design Approach," J. Am. Chem. Soc., 135 (27), pp. 10048–10054, May, 2013.

[5] A. Shaffer, et al., "Strategic Vision for DoD Modeling and Simulation," M&SCO, Aug., 2012. Available: http://www.msco.mil/strategicVision.html

Author's Biography

Dr. Robert T. McGrath

Dr. Robert T. McGrath serves as senior vice president at the Georgia Institute of Technology and director of the Georgia Tech Research Institute (GTRI), Atlanta, GA. With over 1,700 employees and annual R&D awards in excess of $300 million, GTRI is an applied research powerhouse providing an array of technology solutions for the U.S. Department of Defense, Intelligence Community, Department of Homeland Security, other federal and state agencies, and industry. Dr. McGrath served on the 2012/13 Defense Science Board Study on Technology and Innovation Enablers for Superiority in 2030.

Dr. McGrath previously served in senior leadership positions at Battelle Memorial Institute, The Ohio State University, Penn State University, and Sandia National Laboratories. While at Sandia, he served as division director for the Advanced Scientific Computing Initiative's (ASCI's) Applications Programs, overseeing development and application of advanced finite element, fluid flow, and Monte Carlo models. These M&S tools were employed for simulation and design of neutron generators within the U.S. nuclear stockpile, manufacture of next generation microelectronic components, and design of flexible cockpit displays. He was also appointed by the U.S. Department of Energy (DoE) to direct collaborations with Japan, Europe, and the former Soviet Union on plasma-materials interactions and engineering of components for magnetic fusion reactors.

While at Penn State, Dr. McGrath served as professor, associate vice president for research, director of strategic and interdisciplinary initiatives, and director of the Marine Corps Research University. Throughout his career, he has been a passionate leader for STEM education at the graduate, undergraduate, and K-12 levels. Dr. McGrath received a B.S. degree in engineering science, an M.A. in mathematics, and an M.S. in physics from Penn State. Additionally, he received a Ph.D. in nuclear science and engineering from the University of Michigan.

Development of an Agent–Based Model for Aircraft Carrier Flight Deck Operations

AUTHORS

Mr. Jason C. Ryan
Engineering Systems Division
Humans and Automation Lab
Massachusetts Institute of Technology
Cambridge, MA 02143
Jcryan13@mit.edu

Dr. Mary L. Cummings
Associate Professor, Department of Mechanical Engineering and Materials Science, and Director of the Duke Humans and Autonomy Lab
Durham, NC 27708

ABSTRACT

WE DISCUSS THE VALIDATION OF AN AGENT-BASED MODEL OF AIRCRAFT CARRIER FLIGHT DECK OPERATIONS. THIS MODEL IS DESIGNED TO EXPLORE THE EFFECTS OF INTRODUCING UNMANNED ROBOTIC SYSTEMS AND RELATED SAFETY PROTOCOLS INTO FLIGHT DECK OPERATIONS. VALIDATING THE SYSTEM HAS BEEN CHALLENGING, AS THERE IS LITTLE PUBLISHED INFORMATION ON FLIGHT DECK OPERATIONS. DATA WAS ASSEMBLED FROM A VARIETY OF SOURCES, WITH THE VALIDATION PROCESS FOCUSING ON THE SIMULATION'S ABILITY TO REPLICATE REAL-WORLD DATA AND ITS RESPONSE TO CHANGES IN INPUT PARAMETERS ALIGNED WITH OBSERVED DATA AND SUBJECT MATTER EXPERT EXPECTATIONS. THIS PAPER PRESENTS THE RESULTS OF THIS VALIDATION PROCESS AND DISCUSSES FEATURES OF THE SIMULATION THAT WILL BE ADDED IN THE FUTURE.

INTRODUCTION

Several domains, including military, commercial, and private sectors, are actively researching the integration of unmanned vehicle systems into human-dense environments [1]–[5]. In these domains, human workers and robotic systems work in close quarters, marking a significant departure from the current tactic of maintaining strict physical separation of men and machines. Historically, this separation has been due to safety concerns in that the robotic systems were incapable of observing the position of human workers and avoiding striking them while in motion. Additionally, human workers may not properly understand the behavior of the robotic system and be unable to form a correct mental model of its behavior.

As robotic systems are brought into these environments, understanding how to safely integrate them into work processes involving human crew is paramount. Typically, however, this integration process happens slowly through iterative real-world testing, making small changes and observing results over time. For large-scale complex systems, like the national airspace system or the national highway network, such testing is difficult and costly to perform. While changes to work processes and interactions between humans and robots may improve efficiency and safety in the environment, the testing process is potentially dangerous for those involved, given unproven processes and robotic technology. The use of agent-based simulation methods that replicate the movement, decision-making, and organization of agents in the world allows for an exploration of various design changes to the system without requiring these costly, and potentially dangerous, real-life tests, at least in the initial stages of testing and design.

Towards this end, we developed a model of aircraft carrier flight deck operations (the multi-agent safety and control simulation, or MASCS), which once validated, will allow for exploration of possible system configurations (both in terms of people and hardware). Of particular interest are changes to the environment stemming from the introduction of unmanned aerial vehicles (UAVs) into the flight deck. This includes not only different types of UAVs, but also potential changes to crew organization and communication strategies, including between crew and aircraft. The primary measures of interest in such studies are how these changes affect both the safety (in terms of accidents and "hazards," or near-accidents) and productivity (how quickly the flight deck launches aircraft) of the flight deck, as well as how these changes may interact with each other to create unforeseen effects. This paper discusses the construction of the simulation environment and the validation process used to examine its ability to replicate current operations, including the data and results of the validation process.

DEVELOPMENT OF THE SIMULATION ENVIRONMENT

The flight deck environment, often described as "organized chaos" [6] is actually governed by a set of specific rules and behaviors for crew members and pilots on the flight deck. There is an underlying structure and organization to the activity on the flight deck that lends itself to simulation development. Upwards of 100 crewmembers and anywhere from 12 to 50 aircraft are present at any given time, requiring coordination not only between the human crew and piloted aircraft to execute tasks, but also between crewmembers to coordinate activity across the deck. Crew are given specific tasks governed by rules defining where they should be at what times and how they should interact with other crew and vehicles.

One group of networked crew, termed "aircraft directors," for example, provides instructions to pilots on where to taxi aircraft. Aircraft taxi through one director's area before being handed off to a second director on the way to the aircraft's assigned destination. Once clear, the first director repeats the process with another aircraft. Directors each manage their own zones, but maintain awareness about what is in the upcoming schedule. The pilots being directed are required to follow the directions of aircraft directors; if the director is not visible, pilots are not allowed to taxi. Pilots remain under directors' control until the vehicle is either physically connected to a catapult or is parked. For takeoffs, the pilot is under the authority of "the shooter" that runs the catapult launch process; only once in the air is the pilot not under other deck authority.

During the launch operation, the primary goal is moving aircraft from their initial parked positions to one of four launch catapults on the flight deck (figure 1). Catapults are located in pairs at the fore and aft areas of deck, and adjacent catapults cannot launch aircraft at the same time. During operations, aircraft form small queues at the catapults. The first aircraft parks on the catapult, waiting to be attached to the launching mechanism, while a second parks behind the jet blast deflector, awaiting its turn while launches alternate between catapults. As soon as a space is available, directors send the next available aircraft to that destination.

If only concerned with modeling the rate of launching aircraft from the flight deck, this configuration lends itself to a discrete event simulation (DES), with catapults modeled as servers and aircraft as arrivals into queues at each catapult. However, such a model makes it difficult to model the organization of the flight deck and explore how crew communication and traffic routing policies affect operations. DES models also tend to abstract away the physical motion of entities in the world, replacing physical motion with generic models of arrival rates. In an environment where safety is predicated upon maintaining appropriate separation between vehicles and other objects, modeling physical motion is a necessity. A lack of modeling physical motion also makes it difficult to understand exactly how replacing manned aircraft with UAVs affect operations, since UAVs have different capabilities (in terms of physical motion and logic) and failure modes (including lag in communications and failure to recognize commands or complete tasks) that affect their ability to move and accomplish tasks on deck.

Given these constraints and objectives, an agent-based modeling and simulation approach was selected, with aircraft, crewmembers, and catapults defined as independent agents. Agent-based modeling is particularly useful in modeling the key areas of interest in this work – physical motion in the world and decision-making under logical rules [7]. Building a model of flight deck operations required defining agents, their behavior and interactions (in the form of tasks), and higher-level "supervisory" communication structures, each of which is discussed in the following sections. All of the implementation of the MASCS simulation is done in Java™, through the Eclipse Integrated Development Environment (IDE), and using the Golden T Game Engine (GTGE) to track and update timesteps in the model.

Agents

Human crew, manned aircraft (modeled as F-18s), catapults, the deck itself, and all other entities are created as independent agents within the simulation environment. The most significant agents in this round of testing are models of the aircraft directors (ADs), the piloted F-18s, and the launch catapults. These three groups are the main drivers of launch operations.

Aircraft directors are modeled to move at a human walking pace – set as a random variable with a mean of 3.5 mph, a standard deviation (SD) of 1 mph, and bounded to be between 3 and 4 mph. Piloted aircraft are simulated at similar speeds, as they taxi at this pace in reality. Aircraft director agents include an understanding of their connectivity to other ADs, with logical rules governing when ADs can send aircraft to other directors and how aircraft behave while being directed. Aircraft do not taxi unless they have an aircraft director assigned to them and that director is within the visible range of each aircraft. Aircraft were modeled to only be able to "see" directors within a specified field of view in front of each aircraft. Aircraft were also given rules on avoiding collisions with other vehicles using a set of distance and angle thresholds to detect possible collisions and stop the vehicle.

Catapults and other resources are also modeled as individual agents. For catapults, software structures track the number of aircraft in queue at each catapult and store information related to task execution (discussed in the next section) and state monitoring. "Fouling" of the catapult (when a crewmember or vehicle enters a restricted area) and the catapult's operating status (available, in launch, fouled, or blocked) are both tracked by such structures. For all agents in MASCS, any monitoring routines regarding the state of the agent are stored within the agent and marked by Boolean flags. Doing so ensures that any unnecessary monitoring routines (for agents that are not present) are not run during simulation execution.

Defining Tasks

Tasks describe what, when, and how aircraft and crew function in the world, including interactions between agents. When initializing a scenario in the MASCS environment,

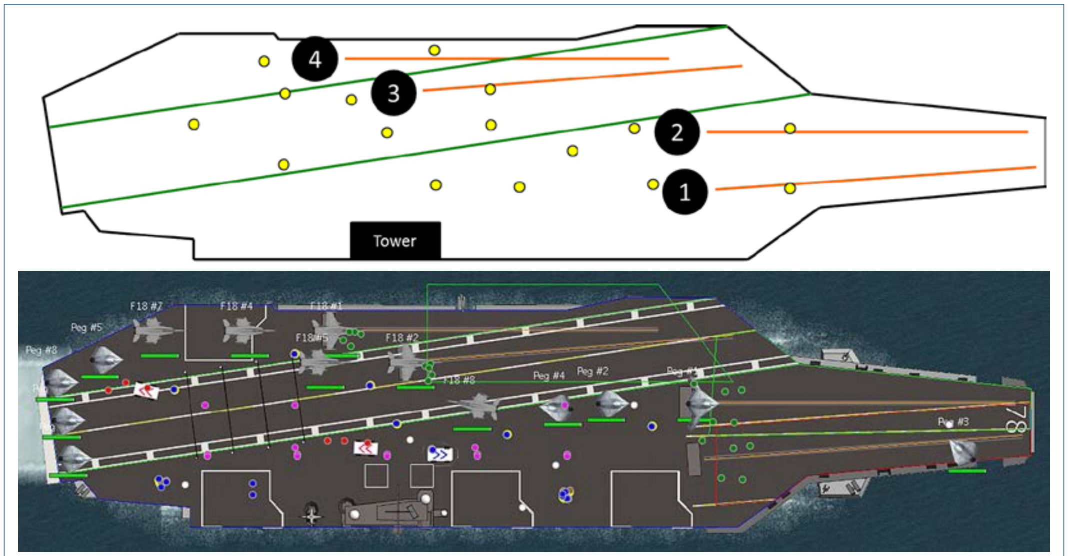


Figure 1: Top: Simplified representation of the flight deck. Catapults (orange lines) launch aircraft. Green lines indicate Landing Area. Parking areas begin at catapult 4 and move counter clockwise around the edge of the deck, stopping at catapult 1. Aircraft Directors appear as yellow dots. Bottom: Screenshot of the MASCS simulation animation.

a list of agents active in the world is input, along with the relevant goals for aircraft. This list of tasks is typically small, including taxi to launch, perform a mission, and return home. The details of the tasks required to reach these goal states are determined at run time based on current conditions in the system, including the number of crew, which catapults and other resources are operational, and the priority ordering of aircraft.

For an aircraft to launch, it must obtain a director and then execute a series of taxi tasks on the flight deck to navigate to a destination. If all adjoining directors are occupied with other vehicles, the aircraft waits for a director to become free. If the director is free but too far away, the aircraft waits until the director comes within view. Aircraft are passed from director to director within the network until they reach their destination catapult. Attachment to the catapult mechanism is modeled by an additional task, as is the act of physically accelerating down the catapult. Each of these tasks is defined as an independent random time distribution, stored within the task structure and randomly sampled during task creation. The random distributions defining time-based tasks such as these were based on interviews with experienced deck personnel, observations of flight deck operations, and video recordings of flight operations. When complete, tasks are removed from the task queue, and the next task is started. In some cases, events that occur during a task's execution require that a new task be substituted after it, or that the goal conditions change during execution.

Supervisory Logic Routines

A variety of "supervisory" logic routines mimic the functions of high-level supervisors in the environment, or replicate team situational awareness and coordination. In reality, the deck handler, a senior enlisted crewmember who is in charge of maintaining the efficiency of the deck, largely governs flight deck operations. The MASCS model replicates this in an aircraft scheduling assignment algorithm based on interviews with a variety of experienced personnel [8]. The assignment algorithm runs in real time; when a catapult becomes available, the algorithm attempts to assign the highest priority unassigned aircraft to the catapult, conditional on current traffic constraints on the deck. These constraints include conflicting traffic patterns and obstacles such as aircraft parked in the area between catapult 1 and the tower (figure 1).

Additional supervisory level routines address the coordination of traffic on the flight deck. In certain high-congestion zones, directors must delay the activities of their current aircraft in anticipation of the arrival of other higher-priority area that must taxi through the area. These are established as if-then logic checks based on a set of rules about taxi motion. During the development of the MASCS environment, identifying what conditions these high-level supervisory routines should address, how they should address them, and what conflicts between various subroutines existed and how to overcome them was challenging and one of the most time-consuming elements of the development process.

VALIDATING THE MASCS MODEL

The choice of validation methods depends largely on the end use of the simulation environment and what data is available to compare against [9]–[11]. Typically, the most stringent tests require large amounts of high quality, high fidelity data. Barring this, alternative methods such as reviews with subject matter experts (SMEs) and examining trends in simulation behavior are also acceptable. Agent-based models like the MASCS system pose an interesting challenge; however, as they replicate both individual behavior and interactions between individuals. Agent interactions are affected by individual agent behavior, requiring validation of both aspects. The validation process for MASCS aimed to address individual behavior first, followed by validations of mission-level scenarios. However, there is little quantitative data available on flight deck operations accessible to the public.

Validation data for single aircraft operations focused on interviews with SMEs who estimated certain parameters regarding flight deck operations, which were supplemented with recorded video footage, including that from a visit to the USS *Carl Vinson*. The only quantitative data obtained for mission operations came from two internal U.S. Navy reports from the Center for Naval Analyses (CNA) [12], [13]. The most important data from these reports center on the number of aircraft used in a single mission sortie and the interdeparture rate[1] of launches from the deck. The former provides guidance on the "typical" number

[1]An interdeparture time is the time between two successive launches (departures) from the flight deck, regardless of the catapult used. For instance, if launches happen from catapult 1 at t = 25 and from catapult 3 at t = 40, the interdeparture time is 15 seconds.

of aircraft used in sorties and helps define test scenarios. Interdeparture times are a measure of the performance of the aggregate system and are useful in determining how well the flight deck, as a whole, is performing. Moreover, this interdeparture data comes from a variety of mission profiles using various numbers of aircraft and catapults in both day and night operations, providing a more general description of flight deck behavior. The CNA reports also include some general "rules of thumb" concerning how often aircraft launch and land. While not explicitly noting values for mission duration (MD), the time from start of aircraft motion to launch, these measures are useful for developing estimates and ranges of performance.

Sensitivity analyses [11], [14] provide insight into another important aspect of a simulation model – robustness to any potential errors in parameter values, especially those whose values were not well-defined through empirical data. In sensitivity analysis testing, parameter values are systematically varied over a series of tests, examining the resulting changes in output metrics. If the variations in output are beyond what is expected, it suggests that errors exist within the simulation model. Sensitivity analyses of the MASCS model of flight deck operations varied two key input parameters (the number of aircraft and catapults used in scenarios) and three key parameters affecting subtasks on the flight deck (taxi speed, collision avoidance settings, and mean launch preparation time).

A final method of judging model validity involved an interview with SMEs. In these interviews, attendees were asked to evaluate both the quantitative results of the simulation as well as review live animations of the simulation as it processes. The validation of the MASCS simulation proceeded through these steps in the order presented above – single aircraft testing first, followed by mission validation tests reviewing launch interdepartures and sensitivities to parameter changes, followed by an SME review involving veterans of aircraft carrier flight deck operations. The details and results of each of these tests are presented in the following sections.

Single Aircraft Testing

Single aircraft testing relied on data extracted from a YouTube video [15] that followed a single aircraft from its initial parked location in the aft area of the flight deck through taxiing to and launching from catapult 3. This task breaks down into three distinct subtasks. First, a series of taxi tasks moves the aircraft from its initial position to the launch catapult. Second, launch preparation tasks (total duration randomly sampled from a normal distribution with a given mean and SD) at the catapult ready it for launch. Third, the aircraft physically accelerates down the catapult and launches. The goal of this testing phase is to demonstrate that the simulation can reasonably replicate this scenario, as it is representative of all parking-to-launch tasks for aircraft on the flight deck.

From the video, time values for each of the subtasks were extracted, including the total MD. A simulation scenario replicating the video was created for the MASCS environment. Scenarios in MASCS only require the initial conditions (the list of aircraft and their parking spaces) and the general tasks required (for instance, taxi, launch, and proceed to mission). The details of the tasks are not determined until execution in the simulation. This scenario was executed 30 times within the MASCS environment to provide sufficient exploration of the random variables within the simulation. These results were then compiled and compared to data extracted from the video observation (figure 2).

Because we cannot be sure as to where the observed task truly falls in the distribution of possible tasks in the environment, we desire that the empirical results be reasonably likely in the simulation, but not necessarily be equal to the simulation mean. As can be seen in the figure, results for each of the four measures are near to or less than 1 SD away (from left to right: 0.73, -0.45, 0.77, and 1.15 standard deviations, respectively), but also not within 0.5. This is an acceptable spread of results; the empirical results are not highly unlikely in the simulation, but they are also not very near the mean. Additionally, the direction of the differences in the MD values and launch preparation times (which increases MD) is as expected. Interviews with SMEs suggest that the observed launch preparation time (the biggest driver of MD) is far higher than the average,

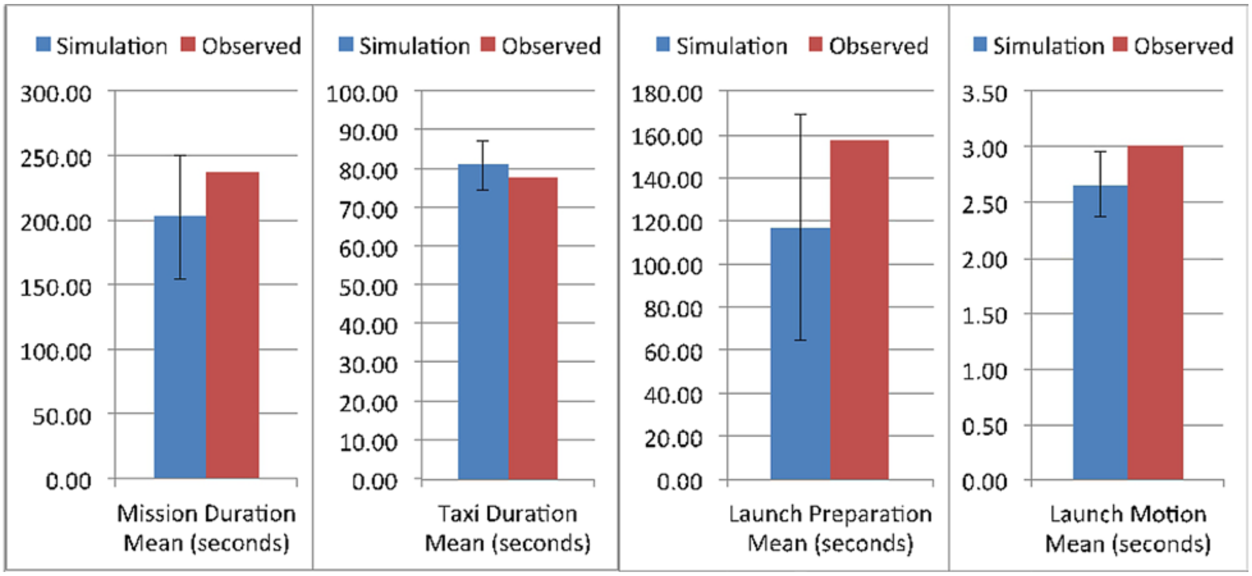


Figure 2: Results of single aircraft testing. Simulation results in blue, observations in red. Whiskers denote ±1 standard deviation from mean. Left to Right: Mission Duration, Taxi Duration, Launch Preparation, and Launch Motion times.

which ranges between 60 and 120 seconds for non-combat operations and between 45 and 75 seconds in combat. Given this, it was accepted that the MASCS simulation is a reasonable model of individual aircraft operations and that testing could be expanded to the mission level phase, discussed in the next section.

Mission Validation Testing

Mission validation focused on the three primary areas discussed earlier: launch interdeparture values, sensitivity analyses of both internal and input parameters, and SME interviews. Launch interdeparture measures depict the performance of the flight deck as a whole; correctly matching these indicates that the modeling of processes and the timing of subtasks are generally correct. Sensitivity analyses varied both input parameters (i.e., number of aircraft and catapults) and the three internal parameters related to task completion: the launch preparation distribution mean, mean taxi speed, and collision avoidance (CA) thresholds. The results of this testing appear in the following sections.

Launch Interdeparture Values

The aforementioned reports from the Center for Naval Analyses [12], [13] included both launch profiles used during operations (the number of times a launch of *N* aircraft was observed), as well as the cumulative density function (CDF) of launch interdeparture times on the flight deck. This data is fit by a negative exponential distribution of the form $\lambda e^{(-\lambda t)}$, with $\lambda=0.0177$; a Kolmogorov-Smirnov goodness of fit test returns a p-value of 0.07173, failing to reject the null hypothesis H_0: the distribution adequately captures the data at the $\alpha = 0.05$ level. This provides the baseline comparison point.

Simulation testing in this phase used numbers of aircraft ranging from 22 to 34 aircraft, based on their status as boundary values in the launch profiles described in the CNA data. For each aircraft setting, 30 replications were performed with both 3 and 4 catapults, described by SMEs as the most typical operating conditions. Launch interdeparture values were calculated for each mission replication and fit by negative exponential distributions. P-values for the Kolmogorov-Smirnov goodness of fit tests for these results ranged from 0.09987 (the 24-3 case) to 0.450 (the 22-3 case), all failing to reject the null hypothesis. Figure 3 contains a graph of the λ values from the interdeparture fits (triangles) and their 95% confidence intervals (vertical lines). The original CNA report data appears on the left of the figure in orange. The λ values for all fits fall within the confidence interval of all other fits, suggesting that launch interarrivals are not significantly different from one

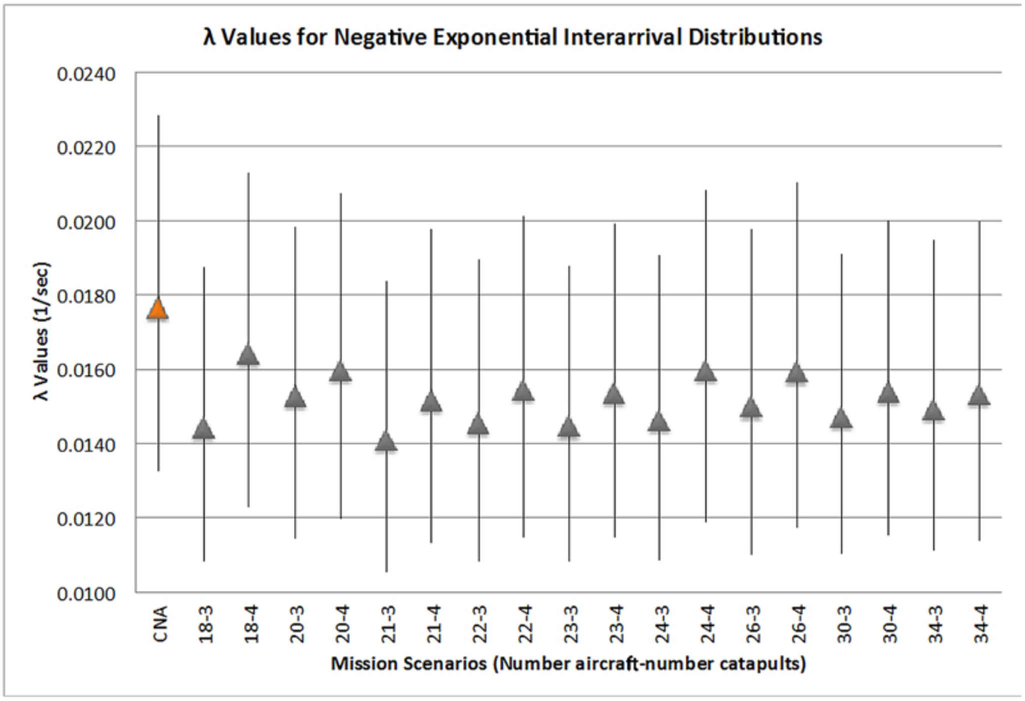


Figure 3: Results of negative exponential distribution fits to launch interdeparture values. λ values appear as triangles. Vertical lines denote 95% confidence interval on fits.

another. This signifies that the MASCS simulation correctly replicates the interarrival data from the CNA reports.

Sensitivity Analyses

The MASCS validation sensitivity analyses reviewed variations in the launch preparation time distribution, taxi speeds, collision avoidance rules (which govern how vehicles avoid collisions with other vehicles on deck), and the effects of adding aircraft or increasing the number of catapults available during a mission. There was no strict quantitative data on performance to compare with, but these tests instead review the changes in outputs and whether or not they are reasonable. Increases in taxi speed, as a rate measure, should decrease overall MD. Increasing the launch preparation time mean or collision avoidance parameters should result in an increase in MD. For launch preparation times, due to the nature of the queuing system on the flight deck, the change in MD values should be identical to the change in the parameter mean. For taxi and CA parameter changes, their effects on decreasing MD (making operations faster) are limited by the nature of the queues on the flight deck. Therefore, as long as queues are maintained at the catapults, performance is near optimal. At some point, additional changes to taxi speed or CA parameters only result in aircraft waiting longer at the catapult. In the other direction, slowing the arrival of aircraft at catapults may have significant effects on operations if it disrupts the queuing process.

Even though there are four catapults, each pair (forward and aft) works together as a single large server. The forward and aft pairs can process in parallel and should launch roughly the same number of aircraft. These launches then sum linearly to produce the MD. This MD value should also be nearly identical to the average interdeparture time multiplied by the total number of aircraft launched. Thus, increasing the number of aircraft in a scenario should lead to a linear increase in MD with a slope equal to the average interdeparture time. The addition of another catapult only changes how the allocation of tasks occurs; adding a fourth catapult means that the deck can clear launches in one less "round" of allocations. The difference between three and four catapults, then, should be between one to two launches. As with variations in other parameters, the goal of this test is to demonstrate that the effects of these changes on simulation output are not extreme.

Figure 4 depicts both mission duration versus number of aircraft for the three and four catapult scenarios, as well as the linear regression fits for the lines. Both scenarios return linear coefficients close to the mean launch interdeparture time value (66.50 seconds, or 1.1083 minutes). Statistical tests of the regression fits reveal a good overall model fit for both the three ($F(1,7) = 2436.58$, $p<0.0001$) and four ($F(1,7) = 520.98$, $p<0.0001$) catapult cases. Tests also showed significance for the regression coefficients for both cases ($t = 3.01$, $p < 0.0001$ and $t=-0.79$, $p<0.0001$, respectively). Differences between missions averaged 101.34 seconds across cases, just below the mean launch preparation time of 109.65 seconds. Together, these results indicate that the responses of the simulation to changes in number of aircraft and catapults is as expected and that the simulation is not behaving in an unexpected fashion.

Further testing varied internal parameters within MASCS, changing the launch preparation time mean by ±10% and the taxi and CA parameters by ±25% each. Only tests at the 22 and 34 aircraft cases (the largest and smallest mission sizes) were tested. The results of these tests appear in figure 5, with red bars indicating tests where parameter values were increased and blue bars where values were decreased. The figure is formatted as a "tornado chart," with ranges of variation ordered highest to lowest. In these tests, the most significant variations in performance come from launch preparation times, with a range of variation on the order of 15% (7.5% in each direction). It was noted above that the changes in MD value due to launch preparation time mean were expected to be identical to the parameter variation – about 10%, not the observed 7.5%. Further testing revealed that the random samples generated in the ±10% conditions were actually closer to ±8% due to the effects of a minimum cap on sample values. Once accounting for this, the simulation results were within 30 seconds of the expected values of the tests. This error is well within the s.d. of each mission, and the results are considered to validate that the MASCS simulation responds appropriately to changes in launch preparation mean.

For taxi and CA changes, responses were relatively small (ranges of variation less than 7%) and often not statistically significant. ANOVA tests (H_0: all groups are random samples of the same population) revealed significant differences at the $\alpha = 0.05$ level only for CA changes at the four catapult cases ($p=0.0124$ and $p=0.0056$ at 22 and 34 aircraft, respectively) and for taxi speed changes at the 34 aircraft scenario ($p=0.0090$ and $p=0.0261$ for three and four catapults, respectively). Running simultaneous t-tests on the sets demonstrated that, for CA parameter changes, runs with decreased CA parameters (more freedom of

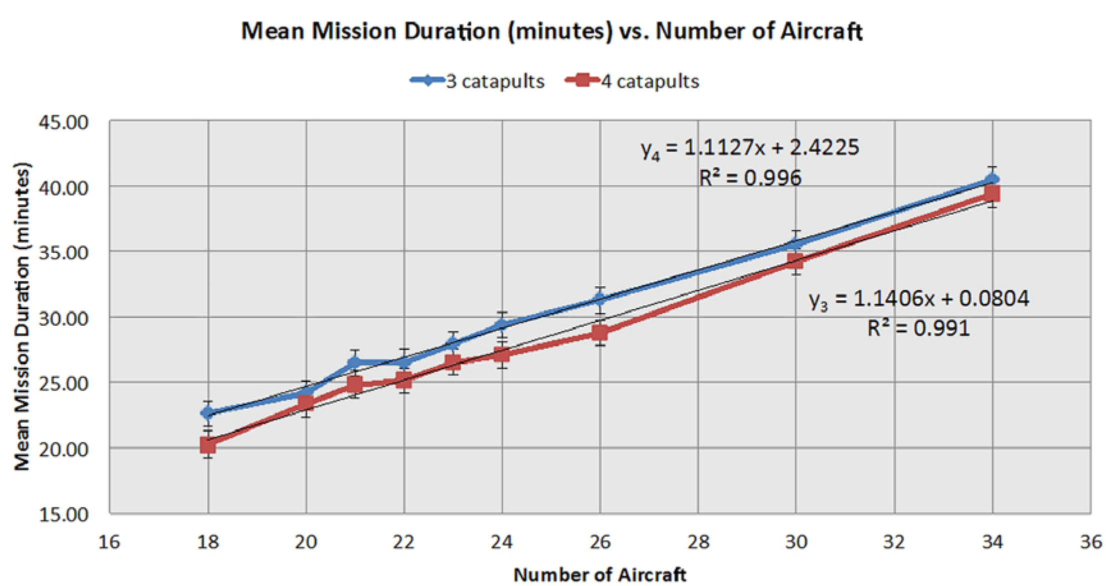


Figure 4: Mission Duration versus Number of Aircraft in mission, by number of catapults used (red, blue). Whiskers indicate ± 1 standard deviation.

motion on the deck) were significantly different from the others; runs increasing the parameters (restricting motion on deck) were not different from the baseline. For taxi speed changes for 34 aircraft, simultaneous t-tests revealed that the increase and decrease levels were significantly different from one another, but not from the baseline. These results, however, are consistent with conditions on the flight deck. Four catapult cases create more congestion on deck than three catapult cases, and it is reasonable that CA changes increasing freedom of motion would have the most effect here. Likewise, taxi changes only had affect for the larger of the two mission sizes, which is sensible if it only introduces slight performance benefits that accrue over time. Overall, the results of the sensitivity analysis suggest that the MASCS system responds as expected to changes in both input and internal parameters; no significant deviations from expectations were observed.

Subject Matter Expert Interviews

After concluding the sensitivity analyses, the simulation environment and test results were presented to a set of SMEs for review. SMEs attending the meeting had a wide range of experience on the flight deck, each over multiple years (e.g., two pilots each with over 800 flight hours in fighter aircraft; another individual had over 3,000 hours as a Naval Flight Officer and 2 years as a launch and recovery officer). All were employed by Navy research organizations at the time of the meeting. During the meeting, the simulation (which displays a live animation of the evolution of the flight deck environment) was presented first to SMEs for their critiques. This continued until participants had no remaining comments, at which time the discussion shifted to discussing the results of validation testing.

The SMEs agreed that, qualitatively, the animations of the simulation were accurate overall. Several minor errors were noted; for instance, certain taxi actions and alignments, like the location where the catapult turned to enter catapult 3, were not quite correct. They also felt that there were minor problems in the scheduling program in terms of what areas of the flight deck were cleared first (aft-most aircraft should be prioritized, whereas the simulation had moved a few aircraft adjacent to this area ahead in the order). The SMEs also noted an inconsistency in the initial conditions of the simulation. Previous interviews had not noted that mission operations begin with the catapults "stacked" – one aircraft parked at each catapult, preparing to launch, with another queued behind and others in transit. Under these conditions, the mission duration starts counting as soon as the first aircraft launches. This results in MD values higher

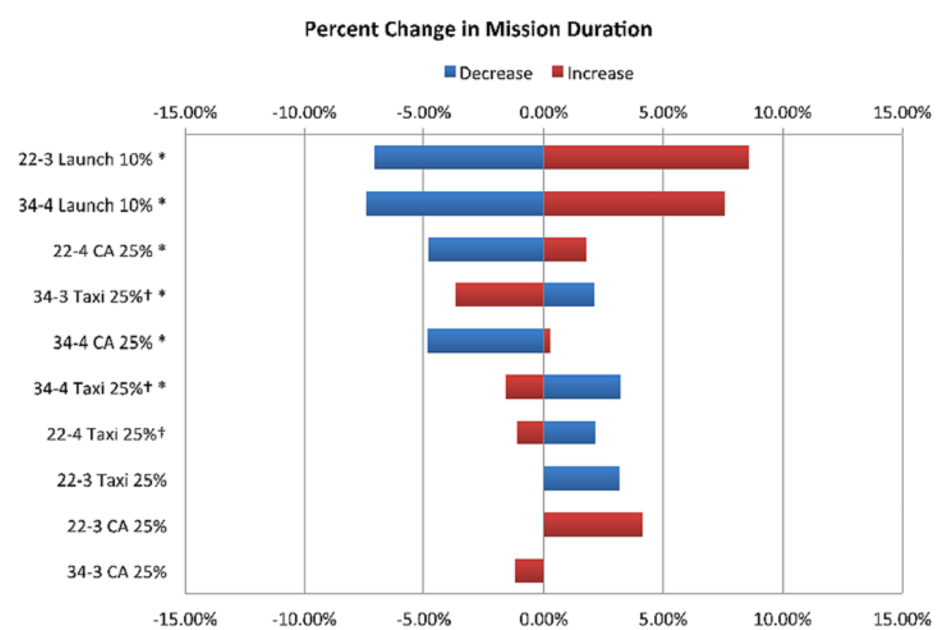


Figure 5: Sensitivity Analysis "Tornado" Chart. Daggers (†) indicate an inverse response to changes in parameter values is expected. Asterisks (*) indicate significant differences in ANOVA tests.

than what SMEs would expect, but this can be corrected with minor corrections in the simulation.

The SMEs disagreed with the mission duration values but were not overly concerned with these given observations of the rest of the simulation. SMEs admitted that they were largely biased to thinking about combat operations, which occurs at a higher tempo due to faster launch preparations. MASCS, for these tests, was set to reflect "typical," slower, non-combat operations. Additionally, SMEs also noted that some of the discrepancy would be corrected if the catapults were properly "stacked" before the first launches.

In terms of sensitivities, SMEs explained that they had relatively poor mental models of these effects. Aircraft only operate at one taxi speed; aircraft directors, in navigating aircraft, are typically very risky. Neither of these varies substantially amongst directors, ships, or mission conditions. Additionally, the selection of the number of catapults is not viewed as an option; what is available is used, and extra catapults are thought to help clear more space on deck. Little thought is given to the effects of adding more aircraft as well, as this number is not really an option for the deck crew. Even so, conceptually, the SMEs did not disagree with the results and approved of the simulation as a valid replication of flight deck operations, given the minor corrections discussed above.

DISCUSSION AND FUTURE WORK

This paper discussed the process of building and validating an agent-based model of aircraft carrier flight deck operations. The primary challenges in this process have been the acquisition of data on operations, in terms of both quantitative performance and organizational structure, and translating it into either usable validation data or correctly modeling behavior in simulation. To overcome this, information regarding operations was compiled from a variety of sources. Where significant validation data was lacking, sensitivity analyses examined the response of the simulation to changes that were then judged for their accuracy by SMEs with an understanding of the mechanics of the real flight deck. While, ideally, any further work would involve the acquisition of additional performance data, this process provided one way of analyzing the system without such data.

The sensitivity analyses presented in this paper suggest that the main drivers of performance, in terms of mean mission duration, lay not in the number of aircraft, the number of catapults, or necessarily the speed and freedom of motion, but rather with the launch process itself. However, the effects of movement speed and freedom of motion (due to collision avoidance parameters) are negligible only up to the point where they prevent queues from being maintained at catapults. Given the Navy's goal to put UAVs on carriers, it is unclear how UAVs will perform in this current environment, or what changes will be needed to incorporate their inclusion. In view of the sensitivity of the launch time preparation parameter, our results suggest that this should be a primary concern for UAV developers in terms of aircraft carrier integration.

In general, the results also suggest that important features of deck operations are not yet captured by SME heuristics. While extra catapults are always used in order to clear space, this comes with an associated cost of more congestion on the flight deck and presumably would cause significant changes in the safety metrics that will be examined in future work. Results here demonstrate that it may be more beneficial, and perhaps safer, to commit resources to speeding the launch process rather than to include an additional catapult. Future work within MASCS will further examine these interesting phenomenon in deck operations, as well as explore the effects of integrating unmanned vehicles of various types (air and ground), ranging from teleoperated to gesture-controlled to fully autonomous systems. The validation of the MASCS model of current operations presented here lends validity to these future implementations, in that the methods of modeling human and vehicle motion, communication, and coordination are effective in replicating real-world results and can be effective for modeling future complex systems.

Acknowledgments

This research was supported by the Office of Naval Research Science of Autonomy program, Contract #N000140910625.

The opinions expressed in this paper are those of the authors and do not necessarily reflect the opinions of the sponsoring agency.

References

[1] Office of the Secretary of Defense, "Unmanned Systems Roadmap 2011-2036," Office of the Secretary of Defense, Washington, DC, 2011.

[2] A. Topf, "Rio Tinto boosts driverless truck fleet for use in Pilbara," 2011. Available: http://www.mining.com/2011/11/02/rio-tinto-boosts-driverless-truck-fleet-for-use-in-pilbara/.

[3] Rio Tinto, "Mine of the Future," 2011. Available: http://www.riotinto.com/ironore/mine-of-the-future-9603.aspx.

[4] R. J. Rosen, "Google's Self-Driving Cars: 300,000 Miles Logged, Not a Single Accident Under Computer Control," 2013. Available: www.theatlantic.com/technology/archive/2012/08/googles-self-driving-cars-300-000-miles-logged-not-a-single-accident-under-computer-control/260926/

[5] D. DeBolt, "Google's self-driving car in five-car crash," 2011. Available: http://www.mv-voice.com/news/2011/08/08/googles-self-driving-car-in-five-car-crash

[6] G. I. Rochlin, T. R. La Porte, and K. H. Roberts, "The Self-Designing High-Reliability Organization: Aircraft Carrier Flight Operations at Sea," *Naval War College Review*, no. 42, pp. 76–90, 1987.

[7] E. Bonabeau, "Agent-based Modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. Suppl. 3, pp. 7280–7287, 2002.

[8] J. C. Ryan, A. G. Banerjee, M. L. Cummings, and N. Roy, "Comparing the Performance of Expert User Heuristics and an Integer Linear Program in Aircraft Carrier Deck Operations," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2013.

[9] A. M. Law, *Simulation Modeling and Analysis*, 4th ed. Boston: McGraw-Hill, 2007.

[10] O. Balci, "Verification, Validation and Accreditation of Simulation Models," in *Winter Simulation Conference,* 2003.

[11] R. G. Sargent, "Verification and Validation of Simulation Models," in *Winter Simulation Conference*, 2009.

[12] A. Jewell, "Sortie Generation Capacity of Embarked Airwings," Center for Naval Analyses, 1998.

[13] A. Jewell, M. A. Wigge, C. M. K. Gagnon, L. A. Lynn, K. M. Kirk, K. K. Berg, T. A. Roberts, A. J. Hale, W. L. Jones, A. M. Matheny, J. O. Hall, and B. H. Measell, "USS Nimitz and Carrier Airwing Nine Surge Demonstration," Center for Naval Analyses, Alexandria, VA, 1998.

[14] J. D. Sterman, *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York: Irwin/McGraw-Hill, 2000.

[15] Lockergnome, "USS Nimitz Navy Aircraft Carrier Operations," *Youtube.com*, 2009. Available: http://www.youtube.com/watch?v=_2KYU97y2L8

Authors' Biographies

Mr. Jason C. Ryan

Mr. Jason C. Ryan received a B.S. degree in aerospace engineering at the University of Alabama and an M.S. degree in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT). He is currently a third-year Ph.D. candidate in the Engineering Systems Division at MIT. His dissertation work focuses on building agent-based models of human-unmanned vehicles systems, with an emphasis on examining the tradeoffs between safety and mission effectiveness for different implementations of unmanned vehicles. His other research interests include human supervisory control, human-unmanned vehicle interaction, and collaborative human-computer decision making.

Dr. Mary L. Cummings

Dr. Mary L. Cummings received a B.S. degree in mathematics from the United States Naval Academy, an M.S. degree in space systems engineering from the Naval Postgraduate School, and a Ph.D. degree in systems engineering from the University of Virginia. A naval officer and military pilot from 1988 to 1999, she was one of the Navy's first female fighter pilots. She is currently an associate professor of Mechanical Engineering and Materials Science and director of the Duke Humans and Autonomy Lab. Her research interests include human supervisory control, human-unmanned vehicle interaction, collaborative human-computer decision making, decision support, human performance modeling, and the ethical and social impact of technology.

Systems Engineering an Executable Architecture for M&S

Authors

Mr. Scott Gallant
Effective Applications Corporation
Orlando, FL 32826
Scott@EffectiveApplications.com

Mr. Christopher J. Metevier, Mr. Chris Gaughan
Army Research Laboratory's
Simulation and Training Technology Center
Orlando, FL 32826
Chris.Metevier@us.army.mil
Chris.Gaughan@us.army.mil

Keywords

Systems Engineering, Distributed Modeling & Simulation, Executable Architecture

Abstract

DISTRIBUTED MODELING AND SIMULATION (M&S) IS FUNDAMENTALLY BASED ON THE EXCHANGE OF INFORMATION BETWEEN FUNCTIONS THAT MAY NOT HAVE BEEN BUILT TO WORK TOGETHER. MODELS ARE USUALLY SEPARATELY MANAGED WITH VARYING BUDGETS AND OFTEN WITH DISPARATE PURPOSES. THE LIFE-CYCLE OF AN M&S EVENT IS LONG DUE TO THE COMPLEXITY OF THE SYSTEMS ENGINEERING REQUIRED TO DESIGN, IMPLEMENT AND DEPLOY A COHESIVE SET OF SYSTEMS TOWARDS THE EVENT'S OBJECTIVES.

OUR RECENT RESEARCH HAS FOCUSED ON DEVELOPING A TOOL TO FACILITATE THE SYSTEMS ENGINEERING PHASE TO ENABLE MORE ACCURACY AND AUTOMATION WITHIN THE IMPLEMENTED EVENT. WE HAVE SUCCESSFULLY CAPTURED THE TECHNICAL SPECIFICATION FROM REQUIREMENTS THROUGH DESIGN TO EXECUTION INFORMATION (INCLUDING CONFIGURATION) IN A DATABASE-DRIVEN AND LINKED MANNER.

THIS PAPER WILL DESCRIBE OUR SOLUTIONS FOR SYSTEMS ENGINEERING AND AUTOMATING A DISTRIBUTED M&S IMPLEMENTATION. WE'LL DEMONSTRATE HOW WE CAPTURE HIGH LEVEL SYSTEM REQUIREMENTS AND THEIR LINKAGE TO LOW LEVEL MODEL SPECIFICATIONS. WE'LL SHOW HOW WE CAPTURE METADATA ABOUT THE MODELS, SCENARIOS AND EXECUTION ENVIRONMENT AND ULTIMATELY HOW WE DEPLOY AND EXECUTE THE SPECIFIED MODELS USING VIRTUAL MACHINES. OUR SYSTEM INTERFACE INCLUDES AN ELECTRONIC INTERVIEW PROCESS THAT DETERMINES WHICH OF THE MANY POSSIBLE IMPLEMENTATION CHOICES (MODELS, SCENARIOS AND SYSTEM DESIGNS) TO USE FROM THE USERS' REQUIREMENTS. BASED ON THE STRATEGY WE USE FOR CAPTURING THE SYSTEM DESIGN AND A GOVERNMENT-OWNED SET OF TOOLS, WE CAN ALSO CREATE AND RAPIDLY GENERATE SURROGATE APPLICATIONS TO SUBSTITUTE FOR LATE, FAULTY OR UNAVAILABLE MODELS.

WE'LL DESCRIBE HOW THESE CAPABILITIES COME TOGETHER WITHIN OUR INITIATIVE, THE EXECUTABLE ARCHITECTURE SYSTEMS ENGINEERING (EASE) FOR M&S THRUST. WE'LL ALSO MENTION HOW THE COMMUNITY COULD BENEFIT FROM THESE METHODOLOGIES AND OUR FUTURE RESEARCH AREAS.

BACKGROUND AND INTRODUCTION

Simulationists who require the use of distributed simulation typically do not have a long life cycle for an experiment, analysis initiative or simulation-based event. To reduce cost, they need to use a well-established simulation architecture and robust models that are easy to integrate with other distributed simulations. This short lead time for system design, development, integration and execution forces the system definition and design to happen very quickly.

These modeling and simulation (M&S) users rely on standards and simulation developers to get the systems to communicate using the same syntax. This often works to instantiate a System of Systems (SoS) architecture [1] and get models to share information. A SoS environment is an assembly of applications that together provide more capability than the sum of their individual capabilities. Within the M&S community, the applications assembled are each focused on representing a specific warfare function based on data and models from an organization considered to be the center of excellence for that function. The SoS architecture provides many benefits when compared to executing a single monolithic model including performance, model management and information transparency for analysis.

However, the biggest problem in these cases is that the models do not work together semantically for the accomplishment of the high level functions that the users require. In other words, applications may not be communicating based on a consistent understanding of the context and connotation of the information being shared. We have developed a tool that ensures semantic interoperability traced back to functional requirements. We have learned many lessons in our work and see a vision for the future of systems engineering for SoS architectures.

We have established a systems engineering data-driven infrastructure that allows SoS design encapsulation and connected an interview system that allows a user to launch a distributed M&S execution based on functional and scenario choices. We have implemented generative programming techniques [2] (automatically generating executable computer programming artifacts from a higher level source) in order to quickly deploy a SoS architecture for military analysis. The flexibility required to implement our goal requires systems architecture qualities and objectives such as encapsulation of functionality into appropriately sized portions to be able to manipulate and construct larger capabilities as needed with as little engineering effort as possible. We aim towards an architecture that is fully compliant with U.S. Army-grade verification and validation guidance [3] and robust enough for decision-oriented analysis while maintaining flexibility and quickness in order to save the Army tremendous amounts of time and effort [4] when constructing distributed M&S environments for various uses.

SYSTEMS ENGINEERING APPROACH

There are countless configurations of M&S that can achieve a user's functional requirements. In order to allow the federation designer to pull together only those functional components required to build a simulation environment, we implemented a philosophy of capturing detailed information about each of the applications under consideration. Not all models are necessary for all types of scenarios or all variations of simulation use cases (analysis, training, etc.). The systems engineering infrastructure is meant to be fully traceable from high level requirements through the design and ultimately to low level specifications. By maintaining the systems engineering information in a composable and linked fashion, the event designer can start by selecting high level functions that need to be incorporated into the event. The system then easily produces all the applications that can meet those needs, the object model elements important for information exchange between those applications and some automated generation of event-specific systems engineering artifacts, such as the system requirements specification (SRS) or the publish/subscribe matrix. The system can even generate executable test cases and surrogates. Generating these products from the design database ensures more accurate products and saves time by reducing the systems engineering effort of managing this type of information in spreadsheets or documents while trying to maintain configuration management.

Systems Engineering Structure

We utilize a System Design Description (SDD) to capture the system design at a functional level and subsequently link the functional design to the technical design. This allows the functional requirements to be linked to system design and allocated to specific models as shown in figure 1.

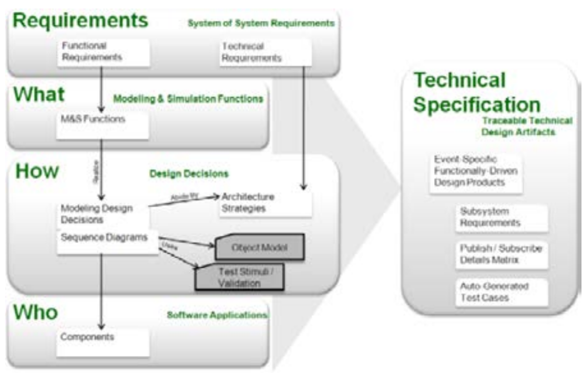


Figure 1: Database driven systems engineering infrastructure to link SoS information

The low level requirements, object model and test cases can then be auto-generated based on model allocation to functions. The SDD is data-driven, easing information maintenance duties by linking the system engineering products and simplifying the editing of the system design. The SDD includes the mapping between the data to be collected during the exercise, the initial exercise goals and the semantic explanation of what the data means. There are two types of requirements: functional and non-functional.

Functional requirements dictate the military representation necessary within the simulation environment, such as, "Represent communications effects on messages sent between all entities." Non-functional requirements are more technically or politically oriented and ultimately drive how the simulation environment should be built, executed or used, such as, "Execute within a classified environment." These non-functional requirements are assigned to architectural strategies which can then be enforced across design decisions. Functional requirements are linked to an explanation of the functionality in "M&S functions." The M&S functions are then linked to design decisions which describe how low level functions need to exchange information and work together to realize the high level M&S functions. Functional design can then be indirectly assigned to the technical design by allocating models and/or simulations to the low level functions and allocating elements within the object model to the information exchange details.

Figure 2 shows an example of a functional sequence

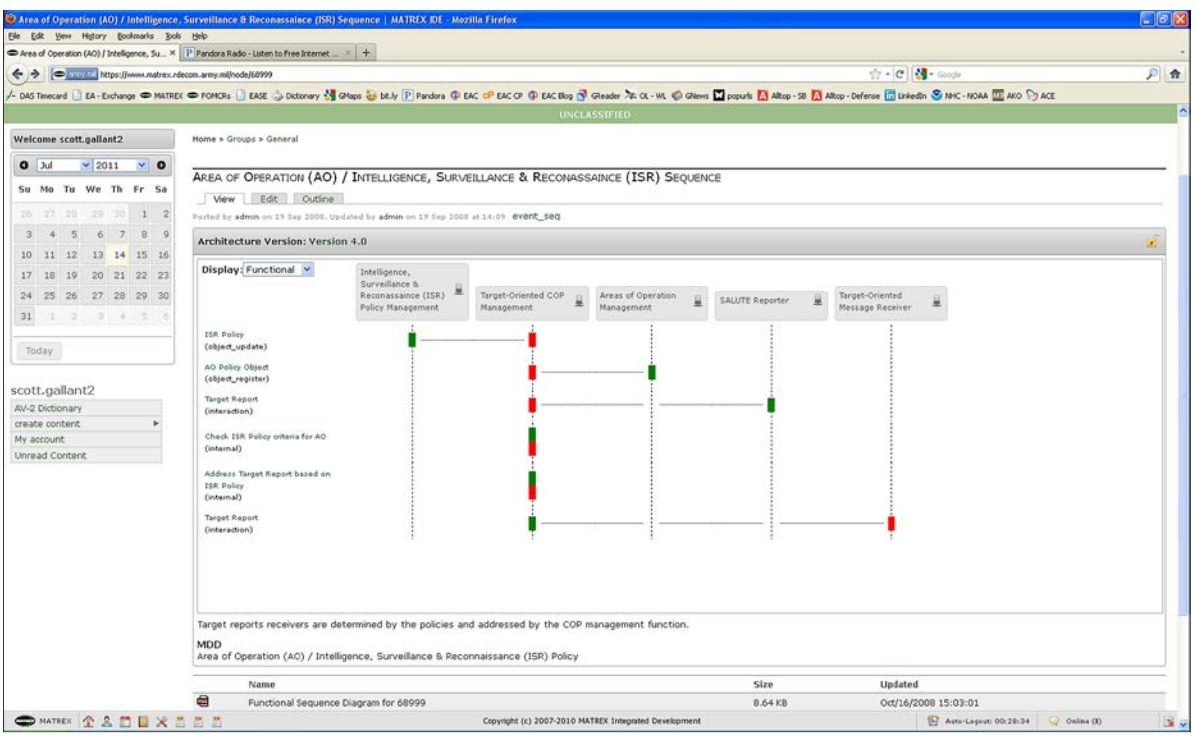


Figure 2: Functional sequence diagram from SDD

Note: COP is Common Operating Picture. All other acronyms are defined within the figure.

diagram. Notice that there are no implementation details in this view. These functions (sequence diagram swim lanes) and information exchange events are describing the military representation design to realize a high level M&S function for Area of Operation (AO)/Intelligence, Surveillance & Reconnaissance (ISR) Policy. Figure 3 shows the technical view which has applications allocated to the swim lanes and object model elements (interactions and objects) allocated to the information exchange events.

This data and relationships amongst the data are stored in a My Structured Query Language (MySQL) [5] database. The data is entered, navigated and viewed using a custom tool built on an open source content management system called Drupal [6].

AUTOMATED ENGINEERING FOR DISTRIBUTED M&S

We have made great strides in implementing some of the core building blocks for applying generative programming techniques to the distributed M&S domain. It appears that there is a tremendous return on investment when supporting the design and implementation of distributed M&S environments, based on anecdotal experience while designing the system and the time it has taken to execute M&S systems in the past.

Capturing the Systems Engineering data within a database-driven infrastructure has allowed us to generate event-specific and design-specific products. The products generated include working test applications that are based on the design, surrogate applications as designed within the SDD, and function and thread specific data collection plans.

To facilitate the development of this capability we leveraged two Government Off-The-Shelf (GOTS) capabilities (1) A software library that abstracts away the middleware protocol details called the ProtoCore and (2) an application generation test harness called the Advanced Testing Capability (ATC) [6]. These software tools serve as the foundation for our design to implement automation. The

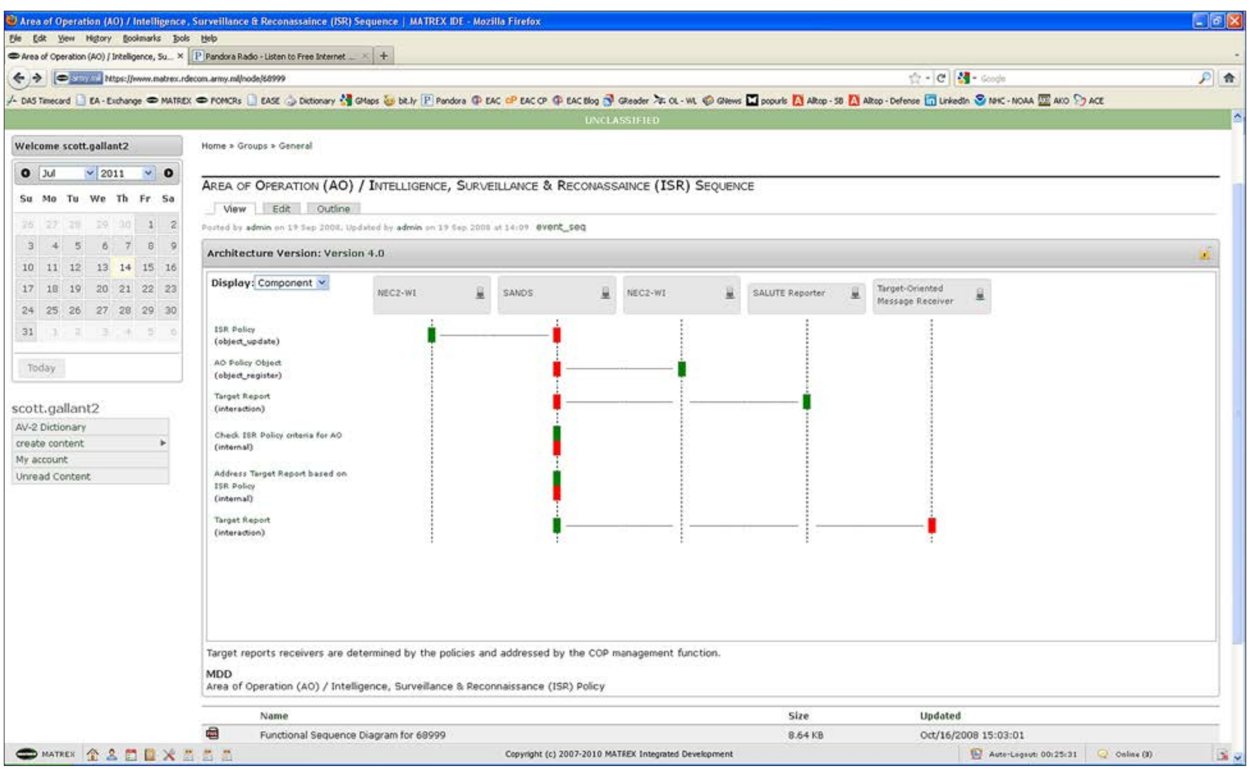


Figure 3: Technical sequence diagram from SDD

Note: NEC2-WI is Network Effects Command & Control Warfighter Interface; SANDS is Situational Awareness Normalization & Dissemination Service; and, SALUTE is Size, Activity, Location, Unit, Time, and Equipment. All other acronyms are defined within the figure or in the previous figure.

ProtoCore is used for connection to middleware protocols and is built using a plug-in architecture. Plug-ins exist to connect to High Level Architecture (HLA) 1.3, HLA 1516 [7], Test and Training Enabling Architecture (TENA) [8] and One Semi-Automated Forces's (OneSAF's) Simulation Object Runtime Database (SORD) [9]. This leads to having all design-based generation of applications to work across many middleware protocols as shown in figure 4.

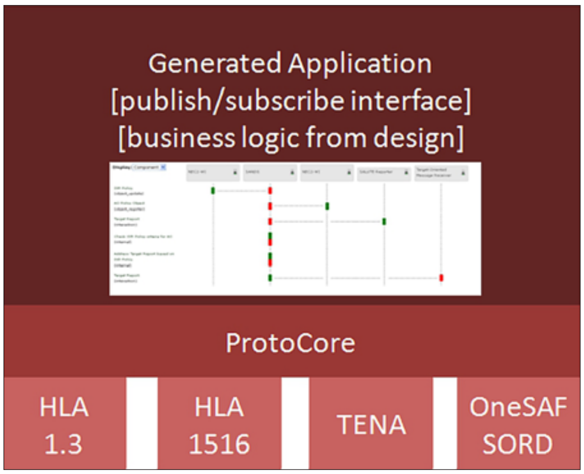


Figure 4: Technical sequence diagram from SDD

To facilitate capturing pseudocode from the systems engineer, we leveraged Groovy [10]. Groovy is a high level language that compiles straight to Java byte code for execution, which is our programming language for the ATC tool and integrates well with the existing capability of ATC to generate applications for test cases. The surrogate generation process starts with the user identifying the function to create the surrogate based on the sequence diagrams within the design that the function participates in. For each inclusion into a sequence diagram, there are inputs and outputs defined for that function (swim lane in the sequence diagram). All of the inputs and outputs are provided in the starting Groovy script. The user can then write their own business logic to use the inputs and publish the outputs in the provided programming language objects. The editor interface allows you to double click on the fields to manipulate and it fills in Groovy syntax for getting access to those fields. This simplifies application development, ensures that the application will provide the correct data based on the design and facilitates a robust implementation based on using existing tools and libraries that have matured (and will continue to mature) over the course of the program's life.

INTEROPERABILITY CHALLENGES

There are many critical qualities that managers of a simulation environment must achieve: traceability from requirements to implementation (and the resultant data collected); integration of applications to the chosen middleware protocol; common data syntax; alignment of data semantics across applications; ease of maintenance; and change propagation throughout the architecture. All of these goals are important and this research attempts to provide traceability, common protocol and object model use and management of changes in design to implementation.

Decomposing the design into smaller blocks of functionality allows the designers to reorganize the blocks in order to accomplish varying high level capabilities. It facilitates reuse of design and ultimately applications and object models. While decomposing the high level capabilities into smaller design decisions, the semantics of the design must also be captured and aligned intelligently. Aligning data semantics ensures applications are communicating based on a consistent understanding of the context and connotation of the information being shared.

When integrating existing applications that are chosen because they share a common syntax, or even for political reasons, (e.g., someone with the authority orders the use of a model; the integration of applications must be backward engineered to the functionality required). Systems are often chosen because of the object model and middleware protocol that they are compatible with. However, compatibility is more than the ability to communicate without compilation errors or crashing. The applications' capability must provide necessary portions of a high level capability and they must provide that functionality in semantic harmony with the other applications within the architecture.

We have been involved in dozens of events and exercises and in every single one; we have witnessed changes to the implementation throughout the integration and preparation. Most of the time, heavy change is still required up to only a few days or hours before the start of execution. Engineers often pull off technical miracles at the last second including working through the night or using one-time fixes that they know are not good long-term solutions. Sometimes those changes work out, but frequently they are the cause of reduced availability, reliability and effective modeling.

Currently, our design relies on the design team to write statements in the design capture about the semantics of information exchange and how the information will be used. This approach is limited and we will be researching better ways of capturing data and process semantics in a manner that is enforceable by the system rather than relying on human engineers to fully account for all the possible semantic integration issues.

M&S CAPABILITY AND SCENARIO INTERFACE

M&S capabilities are captured in the most atomic form to allow for rearrangement toward accomplishment of varying capabilities. When arranging low-level M&S functions into sets of capabilities, the systems engineer can associate high-level warfare capability descriptions with those designed components and threads. Those high-level capabilities are presented to the Executable Architecture Systems Engineering (EASE) user for selection and ultimately execution. The intended users for EASE are M&S users who understand the requirements for an M&S implementation and can select an appropriate set of military representations and scenarios within which to execute the capabilities.

Scenarios have various meanings depending on a person's background. For the purposes of this paper, a scenario is isolated to the military operations that need to be represented in any given execution. Scenarios do not include the technical implementation details. The technical details are assigned to lineups of applications (more specifically, their configurations) indirectly in order to separate technical dependencies from functional capabilities ultimately allowing for a more flexible and reusable set of simulation capabilities within a common architecture.

As seen in figure 5, the user is presented with several categories of scenario criteria and functional capabilities. As the user selects options, a dynamic query reduces the possible execution instances to choose from based on the selections. Once all the requirements have been selected, the user can determine which of the remaining options is

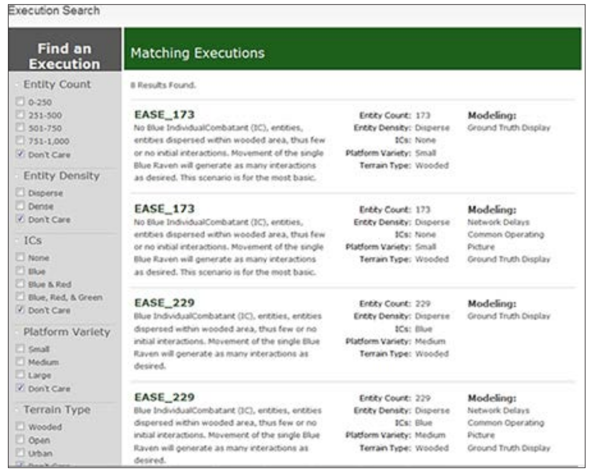


Figure 5: EASE prototype user interface for selecting M&S execution choices

most appropriate for their needs. More advanced information is available on each execution block including the models that will be used, the entity count, terrain type, etc. The user can modify the execution time, the scheduled execution time (in the case that the purpose is to integrate with a separate environment that will occur in the future), and the number of runs they would like. They hit a submission button labeled "Go" and a task request is sent to the deployment management system.

During the execution, live screen shots are captured and displayed back to the user interface. This allows the user to watch the scenario as it executes. Once the execution run completes, the data collected is provided on the user interface. The user has access to the entire set of collected data, a smaller set that only has the object model elements defined as important in the system design and a set of After Action Review (AAR) products generated automatically by the AAR tool that we use. The user can use the databases for their own analysis or examine the set generated for them to determine if the execution run was successful for their specific needs. If not, adjustments to the execution can be made within the limitations of EASE and another run can be launched within minutes. This demonstrates a few of the major benefits of EASE rapid execution launching; quick changes based on scenario and functional capabilities/requirements; and a fully traceable M&S activity from capability needs through the data collected and the AAR artifacts generated from them.

There is also a section of the user interface only accessible by the more technical systems engineer where applications, as discovered from the SDD, are mapped to deployment configurations that are stored on the deployment server. The systems engineer user can adjudicate possible application lineups, apply specific application configurations to a scenario and manage the execution options that the M&S user will have the option of selecting.

DEPLOYMENT AND EXECUTION

There are numerous deployment considerations when designing and executing a distributed M&S event. The required hardware, operating systems, lab space, network connectivity, application installations, configurations, and the alignment of all applications on a common protocol and object model are the major categories of considerations.

We have developed and integrated components that comprise an infrastructure that manages virtual machine assets across several specialized virtualization servers. The infrastructure for virtual machine management by itself is not innovative, but applying virtual machines to M&S in this unique way proves invaluable for the goal of easing execution of distributed M&S. The method for separating and binding functional, scenario specific, mode and technical configurations of distributed M&S applications allows us to minimize the burden of both configuration and execution for simulation engineers.

For each application, there are usually some configuration elements that can be independently adjusted without impacting other application capabilities. An example of this would be the altitude that an aircraft simulation should use. In these types of cases, we have developed an Application Programming Interface (API) between the EASE interface and the deployment system to allow the user access to modify these types of configuration elements.

Composable Application Deployment

A typical application in the M&S domain consists of many elements to configure. Each of these configurations has a purpose so we've attached meaning to each of the configurations as it applies to the functional and technical capabilities of the application. An application, its mode and the specific configurations are pieced together and assigned a unique identifier for the scenario interface portion to align with the design and eventually the execution.

The design and implementation of EASE has highlighted the main types of configuration: scenario (military actions); functional (changing the representation within the model/simulation, such as performance data for a sensor); and technical (e.g., middleware connectivity). Each of these types of configuration should ideally be separable and assigned depending on well-defined execution options. Unfortunately, most applications combine one or more of these types of configurations within the same configuration files and have some of these types of configuration spread across multiple locations making it difficult to easily automate the execution of the application based on the EASE interview.

By decomposing and understanding the configuration options of each application as it relates to the functional and technical capabilities, we can make better reuse decisions when storing the applications in an executable manner. We are storing and linking configurations rather than just simply configuring an application for each possible execution option and saving those results in snapshots.

SUMMARY

The EASE research effort is leading the way with research and development for systems engineering decomposition and traceability from functional requirements to technical implementation. The effort includes tying functionality within distributed M&S to scenarios for execution, code generation, execution automation and execution management, including providing data collection and AAR artifacts back to the user.

The EASE team is constantly coordinating with programs across the Department of Defense (DoD) for technical teaming partners, recommendations, and possible users. The research will continue and we intend on developing depth in each of the described areas with the intent of being used within many other M&S programs. We will gather metrics for distributed M&S execution and report what we expect to be large amounts of cost-savings for developing, managing, executing, and using the resultant products of distributed simulation.

Acknowledgments

Portions of this paper were accepted for publication by the Winter Simulation Conference in December 2011.

This paper was originally published as 11F-SIW-33 in Proceedings of the 2011 Fall Simulation Interoperability Workshop (SIW).

This article has been updated by the original author for this publication, and adjusted to conform with the *M&S Journal* guidelines and format.

References

[1] M. Jamshidi, *System of Systems Engineering,* 1st ed. Hoboken: Wiley & Sons, Inc., 2008.

[2] K. Czarnecki, *Generative Programming: Methods, Tools, and Applications*, Boston: Addison-Wesley Professional, 2000.

[3] Acquisition Community Connection at Defense Acquisition University (DAU). *MIL-STD-3022 DoD Standard Practice Documentation of VV&A for Models and Simulations,* 2008. Available: https://acc.dau.mil/CommunityBrowser.aspx?id=205916

[4] E. Page and D. Lunceford, "Architectural principles for the U.S. Army's simulation and modeling for acquisition, requirements and training (SMART) initiative," *Proceedings of the 2001 Winter Simulation Conference,* Arlington, VA, Volume 2, pp.767-770.

[5] P. DuBois, *MySQL*, 4th ed. Upper Saddle River: Pearson Education, 2008.

[6] A. Byron, A. Berry, N. Haug, J. Eaton, J. Walker and J. Robbins, *Using Drupal,* Sebastopol: O'Reilly Media, Inc., 2008.

[7] Institute of Electrical and Electronics Engineers. *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Framework and Rules, 2010.* Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=893287

[8] R. Noseworthy, "The Test and Training Enabling Architecture," *12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications,* 2008.

[9] J. Logsdon, D. Nash, and M. Barnes, "One Semi-Automated Forces (OneSAF) Capabilities, Architecture, and Processes," DTIC, 2008. Available: http://handle.dtic.mil/100.2/ADA501150

[10] V. Subramaniam, *Programming Groovy: Dynamic Productivity for the Java Developer*, Pragmatic Bookshelf, 2008.

Authors' Biographies

Mr. Scott Gallant

Mr. Scott Gallant is a systems architect and software engineer with Effective Applications Corporation. He has over 15 years experience in distributed computing including U.S. Army modeling and simulation. He has led technical teams on large U.S. Army programs for distributed software and federation design, development and execution management in support of technical assessments, data analysis, and experimentation. He is currently the lead systems engineer and system architect on the Modeling Architecture for Technology, Research, and EXperimentation program managed by the U.S. Army Research Laboratory Simulation and Training Technology Center. He earned a B.S. in computer science from George Mason University in Fairfax, VA.

Mr. Christopher J. Metevier

Mr. Christopher J. Metevier is the technology program manager of the Modeling Architecture for Technology, Research, and EXperimentation program and chief of the Advanced Simulation Branch at the U.S. Army Research Laboratory Simulation and Training Technology Center. He has over 20 years of experience with the U.S. Navy and U.S. Army in the Modeling & Simulation (M&S) field. His M&S experience extends across the acquisition lifecycle and includes the research, development, adaptation, integration, experimentation, test, and fielding of numerous simulation technologies and systems. He received an MBA from Webster University and a B.S. in electrical engineering from the University of Central Florida.

Mr. Chris Gaughan

Mr. Chris Gaughan is the chief engineer for advanced simulation and deputy technology program manager of the Modeling Architecture for Technology, Research, and EXperimentation program at the U.S. Army Research Laboratory Simulation and Training Technology Center. He has a diverse portfolio of distributed simulation projects that support the full spectrum of the Department of Defense Acquisition Life Cycle. From 2004-2009 he worked at the Edgewood Chemical Biological Center (ECBC), where he served as the configuration manager of the Chemical-Biological-Radiological-Nuclear (CBRN) Simulation Suite. During his tenure at ECBC, he was the principal investigator for numerous Joint Science and Technology Office projects focused on CBRN Modeling and Simulation (M&S). He has provided M&S analytical support to the Joint Program Executive Office for Chem-Bio Defense and to the Training & Doctrine Command Maneuver Support Battle Lab. He received a M.S. and a B.S. in electrical engineering from Drexel University in Philadelphia, PA.

Modeling and Validation Challenges for Complex Systems

AUTHOR

Dr. Mikel D. Petty
University of Alabama in Huntsville
301 Sparkman Drive, Shelby Center 144, Huntsville, AL 35899
pettym@uah.edu

KEYWORDS

Verification and Validation, Complex Systems, Sensitivity to Initial Conditions, Emergent Behavior, Composition of Components

ABSTRACT

MANY IMPORTANT SYSTEMS, BOTH NATURAL AND ARTIFICIAL, MAY BE CLASSIFIED AS COMPLEX, AND THE STUDY OF COMPLEX SYSTEMS IS ONGOING. SUCH SYSTEMS HAVE SPECIAL DEFINING CHARACTERISTICS, INCLUDING SENSITIVITY TO INITIAL CONDITIONS, EMERGENT BEHAVIOR, AND COMPOSITION OF COMPONENTS. COMPLEX SYSTEMS ARE INCREASINGLY PREVALENT AS THE SUBJECT OF MODELING EFFORTS. THERE ARE AT LEAST TWO REASONS FOR THIS; FIRST, THE SYSTEMS THAT ARE OF THE GREATEST PRACTICAL INTEREST AND THUS MOST LIKELY TO BE MODELED TEND TO BE COMPLEX, AND SECOND, BECAUSE COMPLEX SYSTEMS RESIST CLOSED FORM ANALYSIS MODELING IS OFTEN THE ONLY WAY TO STUDY THEM. UNFORTUNATELY, THE SPECIAL CHARACTERISTICS OF COMPLEX SYSTEMS LEAD TO ADDITIONAL CHALLENGES IN BOTH EFFECTIVELY MODELING THEM AND IN VALIDATING THE MODELS. THIS PAPER, WHICH TAKES THE FORM OF AN INTRODUCTORY TUTORIAL AND LITERATURE SURVEY, FIRST DEFINES COMPLEX SYSTEMS IN TERMS OF THEIR KEY CHARACTERISTICS AND DESCRIBES HOW VALIDATION RISK APPLIES TO MODELS OF THEM. IT THEN IDENTIFIES A SERIES OF MODELING AND VALIDATION CHALLENGES THAT FOLLOW FROM THE DEFINING CHARACTERISTICS AND SUGGESTS MITIGATION APPROACHES FOR THOSE CHALLENGES.

1. INTRODUCTION

> *...our best equations for the weather differ from our best computer models based on those equations, and both of those systems differ from the real thing...* [1]
>
> *...complexity lies somewhere between order and chaos* [2].

Complex systems, where "complex" is meant in the sense of complexity theory as opposed to simply a synonym for "complicated," are with increasing frequency the subject of modeling efforts. Among several reasons for this, two stand out. First, the systems of the greatest practical interest, and thus those most likely to be worth the effort and expense of being modeled, tend to be complex. Second, as a result of their special characteristics, complex systems generally resist closed form mathematical analysis, and so modeling is often the best or even the only way to study and experiment with them.

Complex systems have a number of special defining characteristics, including sensitivity to initial conditions, emergent behavior, and composition of components. Unfortunately for those involved in modeling complex systems, these special characteristics of complex systems lead to additional challenges beyond those encountered with non-complex systems in both modeling them accurately and effectively and in reliably and completely validating the models.

This paper, which is meant as an introductory tutorial and brief literature survey, has four main sections.[1] The first describes complex systems and lists their defining characteristics, and motivates the interest in validating models of complex systems by discussing validation risk. Then, each of the following sections discusses one of three selected defining characteristics of complex systems (sensitivity to initial conditions, emergent behavior, and composition of components), explaining why the characteristic in question makes modeling and validation more difficult and offering some approaches to dealing with and mitigating the difficulties.

2. COMPLEX SYSTEMS

Complex systems were recognized as qualitatively distinct from non-complex systems at least as early as 1984, with the founding of the Santa Fe Institute, a research institute devoted to complexity theory [3]. Since then, a body of specialized knowledge has been developed on the subject, driven by both theoretical and experimental investigations [4].

2.1 Definition of complex systems

A range of definitions of complex system are available. Although the definitions are far from as reassuringly consistent or precise as that of, say, an equivalence relation (e.g., see [6]), they are nevertheless informative.

> *A system comprised of a (usually large) number of (usually strongly) interacting entities, processes, or agents, the understanding of which requires the development, or the use of, new scientific tools, nonlinear models, out-of equilibrium descriptions and computer simulations* [7].[2]

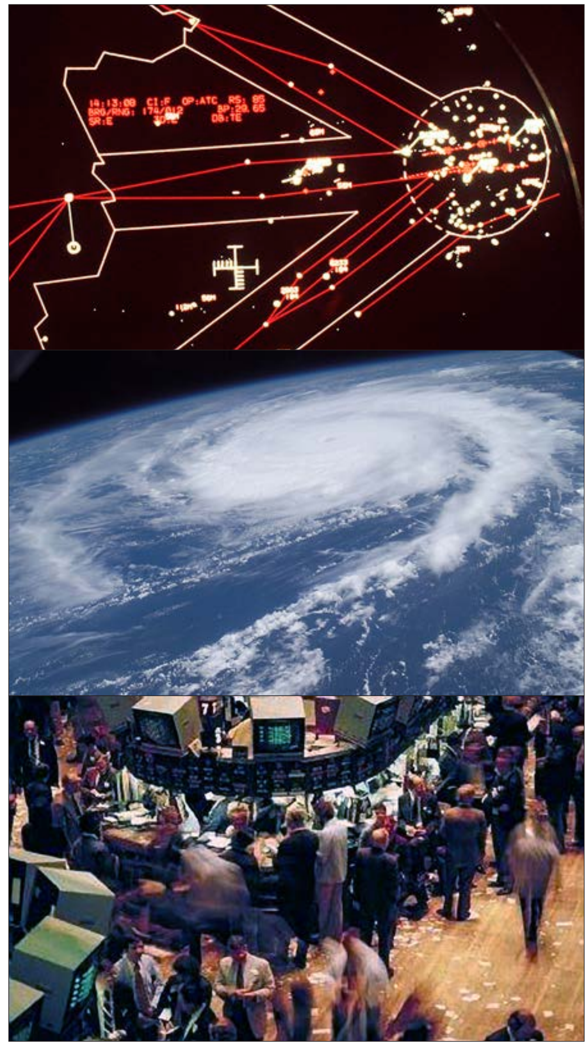


Figure 1: Examples of complex systems: air traffic control, weather, and the stock market.

> *A complex system is one whose evolution is very sensitive to initial conditions or to small perturbations, one in which the number of independent interacting components is large, or one in which there are multiple pathways by which the system can evolve* [8].

> *Complex systems are neither ordered nor random, but combine elements of both kinds of behaviour in a very elusive but striking manner* [9].

[1]An earlier version of this paper was published as 12S-SIW-025 in the Proceedings of the Spring 2012 Simulation Interoperability Workshop (SIW) [5]. This version has been corrected and enhanced by the original author for this publication. See copyright and permissions for use at the end of this paper.

[2]Quoted from [7], where it is attributed to [10].

Both natural and artificial systems satisfy these definitions; examples of systems that are considered to be complex are illustrated in figure 1.[3]

There is general consensus that certain defining characteristics or properties are associated with complex systems. These characteristics are individually arguable, in that not every complex system necessarily exhibits every one of these characteristics, but they are collectively definitive; most complex systems will exhibit most of these characteristics. Taken together, they define the class of complex systems and serve to distinguish them from non-complex systems. A list of the defining characteristics with brief descriptions follows; the first three are described in detail in the following sections, whereas the others are briefly described here.[4]

1. ***Sensitivity to initial conditions.*** See section 3.
2. ***Emergent behavior.*** See section 4.
3. ***Composition of components.*** See section 5.
4. ***Uncertain boundaries.*** Determining the boundary between a complex system and the environment in which it is situated and with which it interacts can be difficult.
5. ***Nesting.*** Components of a complex system may themselves be complex systems.
6. ***State memory.*** Future states of a complex system often depend on past states in ways that are difficult to understand or model.
7. ***Non-linear relationships.*** Relationships between components of a complex system may be non-linear, which means a small cause may have a large effect.
8. ***Feedback loops.*** Negative (damping) and positive (amplifying) feedback loops exist between elements of complex system.

2.2 Validation risk in models of complex systems

Important systems, complex systems, and modeled systems overlap to a great extent. Systems that are important to their users, for reasons of safety, economics, or ubiquity, are often complex; the reverse is also true. For example, financial markets are important to those who participate in them, whether voluntarily or involuntarily, because of their potential impact on the participants' quality of life and long-term security, and they exhibit all of defining characteristics listed earlier. Similarly, systems that are important are also often modeled, because their importance makes them more likely to be worth the effort and expense of being modeled; and again the reverse is also true. Finally, complex systems are often modeled, and once more the reverse is true. Because of their inherent structure, complex systems are often difficult to study using closed form mathematical analysis [2]. Consequently, modeling is often the best or even the only way to study or experiment with them.

Models are subject to validation risk. The general concept of validation risk is that validation that is improperly or incompletely performed can result in risk to the developers and/or the users of the model. This general notion has been refined into specific types of validation error and the type of validation risk that results from each. The validation errors are known as Type I, Type II, and Type III, and are defined in a manner that closely parallels the like-named error types in statistical hypothesis testing. Figure 2 summarizes these error types.[5]

	Model valid	Model not valid	Model not relevant
Model used	Correct	**Type II error** Use of invalid model; Incorrect V&V; Model user's risk; **More** serious error	**Type III error** Use of irrelevant model; Accreditation mistake; Accreditor's risk; **More** serious error
Model not used	**Type I error** Non-use of valid model; Insufficient V&V; Model builder's risk; **Less** serious error	Correct	Correct

Figure 2: Types of validation errors and risk [11] (adapted from [12])

[3]Image acknowledgments for figure 1: Air traffic control, U. S. Air Force, public domain; Weather, National Aeronautics and Aerospace Administration, Public domain; Stock market, National Institute for Standards and Technology, Public domain.

[4]An overlapping but somewhat different list is given in [13]; that list includes *adaptiveness* and *self-organization*.

[5]Definitions of Type I and Type II validation errors analogous to the statistical errors of the same name appear in [15] and in subsequent editions of this source, e.g., [16].

Whenever a model is used validation risk exists, and for a model of an important system, that risk is a function of both the importance of the system and to the model's intended use. Obviously, a Type II validation error clearly has less potential consequences for a model of ant behavior being used for a video game than a model of metal fatigue being used to design the airframe of an airliner. Decisions made about important system using models can have major impact. As an example, consider the 2008 financial crisis in the United States. Some financial analysis have argued that that crisis was in significant part triggered by a financial model, namely the famous (or infamous) Gaussian copula, which is a model of the prices of collateralized debt obligations:

$$\Pr[T_A < 1, T_B < 1] = \Phi_2(\Phi^{-1}(F_A(1)), \Phi^{-1}(F_B(1)), \gamma) \quad [14]$$

The mathematical and notational details of this model need not concern us here. Conceptually, the bounds of validity of this widely-used model were not fully understood by its users. The model was based on the assumption that the price of a credit default swap was correlated with, and thus could be used to predict, the price of mortgage backed securities. Because the model was easy to use and compute, it was soon employed by a large portion of mortgage issuers, rating agencies, and financial investors. In fact, the model was ultimately invalid and its use constituted a Type II error. The result of that error is all too well known:

> *Then the model fell apart. ...financial markets began behaving in ways that users of [the] formula hadn't expected. ...ruptures in the financial system's foundation swallowed up trillions of dollars and put the survival of the global banking system in serious peril* [14].

The significant overlap of important systems, complex systems, and modeled systems means that our models are often of systems that are both important and complex; their importance magnifies validation risk, and their complexity complicates validation. Given the validation risk associated with models of important and complex systems, it is prudent to expend validation effort proportional to the risk and to adapt or develop validation methods suitable for complex systems.

3. SENSITIVITY TO INITIAL CONDITIONS

> *This is only true when small variations in the initial circumstances produce only small variations in the final state of the system. In a great many physical phenomena this condition is satisfied; but there are other cases in which small initial variation may produce a very great change in the final state of the system, as when the displacement of the 'points' causes a railway train to run into another instead of keeping its proper course* [17].

> *Small differences can build upon themselves and create large differences, making precise prediction difficult* [2].

The first of the three defining characteristics of complex systems to be examined for its effect on modeling and validation is sensitivity to initial conditions. Here the phrase "initial conditions" refers, of course, to either the starting state of the system (e.g., a rocket motor at ignition), or if the system has an effectively continuous existence (e.g., the weather), the state of the system at the beginning of the time period being studied or modeled. The state

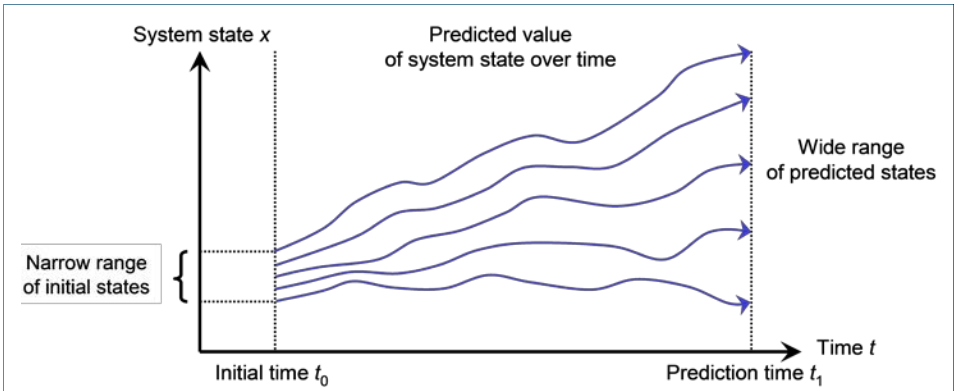


Figure 3: Sensitivity to initial conditions; system state diverges over time.

evolution of complex systems can be highly sensitive to its initial conditions, with the result that small differences in initial state can become magnified over time into large differences in future state [1]. Figure 3 illustrates this; in the figure, the horizontal axis represents time, advancing from left to right, and the vertical axis represents system state, notionally simplified to a single variable. States that are only slightly different at some initial time t_0 can evolve away from each other, becoming arbitrarily different at some future time t_1.

Models of complex systems, if they accurately represent the system's characteristics, can be similarly sensitive to initial conditions. From two model starting states that are quite similar, the execution of a model of a complex system can produce widely divergent end states.

3.1 Modeling

Sensitivity to initial conditions can introduce modeling challenges in these ways:

1. ***Implementation side effects.*** Technical aspects of the model that are purely implementation details and do not correspond to any aspect of the simuland[6] can have significant side effects that influence, or even overwhelm, the results. A well-known example is the effect of the numerical precision of the implementation language on numerical integration of differential equations in physical models [18].[7] In stochastic models that rely on random number generators, the seed and cycle length of the random number stream can, through the magnifying effect of sensitivity, significantly affect the model's results [16].
2. ***Sensitivity consistency.*** If a complex system is sensitive to initial conditions, the modeler may seek similarly sensitivity in a model of that system. However, given the nature of the sensitivity, it can be quite difficult accurately to match the model's sensitivity to that of its simuland. Even if both the complex system and a model of it are sensitive, small differences between simuland sensitivity and model sensitivity can lead to large differences in outcomes.
3. ***Input imprecision.*** Because sensitivity magnifies small differences in initial conditions, a small difference between the simuland's true initial state and the values of the model input data describing that state can again lead to large differences in outcomes. Consider, for example, a weather model that uses a three-dimensional array of air temperature, pressure, and humidity values to define the initial state of the atmosphere. Small errors in measuring those values can be magnified, as the model executes, into large discrepancies between the model's prediction and the actual weather. The input data precision needed by the model to accurately predict the simuland's future may exceed that obtainable due to limits in instrumentation accuracy or observation availability. This observational uncertainty is one reason that the useful predictive power of current weather models is currently limited to a few days, and the maximum achievable limit, even with perfect models, is considered to be "about two weeks" [19].

These methods can mitigate the modeling challenges associated with sensitivity to initial conditions:

1. ***Selective abstraction.*** During conceptual modeling, identify simuland features and state variables that are not required for the model to satisfy its intended purposes. Omit them in the implemented model, thereby eliminating them as possible sources of sensitivity.
2. ***Ensemble forecasting.*** The core idea of ensemble forecasting is to execute multiple runs of a model, each of which was initialized with slightly different initial states, and then develop a prediction based on the multiple results.[8] The differences in the inputs are intended to reflect the uncertainty in the knowledge of the initial state. The multiple results may be aggregated or averaged, and the variation and divergences between them analyzed; the details of aggregation and analysis depend on the application, but statistical methods are often employed. In some forms and contexts this is a familiar idea; modelers using a discrete event simulation to study a queuing system often conduct multiple trials, each beginning with a different random number seed. In the case of weather models, different values for the initial conditions of the atmosphere may be used, with the differences generated based on the noise or uncer-

---

[6]A *simuland* is the system, phenomenon, or process that is the subject of a model, i.e., the modeled system [11].

[7]For example, a seemingly reasonable fourth-order Runge-Kutta integration with a fixed time step used to calculate an orbit in a two-body (sun and planet) gravitational system completely breaks down in the vicinity of the sun due to numerical precision issues, with the result that the simulated planet incorrectly "flies off completely into space [18]."

[8]In addition to multiple runs of a single model, ensemble forecasting may also refer to an aggregating or merging of the results of multiple models. This approach is used to predict hurricane tracks.

tainty in the observations upon which the input is based [1]. The uncertainty of the forecast may be estimated based the variation in the different forecasts generated.

3.2 Validation

Sensitivity to initial conditions can introduce validation challenges in these ways:

1. ***Results distributions.*** Broad distributions (i.e., large variance) in both simuland observations and model results can reduce the power of statistical comparisons of the two [20].
2. ***Sensitivity analysis.*** The potential for widely divergent outcomes from closely similar initial conditions can complicate conventional sensitivity analysis by requiring more closely spaced sampling of the response surface to capture the response variation.
3. ***Input imprecision.*** Measurement errors unavoidably introduce uncertainty into measurements of physical systems [21]. Because of sensitivity to initial conditions, small uncertainties in model input values may be magnified, making comparisons of simuland observations and model results more difficult.

These methods can mitigate the validation challenges associated with sensitivity to initial conditions:

1. ***Increased trials.*** Increasing the number of trials (i.e., executions of the model) can regain some statistical power through larger sample sizes.
2. ***Sensitivity analysis.*** Sensitivity analysis can be used as a validation method by statistically comparing the magnitude and variability in the simuland observations to the magnitude and variability in the model results, in effect using sensitivity as a metric for validation comparison [12], [22].
3. ***Precision awareness.*** Understand the precision available in simuland observation data, and based on that precision, use an appropriate comparison threshold when comparing simuland observations and model results. For example, it is a mistake to expect the model to match the simuland within one unit when the observations are only accurate to within five units.

Figure 4: Emergent behavior in a natural system; flocking emerges from individual bird actions.

4. EMERGENT BEHAVIOR

We are dealing with a [complex] system when…the entire system exhibits properties and behaviors that are different from those of the parts [23].

Much of the focus of complex systems is how…interacting agents can lead to emergent phenomena. … individual, localized behavior aggregates into global behavior that is, in some sense, disconnected from its origins [2].

The philosophical core of complexity theory is the concept of emergence, in which a system may transcend its components… [9].

The second of the three defining characteristics of complex systems to be examined for its effect on modeling and validation is emergent behavior. Emergent behavior is behavior that is not explicitly encoded in the agents or components that make up the model; rather, it emerges during a simulation from the interaction of agents or components with each other and the simulated environment [13].

An important aspect of emergent behavior is that it is not directly predictable or anticipatable from the individual agents' or components' behaviors, even if they are known completely. Figure 4 illustrates a form of natural emergent behavior that exhibits this.[9] Emergent behavior is, in some intuitive sense, unexpected; it produces "surprise" in the observer [2]. There is the possibility of multiple levels of emergence, with mesoscale behavior that emerges from microscale interactions itself contributing to the emergence of even higher level macroscale behaviors [2].

[9]Image acknowledgment for figure 4: C. A. Rasmussen, Public domain, Wikipedia Commons.

4.1 Modeling

Emergent behavior can introduce modeling challenges in these ways:

1. ***Incomplete observations.*** Because emergent behavior is potentially unpredictable, available observations of simuland may not include all possible simuland emergent behavior. Indeed, the modeler may not even be aware of some potential simuland emergent behaviors.
2. ***Indirect representation.*** Because emergent behavior is not, in general, predictable from the individual behavior of agents or components within the complex system, those aspects or characteristics of it that produce emergent behavior can be difficult to identify and include in the model.
3. ***Overabstraction risk.*** Because emergent behavior is produced indirectly from potentially non-obvious aspects of simuland, modeler may unintentionally abstract away those aspects, eliminating the possibility of the model generating interesting or important emergent behavior.

These methods can mitigate the modeling challenges associated with emergent behavior:

1. ***Additional observations.*** Increasing the number or duration of simuland observations, and broadening the range of conditions under which the simuland is observed, can increase the likelihood of observing and detecting the full repertoire of emergent behaviors.
2. ***Conceptual modeling focus***. When developing the conceptual model of a complex system, give explicit attention to the inclusion of emergent behaviors, or aspects of the complex system that may give rise to emergent behaviors (such as inter-agent interactions).

4.2 Validation

Emergent behavior can introduce validation challenges in these ways:

1. ***Incomplete observations.*** Emergent behavior is, by its nature, difficult to predict, observe, and measure in the simuland; this was already noted as a modeling challenge. It is also a validation challenge, as some emergent behavior observed in the model results may not have been observed in the simuland, thus leaving gaps in the data for use in validating the model's behavior.
2. ***Incomplete results.*** Conversely, emergent behavior observed in the simuland can be similarly difficult to generate in the model results. Of course, if the behavior is not in the model results, it can not be validated beyond noting that it is missing.
3. ***Face validation unreliability.*** Because of emergent behavior is unpredictable, face validation based on subject matter experts is less reliable. The experts may overestimate or underestimate the likelihood of occurrence of emergent behavior, or they may have little direct knowledge of it.[10]
4. ***Test case uncertainty.*** Because emergent behavior is not directly predictable, designing model validation test cases (trials) which will generate specific emergent behaviors for validation can be difficult.

These methods can mitigate the validation challenges associated with emergent behavior:

1. ***Additional observations.*** Increasing the number or duration of simuland observations, and broadening the range of conditions under which the simuland is observed, increases the likelihood of acquiring the data needed to validate emergent behavior.
2. ***Structured face validation.*** To overcome deficiencies in the knowledge of any particular subject matter expert, use teams of experts and conduct organized face validation assessments. The latter may be based pre-planned validation scenarios designed to cover the full range of simuland behaviors [11], [24] and employ Delphi methods, wherein panels of experts make forecasts and examine the model's results over multiple rounds, eventually converging on a consensus assessment of validity [25].
3. ***Scenario space search.*** Generate validation test cases automatically via heuristic search in scenario space, i.e., generating new test cases based on previous trials that elicit some emergent behavior; this method requires metrics for emergent aspects of complex systems.
4. ***Semi-automated model adaptation.*** For each abstraction (i.e., simplification or estimation) within a model, pre-defined alternative model abstractions embedded in the source code can be exploited by an optimization-based adaptation process to generate emergent behavior under user specific conditions of interest [26]. Essentially, those conditions that produce emergent behavior are predicted by the user, found by the adaptation process, and compared.

[10] Experts often underestimate the probability of an unlikely event, implicitly assuming a normal probability distribution when a "fatter tailed" distribution would be more appropriate [2]. Examples of such distributions and their asserted applications include power laws for city sizes [27], deaths in warfare [28], and Lévy stable laws for stock market price changes [19].

5. COMPOSITION OF COMPONENTS

> *Because of the prevalence of inter-connections, we cannot understand systems by summing the characteristics of the parts or the bilateral relations between pairs of them* [23].

> *We would, however, like to make a distinction between complicated worlds and complex ones. In a complicated world, the various elements that make up the system maintain a degree of independence from one another. …Complexity arises when the dependencies among the elements become important* [2].

The third of the three defining characteristics of complex systems to be examined for its effect on modeling and validation is composition of components.[11] Complex systems are, by definition, composed of interacting components. Similarly, models of complex systems are often composed of submodels, and those submodels are most typically organized in a structure that reflects the structure of the complex system itself. For example, a spacecraft model may be composed of power system and thermal submodels, with the thermal submodel providing input to power system model to predict power loading. Figure 5 illustrates a notional model-submodel structure. In the figure, the overall model is composed of three submodels. The connecting arrows show data from model inputs, through submodel inputs and outputs, to model outputs.

5.1 Modeling

Composition of components can introduce modeling challenges in these ways:

1. ***Interface compliance.*** The existence of multiple submodels, and thus the need for interfaces between them, adds new opportunities for modeling errors, such as mismatches in data types, measurement units, and execution sequence.[12]
2. ***Architecture selection.*** The appropriate software architecture framework for organizing and connecting the component models (such as hierarchy, blackboard, or agent-based) may not be obvious, and it may have unintended effects on the model results [30].
3. ***Model correlation.*** Different component models may have differences (such as underlying assumptions, representational granularity, or level of fidelity) that negatively affect the overall model's results [31].

These methods can mitigate the modeling challenges associated with composition of components:

1. ***Interface analysis.*** Specifically examine submodel-to-submodel interfaces to determine if interface structures are consistent and accurate [12].
2. ***Known problem review.*** Review available lists of known interoperability problems typically encountered to see if they apply [32].
3. ***Architecture reuse.*** Reuse and revise known model architectures when appropriate and exploit available architecture-based systems engineering processes (e.g., the Distributed Simulation Engineering and Execution Process) [33].

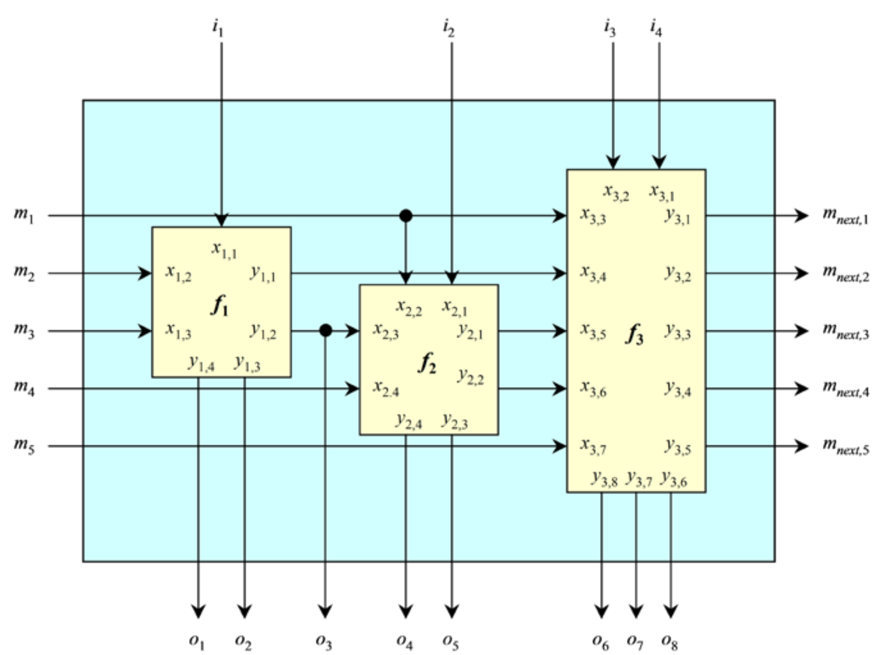


Figure 5: Composition of components; a model composed of three submodels (adapted from [29])

[11]As discussed earlier, interactions between those components can lead to emergent behavior.

[12]Arguably, the entire subject of simulation interoperability is embedded in this modeling challenge. Clearly, this is no small matter.

4. ***Conceptual model verification.*** Compare component models' conceptual models to detect model correlation errors.

5.2 Validation

Composition of components can introduce validation challenges in these ways:

1. ***Weakest link validity.*** The overall validity of a model assembled as a composition of component models may be limited by the lowest fidelity component model. For example, a high fidelity ground vehicle movement model composed with a low fidelity terrain model will likely not produce accurate movement speeds.
2. ***Error location ambiguity.*** Errors in model results detected during model validation may be difficult to associate with correct component model; indeed, they may result from an interface error, rather than one of the component models.
3. ***Statistical method unsuitability.*** The statistical methods used most often in validation typically compare single variables, e.g., the Student *t* test compares the means of two populations, or the Mann-Whitney *U* test determines whether two independent samples of observations come from the same distribution. Models of complex systems have states represented by multiple non-linear variables related non-linearly, requiring the of use multivariate methods that accommodate non-linear effects [12].
4. ***Noncomposability of validity.*** In a model assembled as a composition of components, i.e., from submodels, the submodels are typically validated individually. Unfortunately, submodel validity does not ensure composite model validity; even if the submodels are separately valid, the composite models may not be. It has been mathematically proven that for non-trivial models separately valid component models can not be assumed to be valid when composed [34].

These methods can mitigate the validation challenges associated with composition of components:

1. ***Uncertainty estimation.*** Determine or estimate the possible error range for key model results variables for each component model. Then propagate and accumulate those errors to find the overall error range for the same variables for the composite model [35]. If the overall error is too large, revise the model.
2. ***Non-linear multivariate statistics.*** Apply multivariate statistical methods to validation of non-complex systems models. For example, Hotelling T^2-statistic, which is a generalization of Student's *t* statistic that is used in multivariate hypothesis testing, can be used for constructing ellipsoidal joint confidence intervals in validation [36].
3. ***Composition validation.*** During validation of a composite model, validate both the component models individually and overall composite model. This is directly analogous to conventional unit and system testing in software engineering practice.

SUMMARY

Complex systems, which are increasingly often the subject of modeling efforts, have certain defining characteristics that make them more difficult to model and make models of them more difficult to validate. The specific modeling and validation challenges can be associated with the complex system characteristic that causes them. Although these challenges can be problematic, and in some cases are in principle impossible to overcome entirely, they can often be mitigated through informed application of appropriate methods.

References

[1] L. Smith, *Chaos: A Very Short Introduction*, Oxford University Press: Oxford, England, 2007.

[2] J. H. Miller and S. E. Page, *Complex Adaptive Systems: An Introduction to Computational Models of Social Life,* Princeton University Press: Princeton NJ, 2007.

[3] M. M. Waldrop, *Complexity: The Emerging Science at the Edge of Order and Chaos,* New York: Simon & Schuster, 1992.

[4] R. Lewin, *Complexity: Life at the Edge of Chaos,* Macmillan: New York, 1992.

[5] M. D. Petty, "Modeling and Validation Challenges for Complex Systems," *Proceedings of the Spring 2012 Simulation Interoperability Workshop*, Orlando, pp. 178-187, March 2012.

References (Continued)

[6] S. S. Epp, *Discrete Mathematics with Applications, Fourth Edition,* Brooks/Cole: Boston, 2011.

[7] F. Schweitzer, *Brownian Agents and Active Particles: Collective Dynamics in the Natural and Social Sciences,* Springer: Berlin, Germany, 2003.

[8] G. M. Whitesides and R. F. Ismagilov, "Complexity in Chemistry," *Science,* Vol. 284, No. 5411, pp. 89-92, April 1999.

[9] I. Stewart, *Does God Play Dice?*, Blackwell Publishers: Malden, 2002.

[10] *Advances in Complex Systems* Available: http://journals.wspc.com.sg/acs/acs.html/.

[11] M. D. Petty, "Verification, Validation, and Accreditation," in J. A. Sokolowski and C. M. Banks (Editors), *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains,* John Wiley & Sons: Hoboken, 2010, pp. 325-372.

[12] O. Balci, "Verification, Validation, and Testing," in J. Banks (Ed.), *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*, John Wiley & Sons: New York, 1998, pp. 335-393.

[13] G. P. Williams, *Chaos Theory Tamed,* Joseph Henry Press: Washington, 1997.

[14] F. Salmon, "Recipe for Disaster: The Formula That Killed Wall Street," *Wired*, Vol. 17, No. 3, March 2009.

[15] J. Banks, J. S. Carson, and B. L. Nelson, *Discrete-Event System Simulation, Second Edition*, Prentice-Hall: Upper Saddle River, 1996.

[16] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation, Fifth Edition*, Prentice Hall: Upper Saddle River, 2010.

[17] J. C. Maxwell, *Matter and Motion*, Society for Promoting Christian Knowledge, London UK, 1873.

[18] W. N. Colley, "Modeling Continuous Systems," in J. A. Sokolowski and C. M. Banks (Editors), *Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains,* John Wiley & Sons: Hoboken, 2010, pp. 99-130.

[19] J. L. Casti, *Searching for Certainty: What Scientists Can Know About the Future*, William Morrow and Company: New York, 1990.

[20] C. H. Brase and C. P. Brase, *Understandable Statistics: Concepts and Methods,* Houghton Mifflin: MA, 2009.

[21] M. L. Adams, D. M. Higdon, J. O Berger, D. Bingham, W. Chen, R. Ghanem, O. Ghattas, J. Meza, E. Michielssen, V. N. Nair, C. W. Nakhleh, D. Nychka, S. M. Pollock, H. A. Stone, A. G. Wilson, and M. R. Zika, *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification,* The National Academies Press: Washington, 2012.

[22] F. Liu and M. Yang, "Validation of System Models," *Proceedings of the IEEE International Conference on Mechatronics and Automation*, Niagara Falls Canada, August 2005, pp. 1721-1725.

[23] R. Jervis, "From Complex Systems: The Role of Interactions," *Coping with the Bounds: Speculations on Nonlinearity in Military Affairs,* National Defense University, Washington D.C., 1998.

[24] L. A. Belfore, J. J. Garcia, E. K. Lada, M. D. Petty, and W. P. Quinones, "Capabilities and Intended Uses of the Joint Operations Feasibility Tool," *Proceedings of the Spring 2004 Simulation Interoperability Workshop*, Arlington VA, April 18-23 2004, pp. 596-604.

[25] G. Rowe and G. Wright, "Expert Opinions in Forecasting: Role of the Delphi Technique," in J. Armstrong (Ed.), *Principles of Forecasting: A Handbook for Researchers and Practitioners,* Kluwer, Boston MA, 2001.

[26] R. Gore and P. F. Reynolds, "Applying Causal Inference to Understand Emergent Behavior," *Proceedings of the 2008 Winter Simulation Conference,* Miami FL, December 7-10 2008, pp. 712-721.

[27] H. Simon, "On a Class of Skew Distribution Functions," *Biometrika*, Vol. 42, No. 3-4, December 1955, pp. 425-440.

[28] L. Richardson, *Statistics of Deadly Quarrels,* Boxwood Press, Pittsburgh PA, 1960.

[29] E. W. Weisel, R. R. Mielke, and M. D. Petty, "Validity of Models and Classes of Models in Semantic Composability," *Proceedings of the Fall 2003 Simulation Interoperability Workshop,* Orlando FL, September 14-19 2003, pp. 526-536.

[30] M. Shaw and D. Garlan, *Software Architecture, Perspectives on an Emerging Discipline*, Prentice Hall: Upper Saddle River NJ, 1996.

[31] M. Spiegel, P. F. Reynolds, D. C. Brogan, "A Case Study of Model Context for Simulation Composability and Reusability," *Proceedings of the 2005 Winter Simulation Conference*, Orlando FL, December 4-7 2005, pp. 437-444.

References (Continued)

[32] D. Gross and W. V. Tucker, "A Foundation for Semantic Interoperability," *Proceedings of the Fall 2007 Simulation Interoperability Workshop,* Orlando FL, September 16-21 2007.

[33] Institute of Electrical and Electronics Engineers, *IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP),* IEEE Standard 1730™-2010.

[34] E. W. Weisel, R. R. Mielke, and M. D. Petty, "Validity of Models and Classes of Models in Semantic Composability," *Proceedings of the Fall 2003 Simulation Interoperability Workshop*, Orlando FL, September 14-19 2003, pp. 526-536.

[35] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegart, and K. F. Alvin, *Estimation of Total Uncertainty in Modeling and Simulation*, Sandia National Laboratories, SAND2000-0824, April 2000.

[36] O. Balci and R. Sargent, "Validation of simulation models via simultaneous confidence intervals," *American Journal of Mathematical and Management Science,* Vol. 4, No. 3-4, 1984, pp. 375-406.

Author's Biography

Dr. Mikel D. Petty

Dr. Mikel D. Petty is director of the University of Alabama in Huntsville's (UAH) Center for Modeling, Simulation, and Analysis, associate professor of Computer Science, and research professor of Industrial and Systems Engineering and Engineering Management. Prior to joining UAH, he was chief scientist at Old Dominion University's Virginia Modeling, Analysis, and Simulation Center and Assistant director at the University of Central Florida's Institute for Simulation and Training. He received a Ph.D. in computer science from the University of Central Florida in 1997. Dr. Petty has worked in modeling and simulation (M&S) research and education since 1990 in areas that include verification and validation methods, simulation interoperability and composability, human behavior modeling, and applications of theory to simulation. He has published over 180 research papers and has been awarded over $15 million in research funding. He served on a National Research Council committee on M&S, is a certified M&S professional, and is an editor of the journal *SIMULATION*. He has served as dissertation advisor to four graduated Ph.D. students, including the first two students to receive Ph.D.'s in M&S at Old Dominion University and the first student to receive a Ph.D. in M&S at UAH.

Acknowledgments

This paper was originally published as 12-SIW-25 in Proceedings of the 2012 Spring Simulation Interoperability Workshop (SIW). Copyright 2012, by the Simulation Interoperability Standards Organization, Inc.

This article has been corrected and enhanced by the original author for this publication, and adjusted to conform with the *M&S Journal* guidelines and format.

Augmented Reality Training Application for C-130 Aircrew Training System

AUTHORS

Dr. Charles R. Mayberry
HQ Air Education and Training Command / A3ZM
Randolph AFB, TX
charles.mayberry@us.af.mil

Ms. Sheila Jaszlics
Pathfinder Systems, Inc.
Arvada, CO
sheila@pathfindersystems.com

Mr. Gary Stottlemyer
Pathfinder Systems, Inc.
Highlands Ranch, CO
pittsmans1s@yahoo.com

Mr. Garrett Fritz
Pathfinder Systems, Inc.
Arvada, CO
Garrett.fritz@pathfindersystems.com

ABSTRACT

THE U.S. AIR FORCE (USAF) TRAINS C-130H LOADMASTER STUDENTS AT LITTLE ROCK AIR FORCE BASE (AFB), ARKANSAS THROUGH A CIVILIAN CONTRACT. THE AIRCREW TRAINING SYSTEM (ATS) CONTRACTOR UTILIZES A FUSELAGE TRAINER (FUT) TO PROVIDE SCENARIOS FOR THE LOADMASTER STUDENTS TO PRACTICE LOADING AND UNLOADING A SIMULATED AIRCRAFT. THE PROBLEM WAS THAT USAF DID NOT HAVE ENOUGH TRAINING DEVICES, NOR WERE THESE DEVICES AT A SUFFICIENTLY HIGH FIDELITY TO ACCOMPLISH MANY OF THE AIRCRAFT FUNCTIONS NECESSARY TO MEET THE TRAINING OBJECTIVES BEFORE FLYING ON THE ACTUAL AIRCRAFT. THE ATS MOVED THE PILOT'S INITIAL TRAINING INTO THE WEAPON SYSTEM TRAINER (WST). WST HAS NEARLY ELIMINATED ALL THE AIRCRAFT FLIGHTS MADE FOR PILOT INITIAL INSTRUMENT TRAINING BECAUSE THE SIMULATOR IS LIFE-LIKE ENOUGH TO ACCOMPLISH THE TRAINING TASKS TO QUALIFY THE STUDENTS IN THE DEVICE. SINCE LOADMASTER STUDENT FLIGHTS ARE SCHEDULED BASED UPON THE PILOT'S FLIGHT TRAINING, LOADMASTER STUDENTS ARE FORCED TO UTILIZE SOME OTHER TYPE OF SIMULATOR DEVICE FOR THEIR INITIAL TRAINING.

THE GOAL WAS TO INVESTIGATE AN EFFICIENT AND EFFECTIVE AUGMENTED REALITY (AR) TRAINING SYSTEM TO INSTRUCT LOADMASTER SKILLS BEFORE THEY TRAIN ON THE AIRCRAFT. THE INVESTIGATION EXAMINED THE USE OF A PROTOTYPE HELMET MOUNTED DISPLAY (HMD) AR DEVICE ATTACHED TO A LOADMASTER'S HELMET. THREE SCENARIOS PROVIDED A BASIS TO EVALUATE THE DIFFERENT ASPECTS OF HARDWARE AND SOFTWARE NEEDED TO UTILIZE AN HMD AS A LOADMASTER TRAINING TOOL. THE SCENARIOS TESTED HOW THE AR DEVICE MAY IMPROVE THE C-130H LOADMASTER TRAINING CAPABILITIES TO LEARN NORMAL AND EMERGENCY PROCEDURES TO STUDENTS IN THE FUT. THE RESULTS DEMONSTRATE A WAY THE GOVERNMENT CAN SAVE THOUSANDS OF DOLLARS IN FUEL COSTS AND OPEN THE EYES OF THE TRAINING CONTRACTOR TO A NEW WAY OF TRAINING STUDENTS USING AR.

INTRODUCTION

There is a growing need to train airmen with simulation in the way we have been training pilots for years. This paper examines the potential benefits of using a prototype Augmented Reality (AR) tool to train U.S. Air Force (USAF) loadmaster personnel in C-130 aircraft flying events. This case study research used a mixed methods research design that includes surveys and interviews to collect quantitative and qualitative data [1], [2]. The questionnaires were based on Kirkpatrick's four levels of evaluating a training program

[3]. Kirkpatrick's methods served to answer some of the research questions in evaluating a new training tool for loadmaster instructors and compare the learning outcome of the students who used the tool with students who were not exposed to AR [3]. But first, an introduction is needed to understand where loadmasters work and how they train in the C-130 world.

BACKGROUND

The C-130 Hercules is an aircraft capable of delivering cargo on a short dirt runway, in a hostile area, at night, and with no visible lights on the field. It is a high wing, four-engine, propeller driven cargo aircraft, flown with a crew of five: an aircraft commander, pilot, navigator, flight engineer, and a loadmaster. Loadmasters are the cargo handling and rigging experts on the aircraft. They are responsible for loading and unloading the cargo, rigging the parachutes for airdrop missions, preparing Army troops for personnel airdrop missions, and are charged with the safety and security of the cargo compartment.

USAF trains C-130 students at Little Rock AFB, Arkansas, through a government-funded civilian contract. The civilian contractors provide instruction on the academic and simulator portions of the curriculum in accordance with the Aircrew Training System (ATS) contract guidelines. The current ATS contractor, Lockheed Martin Global Training and Support (LMGTS), is also tasked with maintaining a variety of training devices used in teaching each of the crew positions. Students do not receive any flying skill credit for training in the lower-level non-integrated type devices. The C-130 weapon system trainers (WSTs) do allow flying skill credit for certain crew positions when training specific maneuvers in this device [4]. In fact, some of the emergency procedures practiced in the simulator are not performed on the aircraft or in operational training [5]. Many of the C-130 training devices are geared toward pilot training, but over the last few years more effort has been made to develop training devices for the remainder of the crew, especially loadmasters whose access to training flights is now extremely limited due to the prevalent use of WSTs to replace flight hours.

To support loadmaster training, the USAF took four older C-130 aircraft, removed their wings, stripped the tails off down to the fuselage, and permanently mounted the aircraft in a hangar. These fuselage trainers (FuTs) provide scenarios for loadmaster students to practice various cargo configurations in an actual aircraft. LMGTS instructors currently use the four FuTs to train loadmaster procedures for loading and unloading the aircraft, rigging procedures for airdrop missions and aircraft emergency procedures [6]. Some loadmaster emergency procedures do not lend themselves to full motion simulation, as the WST does for the pilots, or to real-life aircraft scenarios. For example, the USAF frowns upon starting fires in a training aircraft just for practice; therefore, we explored the feasibility of an alternative training tool that uses AR to integrate an image of these dangerous conditions to support training and incorporate instructional strategies that differ from traditional loadmaster training devices.

PROBLEM STATEMENT

Loadmaster training for operational procedures is deficient in providing a platform to familiarize students with each flying training event they are required to perform before they start performing these procedures on the job [7], [8]. In the C-130 FuT, training is limited to procedures that do not involve a reaction from the aircraft. For example, there is no process for practicing engine starts, extinguishing a fire in the cargo compartment, viewing cargo extractions or to dealing with various malfunctions that occur in the cargo compartment. Loadmaster students still require aircraft flights to finish their initial training, unlike pilots, who have had the majority of their initial training moved to the WST [9], [10]. Pilot WST sorties have nearly eliminated all the aircraft flights for initial instrument training because the simulator is life-like enough to accomplish the flying training tasks in the device [11]. This reality has forced the loadmasters to search for ways to achieve their required training with fewer aircraft flights [6]. The loadmaster's flying training schedules are based solely upon the number of sorties a student pilot receives during their initial training [12].

Unlike the WST, the FuT does not move or have any external visual systems to simulate flight. The ATS is now forced to utilize additional devices to meet this initial training gap. Stewart et al. [5] demonstrated that low cost simulators can be an effective training tool when appropriate training strategies are employed. USAF does not currently have enough fuselage training devices nor do these devices possess a sufficiently high enough fidelity to train critical, safety of flight, objectives before flying on the actual aircraft [12]. USAF decided to investigate the integration of an AR device into the current FuTs to increase its availability and fidelity. USAF is also using a virtual reality (VR) device to overcome some of the costs and training environment limitations for loadmaster training associated with tasks outside the aircraft, such as engine start skills. Stewart et al. [5] suggested that skills learned in lower-level training environments will transfer to a higher-fidelity environment such as the aircraft. This study intends to validate if this skills transfer can be accomplished for loadmaster students.

GOAL

The goal of this effort was to put into place an efficient and effective AR training system to instruct loadmaster students in Crew Resource Management (CRM) skills, during critical times on the ground or in-flight, before they train on the aircraft. The efficiency of the training device will enable students to quickly acquire a higher level of proficiency than they would otherwise achieve [13]. The CRM skills include situational awareness, crew coordination, communications and task management, which are all incorporated when dealing with operational and emergency procedures on the aircraft [14]. Situational or spatial awareness gives the student the cognitive ability to be aware of his location in space both statically and dynamically [8]. Training in an actual aircraft fuselage for this physically demanding job further helps transition students to the real aircraft, as they learn where to stand, kneel, etc. during each mission.

The lack of available aircraft flights to instruct loadmaster students in CRM skills drove a requirement to investigate an alternate method to train them, while maintaining the same high quality of student knowledge and skills. Air Education and Training Command (AETC) developed a prototype system that combined AR with the physical reality of a C-130 fuselage [15]. The AR C-130 Loadmaster Trainer (ARCLT) system was developed and tested in a small group try-out (SGTO) at Little Rock AFB from March through June 2008 [7], [16], [17]. The SGTO determined whether the instruction was appropriate for the average student target, which led to the conclusion that the ARCLT may be feasible as a training tool for C-130 loadmaster instruction. Following the guidelines in the USAF Instructional System Design process, lessons learned from the SGTO were applied to the system upgrades: more cameras were added, better software installed, and more capable goggles were used during the large group try-out (LGTO). AETC conducted a study in the LGTO format using the ARCLT to evaluate the training methodology to ensure that the usability goal of an efficient and effective training system is met [17], [18]. The ARCLT allows the trainee to utilize the same equipment used on the aircraft. This type of simulation has great potential for training procedural tasks, especially emergency procedures, which require a realistic haptic feedback during the training [19].

Research Questions

The overall research questions that helped guide the study included the following:

1. Why are computer-based simulations insufficient for learning to master CRM skills needed by Loadmasters?
2. How can an AR device be added to the physical training site to complete the training process?
3. Based upon the initial evaluations of the prototype AR system, what adjustments were made to the hardware, software and to instructor scripts?
4. What lessons have been learned about the use of AR devices in training that will ascribe value to other training situations?

METHODOLOGY

The problem to be addressed is the difficulty encountered by USAF in training new loadmaster students on how to master operational procedures before actually performing them on the aircraft. The goal is to install and test a prototype AR training tool used in the classroom and mounted in a FuT to teach students CRM skills and flight procedures before being trained on the aircraft.

We used a mixed methods research design to collect and analyze quantitative and qualitative data to evaluate whether the ARCLT system is an efficient and effective tool to train Loadmaster students [1]. Quantitative data were drawn from surveys administered to the students and contract instructors. The qualitative data were derived from the interviews conducted with 21 students who used the AR device, 5 contract instructors who taught students on the AR equipped FuT, and 8 flight instructors who flew with these students. The flight instructor interview responses were compared to entries logged in their students' training records. A comparison can also be made with the students who were trained on the ARCLT to those who did not use the AR device [2].

RELATED WORK

Simulation in Training

Flying techniques and aircraft simulator innovations have improved training methodology by incorporating better flying training devices, which are now used more often than teaching certain procedures in the actual aircraft [10]. Some of the early flight simulators started out in a wooden box to capture the *feel* of the controls whenever the pilot made an input. The development of the Link Trainer, (figure 1) made it possible for students to sit in a wooden cockpit, shaped like a small aircraft, enabling them to feel how the aircraft reacts to the movement of the flight controls by actuating the stick and rudder pedals [20].

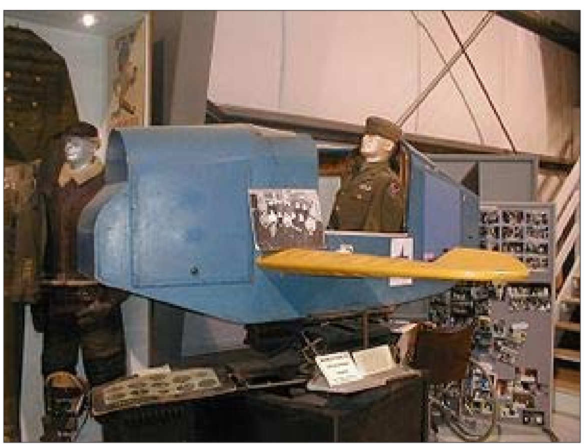


Figure 1: Link Trainer (Wikipedia 2011 [38])

Simulation has vastly improved from the wooden cockpits in the early days of flight, to the sophistication of full scale WSTs used to train USAF pilots. The ability to practice low level flight procedures in a training device enables the crew to better familiarize themselves with the mission, practice checklist procedures over and over until the steps are mastered, and practice instrument approaches into unfamiliar fields before venturing out to an actual site in a real aircraft [10], [8]. The capability to learn flight procedures in different types of simulation devices has gradually improved. Many of the improvements to the WSTs are due to advancements in computing technology, which have improved the *feel* of the motion and controls [21]. Most of the changes to the simulators have been implemented to benefit pilots, since their training is the most expensive. For example, an aircraft flight, such as a C-130, cost about $4,750 per hour, depending on the type of aircraft, whereas a simulator like the C-130 WST, costs approximately $700 per hour [9].

A variety of projection systems have been used over the past 20+ years to simulate views of the real world so that the students feel as though they are in the actual environment. Many aircraft weapon systems use WSTs to show virtual scenes projected onto a large screen in front of a simulated aircraft cockpit. The cockpit is fully populated with all the instrumentation found on the actual aircraft but is surrounded by a metal box and frame, which is mounted on six hydraulic legs to support full motion as shown in figure 2 [11]. The visual scene in the WST is limited in scope to the height and width of the screen itself and by the number of projectors tied together to display the virtual picture.

Figure 2: C-130 Weapon System Trainer

Students sit inside the simulated aircraft and view the virtual world through the cockpit windows. The WST enables students to practice a multitude of flight maneuvers replicating the actual view and feel of the real aircraft. Air Mobility Command (AMC) agreed with the research demonstrating that students using a virtual learning environment can achieve higher learning result and supported AETC in researching ways to lower the cost of training loadmasters through AR [22].

LEARNING CHARACTERISTICS OF SIMULATION

Simulation has been used as a training aid throughout many years of developing learning processes for teaching critical skills, such as aviation or surgery [23]. With the advent of faster and more mobile computer components, computer systems are becoming more ubiquitous in the training aids. The gaming industry has capitalized on new computer systems to promote not only entertainment style games, but the edutainment of today's youth as well [24]. Multimedia companies have made learning fun. Many of the games geared toward younger learners are made so that they achieve the next level in the game as they gain the knowledge needed to defeat the enemy on each level. The integration of educational computer software hidden in the games enables the student to acquire knowledge without knowing the gaming system is actually teaching them certain skills.

Incorporating a wearable computer allows the user to experience simulation on a personal basis. The ability to make simulation more mobile in training critical skills allows for ubiquitous computing in a training system. The U.S. Army has developed an integrated computer system used on fighting gear and weapons. Not only can the students see the virtual target through the scope of the rifle, but the device can be polled for physical conditions the student may encounter in the field [25]. Tracking the student, monitoring their condition, and providing realistic targets in a virtual setting make the student unaware of the wearable computers and the software integrated into the training environment.

Simulation is the imitation, to the extent feasibly possible, of actual conditions in which students can systemically explore different situations without the consequences of risking lives or destroying equipment; simulations attempt to represent the real world with some control over the situation but exclude some aspects of the real world [26]. Pilots receive much of their training through simulators; most of this is spent in extreme conditions [10]. A simulator allows students to greatly reduce the time required to learn these lessons without the consequences of real-life experiences [27]. In the case of loadmaster training, certain skills are required before they are turned over to their units to gain experience. The use of the ARLCT enabled testing on three scenarios used to teach loadmaster skills.

AUGMENTED REALITY TRAINING

To get an understanding of where AR fits into the realm of visual displays, many researchers use Milgram's Virtuality Continuum to show the contrasting ends of the scale [23]. Milgram uses a scale to indicate how AR falls between the physical real world (non-modeled reality) on one end and a completely virtual world (100% modeled reality) on the other, AR falls closer to the real world end of the scale [28]. AR is a method that combines a live view of a physical real-world environment with computer-generated sensory inputs, which is interactive in real time and registers in 3-D. AR is not restricted by display technologies, nor is it limited to the sense of sight; it can virtually remove or occlude real objects with virtual ones [23]. An example of Milgram's scale would show the real world as someone standing in a museum viewing the bone structure of a dinosaur; the AR view would show a prehistoric fish swimming around in the museum; and the fully virtual world would show the whole museum in a fully digital video game style display [28]. AR has been used in television broadcasts, such as the Summer Olympics, to superimpose countries' flags on swimming and track lanes and during football games to display a yellow line to indicate the first down line [29]. Just as virtual pictures can be broadcast on television, digital images can be projected through a device mounted on a helmet.

Research and development into new helmet mounted displays (HMDs) has been growing steadily over the last few years. AR technology has progressed substantially since the 1980s and 90s. The advent of smaller computer parts, the increase in the speed of the processors, and the ability to *wear* the computer has made it easier to incorporate HMDs into student training [30]. Rockwell Collins developed the SimEye™ series of HMDs, enabling USAF F-35 pilots to see out the window with a 40 X 30 degree field of view (FOV) [31]. HMDs provide the user with the ability to access graphical information immediately, since the view is directly in front of their eyes [30].

Although HMDs have grown steadily, real time user tracking has become one of the main concerns in developing an AR system. Several different tracking approaches have been used for various purposes, but there has not been a *standard* set for tracking [32]. Today portable computing is all around us—from smart phones to netbooks or tablets. These devices incorporate small computers that can use the global positioning system (GPS) to track the user's position. Geosynchronous satellites for GPS have made it possible to track the location of any mobile user with relatively low uncertainty [33]. Research into Wireless Local Area Networks, Ultra-Wide Band, and indoor GPS demonstrates each of the tracking methods have particular benefits and limitations, depending on the use of the device [33]. The ability to track where the student is in the training area, and the ability to know what the student sees, both in the virtual world as well as in the real world, helps an instructor to monitor the situational awareness of the students' perceived presence. It is important to note, the loadmaster student in our effort has determined that the position and orientation data available through GPS (even differential GPS) is not sufficient to provide a credible, aligned AR image within the FuT. Our prototype effort uses optical tracking, which can provide the position and orientation of the student's head within +/-5mm.

AR devices have been used across many disciplines to provide a way to practice procedures that may not otherwise be taught without involving human lives [21]. The military has simulated many of the aspects of training warfare into something that can be mastered before the student progresses to the field [8]. In creating an AR system, researchers often underestimate the effort required to incorporate real-world data into training applications. There has been much more research conducted on VR aspects of training than on AR. There are advantages and disadvantages for both VR and AR applications. Botden et al. [34] point out that in laparoscopic simulation, the advantage of AR over a straight VR device is that it allows the user to utilize the same working environment used in an operational setting, which is absent in a VR setting.

LOADMASTER SCENARIOS

In following with training in the working environment, three scenarios were chosen for testing the AR system for their visual experience: engine start, combat offload, and heavy equipment airdrop. Each scenario is part of the loadmaster duties taught on the aircraft. The engine start scenario was found to be taught best in the classroom, while the other two scenarios work best in the fuselage trainer. Funding the ARCLT and contracted time limitations narrowed the study down to these three scenarios.

Engine Start

The first scenario enables the students to practice aircraft engine starts. During the SGTO, the engine start scenario was developed to be administered outside, forcing the student to assume the correct position for each point in the engine start task [35]. The initial concept was to use an actual aircraft as a backdrop to align the virtual propellers and engines displayed in the AR goggles, but aircraft availability and the immaturity of the software forced the scenario to be redesigned using a full size virtual aircraft. This second attempt used fiducial placards (one foot metal squares painted bright green and orange) placed on the side of a hangar. These fiducials were used by a small camera mounted on the student's helmet to align the virtual aircraft in the student's view as he moved in front of the virtual aircraft to monitor each engine as it was started. After many trials of trying to provide the students with a stable platform that was easy to set up, the contract instructors suggested that the engine start scenario would work best in a fully virtual mode, projected on a screen in the classroom, as illustrated in figure 3, to give USAF the most "bang for the buck." During the LGTO, the engine start scenario was taught in the classroom with the instructor manipulating different events and emergencies programmed into the system [36].

Figure 3: Engine Start Scenario

Combat Offload

The second scenario was chosen to replicate the extreme urgency with which the crew is required to perform this task. This scenario is performed in a FuT that has been augmented with the ARCLT device. The scenario enables the students to practice procedures for combat offloading palletized cargo. The C-130 is capable of delivering cargo onto the ground without the use of any type of unloading equipment, such as a forklift [37]. Hostile areas around the world require cargo to be delivered as quickly and efficiently as possible so that the crew spends minimal time on the ground. To avoid being exposed to any danger, the crew must land their aircraft, drop off the cargo, and take off again from an airfield, in as short a time as possible. This scenario provides the students with the ability to practice not only the normal procedures, but also emergency procedures associated with offloading cargo on the ground [36].

The combat offload scenario was set up to virtually show the aircraft on the ground through the AR goggles with engines running and the ramp and door open. In this scenario, the instructor has the option to have students practice reverse taxiing of the aircraft [36]. The student will direct the pilot to maneuver the aircraft to the right or left as he reverses the propellers to back the aircraft up to the offload point. Once at the designated drop-off point, the pilot pushes the throttle forward to tilt the aircraft in such a way as to roll the cargo out of the back of the aircraft and onto the flight line. The student not only directs the pilot in the procedure, but can also see the result of the effort. When the virtual cargo is dropped off, the ramp and door are shut and the crew steps through the rest of the checklists to prepare for departure [37]. All of this is possible by tracking the student's location in the FuT. We chose the optical camera in the ARCLT to give us the most accurate tracking in a small area of the FuT.

Heavy Equipment Air Drop

The third scenario involves dropping heavy equipment from 800 feet above the ground, enabling the cargo to land close to the calculated point of impact. Many students are *in awe* of the event in the aircraft. As a result, the airdrop scenario was performed in the FuT, augmented with the ARCLT device to help familiarize students with the procedure and to hopefully reduce the *awe* factor. This scenario, depicted in figure 4, represents cargo being airdropped out of the back of the C-130 cargo compartment, with the ramp and door open as the aircraft simulates flying over a drop zone [36].

FuT Heavy Equipment Air Drop

Figure 4: C-130 Cargo Compartment

The student will prepare the actual cargo for extraction in the FuT, ensuring that the parachutes are configured and connected properly. A 20-minute advisory is heard from the navigator as the aircraft approaches the drop zone. All of the checklists are run (called out) with the recorded voices of the crewmembers as the loadmaster responds with the proper calls. When the 1-minute advisory is called out, the loadmaster kneels down at the pallet lock release lever, preparing to pull the handle to release the pallet. At the *green light* call, the loadmaster sees the green light illuminate in the cargo compartment through his goggles, sees the virtual drogue parachute released from the bomb rack, and opens up to pull the cargo out. Once the parachute opens up and the locks are released by pulling the release lever, the virtual cargo is swiftly pulled out on the rollers attached to the floor of the cargo compartment with a loud rushing noise.

It is at this stage where students sometime forget where they are with the checklists because of the excitement of the event. Afterwards, the other checklists are run to clean up the aircraft, the virtual ramp and door are closed, and the aircraft escapes off the drop zone [37]. At any point during the scenario, the instructor has the ability to pause or restart it in order to point out or emphasize certain items, or to practice certain procedures repeatedly. An excellent learning characteristic of the airdrop simulation is the ability to introduce emergency procedures during the scenario. Not only can the student be trained to recognize normal procedures, but can also practice emergency situations not normally seen during actual flight training [13].

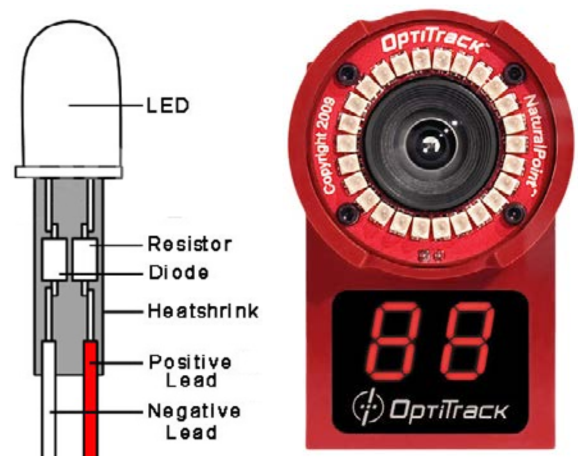
Technical Design

In order to keep up with each of the students in the FuT, the tracking in the ARCLT is accomplished via 38 cameras, which detect tracking markers located on top of the students' helmets. Each camera sends a real-time image to specialized software. The software analyzes the multiple camera images to compute (triangulate) the tracking marker's location and orientation relative to the cameras.

In the SGTO, reflective markers were used on top of the students' helmets in combination with infrared detecting tracking cameras. These cameras emitted infrared light via banks of Light-Emitting Diodes (LED's). Thus, each camera would not only detect the markers, but would also act like an infrared LED flashlight, illuminating the markers while they were in the camera's Field of View (FOV). Unfortunately, this approach resulted in the FuT being flooded with infrared light, which reflected off shiny metal parts and lightly colored surfaces, creating *infrared noise*. The amount of noise varied within the FuT, but in certain locations the noise made it impossible for the cameras to detect or differentiate the tracking markers, resulting in a loss of tracking in that area.

For the LGTO, the reflective tracking markers were replaced with infrared emitting LEDs on top of the students' helmets, and the camera LEDs were simply turned off, (figure 5). This eliminated the *infrared noise* and has greatly improved tracking consistency throughout the FuT.

Since tracking cameras visually see, and therefore, can track within a given area, this area is commonly referred to as a tracking volume. A minimum of three cameras are required to form a usable volume; however, four is more common because it creates a cube-shaped tracking volume.



LED Light OptiTrack Camera

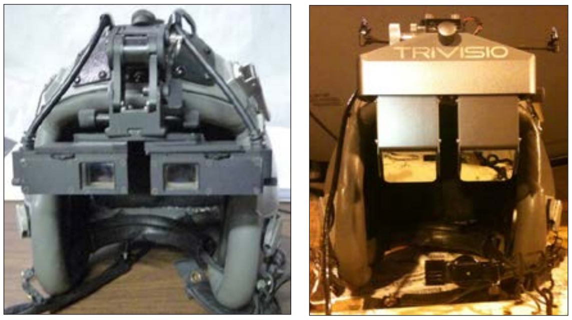
Figure 5: LED Light and OptiTrack Camera

During most tracking applications, the cameras are placed outside the volume, with the distance being optimized for the physical size of the volume. When a camera is farther away from the area to be tracked, more of the area will fall within the camera's FOV. However, the closer a camera is to the tracking markers, the greater the tracking accuracy. Thus, cameras are typically placed about five feet *outside* the desired tracking volume in order to balance both camera coverage and tracking accuracy.

For the LGTO, the initial HMD units were replaced with new units that only slightly restrict peripheral vision in the upward/vertical direction. The display FOV is only slightly increased (to 29 degrees, diagonally) but the elimination of the *tunnel vision* effect created a superior immersive visual experience, (figure 6). Since ARCLT utilizes AR, where most of what is seen is *reality*, the unobstructed FOV allowed the students to see *reality* in a near-normal manner, with the AR imagery appearing in the center of their vision.

Data Collection

The initial plan was to integrate the AR scenarios into the core curriculum for the course after the students were familiar with the procedures in the classroom. Due to contract limitations from the *guaranteed student clause,* the scenarios had to be conducted outside of normal class hours. This resulted in some of the students not being familiar with the checklist before going through the combat offload or the airdrop scenarios.



Liteye – FOV 24° @ 640 x 480, VESA (VGA/SVGA)

Trivosio - 29° diagonal (4:3, 23° (horiz), 17° (vert)), DVI-D

Figure 6: Old and New HMDs

The surveys were filled out after each student went through the scenarios. The quantitative data for the survey questions are built on a six-point Likert scale, with a not applicable option as the seventh button. The limit on the scale provided a dividing line between those who agreed and those who disagreed with the statements on the surveys. This data was compared to interview data from participants that had a positive or a negative reaction to the questions. The C-130 schoolhouse surveys students multiple times throughout the course of training to see how well the instruction is going. Students sometimes get tired of filling out surveys and promptly go down the center of the survey form to quickly finish the task on a five-point scale.

The category sections of the surveys were based on Kirkpatrick's model [3] for reaction to new hardware, fit and function of the device, the learning aspect in the different scenarios, and the behavioral change the instructors noticed in training with the AR tool.

The interviews conducted by the principal investigator used the established interview questions based on Kirkpatrick's model [3]. The same basic questions were asked as the survey questions, but were in an open-ended format. The interviews were recorded and transcribed to a spreadsheet. Once the categories were defined, a comparison of the data was made to the quantitative data from the survey questions. An interpretation was made at that point to see if the AR device was an effective training tool.

CONCLUSION

This case study evaluated the use of an augmented reality training tool to teach loadmaster objectives to new students on the C-130 aircraft. A mixed methods research design captured quantitative and qualitative data and equally compare and interpret the data to see if AR is an efficient and effective training tool for loadmaster students. The results do show that an AR device can be an effective training tool for teaching loadmaster procedures.

To get an idea of how the system was evaluated, a few of the categories from the statements in the surveys and the questions from the interviews are included in this article, but not all the data from the study.

Since the student surveys produced a relatively large sample size, about 50 surveys, the assumption was that the population followed a normal distribution with a reasonable estimate on the population standard deviation (SD). Because of this, a Z-Score was used to calculate whether or not to reject a null hypothesis of: *The students generally disagree with the question*. One example in the category of fit and function, the results indicated for the statement that at a 95% confidence level, and a threshold of 1.64 (depicted by the red shaded area in figure 7), the data does not show that the students agree that the *AR goggles fit well on the helmet;* 1.22 for airdrop, and 1.22 for combat offload depicted by the green value line shown in figure 7.

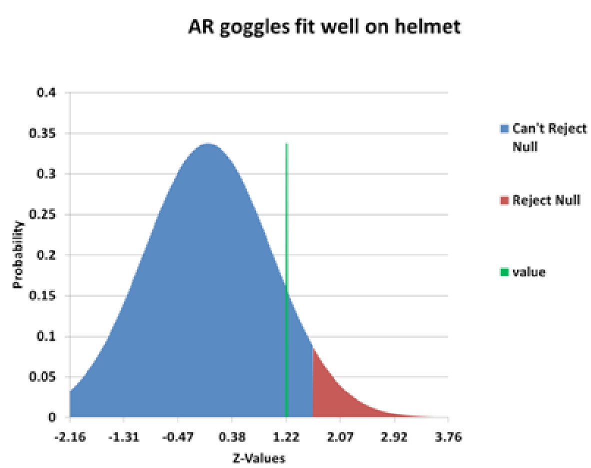


Figure 7: Fit and Function Chart

The follow-on statements used in the surveys were calculated in the same manner, that *the device was comfortable to wear;* Z-Values of .98 for airdrop and .81 for combat offload cannot reject the null; or that *the system ran smooth during the scenario;* .79 for airdrop and .62 for combat offload also showed that the data could not reject the null hypothesis. Comparing the data from the students' interviews about the fit and function of the device using the airdrop and combat offload scenarios showed that 70% vs. 64% of the students had a positive reaction to *the goggles fitting well on the helmet*, 70% vs. 36% were positive to *the device being comfortable to wear*, and 40% vs. 50% were positive toward *the system running smoothly*, showing that the low percentages agreed with the survey results.

In the instructor performance areas, the students were very confident in the instructor's knowledge and performance in running the system. Student surveys showed a Z-Score of 2.87 for airdrop and combat offload as illustrated in figure 8. Compared to the interview data of 100% for airdrop and 91% for combat offload, the students thought *the instructors were knowledgeable about how to train with the AR system.*

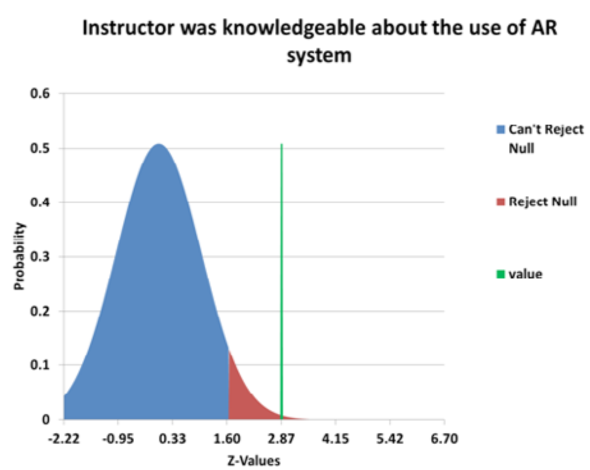


Figure 8: Instructor Performance Chart

The contract instructor sample size was not large enough to estimate the population SD, so a T-Test was used to evaluate whether or not to reject the null hypothesis of, *the instructors generally disagree with the question*. A critical t-value (.05) of -1.74 was used to evaluate the data. The following areas showed evidence (t-value) to reject the null hypothesis: *instructors were given adequate instruction to use the AR system,* -2.77; *the AR system provided a realistic portrayal of the actual events,* -2.13; *the AR system allowed my students to achieve a higher level of proficiency than students in the past,* -1.84; *and the AR system made my instructional time more productive,* 1.84. The interview data agreed with a couple of the statements from the surveys. The interview data showed 100% for engine start, and 50% for airdrop, had a positive reaction that the system was easy to run; 80% for engine start and 100% for airdrop showed that the graphics portrayed a realistic view of the events. Other areas that had a positive response included: the scenarios helped train the lesson objectives better than the current training, 80% for engine start and 67% for airdrop; the students retained more of the lesson objectives, 80% for engine start and 67% for airdrop. Some of the differences may be due to the fact that the surveys were completed early in the study, whereas the interviews were conducted near the end.

Statistically the data does not show an overwhelming agreement for many of the categories evaluated in the way the AR system was presented to the students and instructors. But, what was invaluable were the comments during the interviews that brought out some of the improvements that need to take place in order to create an acceptable production model. It is important to know, when asking a contractor for a response to a request for proposal, what the capabilities should be for an AR training device, know the limitations of the current technology, and be aware of what the millennial students are willing to accept as a realistic training device.

There are more in-depth questions to help analyze the data and narrow down the responses for the use of an AR training tool to show the effectiveness and efficiency of the system, and to answer the overall research questions. Limited space for this article only allows for a sample of what was used.

This LGTO is a primary step in evaluating such a drastic change in enlisted training. Knowledge gained from this study will hopefully spark interest in other training devices for enlisted crewmembers training on cargo aircraft. The fundamental practice of simulated training for pilots has overshadowed enlisted training for many years. The lack of funding has limited upgrades to Loadmaster training devices. This research shows how enlisted simulation devices will benefit the student's learning ability before flying on the actual aircraft. New technology has brought greater insight into building a relatively inexpensive device that can track a user in closed-in spaces. The next step should be to incorporate lighter, wider field of view glasses, not goggles, which can be tracked using fewer cameras with software to integrate smaller areas accurately.

FUTURE WORK

Augmented reality incorporates many different techniques to show the virtual world for training scenarios. Future scenarios for loadmaster training may include the use of avatars to portray U.S. Army paratroopers positioned in the cargo compartment of the C-130 ready to jump out over a hostile area. The emergency procedure for retrieving a hung paratrooper outside the aircraft could be practiced using the actual retrieval system installed in the FuT. This emergency procedure is never practiced and only a few aviators have actually witnessed the event.

LESSONS LEARNED

The camera placement constraints within the FuT limited each camera's area of coverage, requiring the use of more cameras and a carefully designed camera layout with multiple *volumes*, which had to be synthetically merged using client-server network software architecture. Furthermore, during and immediately after site deployment, it became apparent that certain cameras had to be moved, because their placement interfered with existing training being conducted in the FuT. To compensate for the relocation of these cameras, the entire camera layout, and the synthetic merging algorithms, had to be reviewed and modified.

HMD's suitable for displaying AR differ from HMD's suitable for displaying VR. In particular, an unconstrained vs. constrained FOV is highly preferable for AR. While the HMD most widely used for current military training applications is suitable for VR, it did not prove to be the best choice for ARCLT due to its lateral FOV being severely constrained.

Finally, the limitation of AR imagery being centered in the field of view may actually aide certain training cases where the students must be trained to scan for warnings or threats. For example, vehicle operators are often trained to *scan* their instrument panels rather than rely on *noticing* a flashing red light. Similarly, pilots are trained to *scan* for other aircraft and must not rely on *noticing* such aircraft in their peripheral vision. This is imperative because peripheral vision primarily *sees* motion, but aircraft on a collision course generally have little or no proper motion across the pilot's field of view (until it is too late to take evasive action).

ACKNOWLEDGMENTS

This study was sponsored by Air Education and Training Command Directorate of Operations (AETC/A3). We would like to thank Lockheed Martin Global Training and Support for their participation in the study and Air Mobility Command Training Division (AMC/A3T) for the use of the C-130 ATS schoolhouse.

References

[1] J. Creswell and V. Plano-Clark *Designing and Conducting Mixed Methods Research*. Thousand Oaks, CA: Sage Publications, 2007.

[2] R. Yin, *Case study research design and methods,* 4th ed. Thousand Oaks, CA: Sage, 2009.

[3] D. Kirkpatrick, and J. Kirkpatrick, *Evaluating training programs: The four levels, 3rd ed.* San Francisco CA: Berrett-Koehler, 2006.

[4] HQ AMC/A3TA. (2010). *Flying operations: C-130 aircrew training* (Air Force Instruction 11-2C-130 Volume 1) [Online]. Available: http//:www.e-publishing.af.mil.

[5] J. Stewart *et al*., "Fidelity requirements for Army aviation training devices: Issues and answers," U.S. Army Research Inst. for the Behavioral and Social Sciences, Ft. Rucker, AL, Research Report 1887, May 2008. Available: www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA493584

[6] S. Desnoyer, *C-130E loadmaster initial/mission qualification syllabus* (C-130ELMQ) Headquarters Air Education and Training Command/Directorate of Operations [Online]. Available: https://www.my.af.mil/afknprod/ASPs/docman/DOCMain.asp?Tab=0&FolderID=AE-ED-00-55-2-12-1&Filter=AE-ED-00-55

[7] M. Gardley,"Loadmaster training utilizing augmented reality," Final Report ETTAP Project 06-16, Randolph AFB, TX, 2008.

[8] R. Stone *et al*., "SubSafe: A games-based training system for submarine safety and spatial awareness (Part 1)," *Virtual Reality* vol. 13, pp. 3-12, 2009.

[9] G. Jean, G. (2009, June). "Training for war: A multimedia experience," *National Defense Magazine* [Online]. Available: http://www.nationaldefensemagazine.org/archive/2009/June/Pages/TrainingForWarAMultimediaExperience.aspx

[10] R. Mayberry, R. HQ AETC/A3ZM (2010). *C-130E Pilot initial qualification syllabus* (C-130PIQ) Headquarters Air Education and Training Command/Directorate of Operations [Online]. Available: https://trss3.randolph.af.mil/bookstore/HyperLinks/syllabus/C-130E%20PIQ%2012%20Oct%2010.pdf

[11] W. White. (1991). Airplane simulator qualification, *Advisory Circular* 120-40B, July 29, 1991 [Online]. Available: http://rgl.faa.gov/Regulatory_and_Guidance_Library/rgAdvisoryCircular.nsf/0/5B7322950DD10F6B862569BA006F60AA?OpenDocument&Highlight=120-40

[12] HQ AETC/A3RA. (2012). *Program flying training document* (Little Rock AFB FY 2012 Volume 1) [Online]. Available: https://www.my.af.mil/afknprod/ASPs/docman/DOCMain.asp?Tab=0&FolderID=OO-OP-AE-57-3-6-3&Filter=OO-OP-AE-57

[13] J. Fulbrook, J. Ruffner, and R. Labbe, "Near-to-eye display solutions for improving Air Force air traffic tower controller capabilities," Wright-Patterson AFB, OH, Sponsor Report Number AFRL-RH-WP-TR 2008-0028, Final Rep., 2008.

[14] HQ AETC/DOFV. (2012). *Flying operations: cockpit/crew resource management training program* (Air Force Instruction 11-290, Air Education and Training Command Supplement 1) [Online]. Available: http://www.e-publishing.af.mil.

[15] S. Jaszlics, "Augmented Reality C-130 Loadmaster Trainer (ARCLT)," Contract Number NBCH090076, Dec. 10, 2009.

[16] L. Gay, G. Mills, and P. Airasian, *Educational Research: Competencies for Analysis and Applications,* 10th ed. Upper Saddle River, NJ: Pearson/Merrill Prentice Hall, 2011.

[17] G. Twogood. (2002). Information for designers of instructional systems: Application to education. *Air Force Handbook 36-2235, Vol. 10* [Online]. Available: www.e-publishing.af.mil.

[18] M. Molenda, "Origins and evolution of instructional systems design," in *Handbook of Improving Performance in the Workplace, Volume 1: Instructional Design and Training Delivery*, K. H. Silber and W. R. Foshay, Eds. 2010, San Francisco, CA: Wiley, 2010, pp. 53-92. doi:10.1002/9780470587089.ch3

[19] E. Westebring-van der Putten et al., "The effect of augmented feedback on grasp," *IEEE Trans. Haptics,* vol. 3, pp. 280-291, Oct.-Dec. 2010. doi: 10.1109/ToH.2010.23

[20] J. Killgore, "The Planes that never leave the ground," *American Heritage of Invention & Technology*, vol. 4, no. 3, pp. 56-63, 1989.

[21] E. Samset *et al*., "Augmented reality in surgical procedures," in *Proc. SPIE*, 6806, Human Vision and Electronic Imaging XIII, San Jose, CA, 2008 [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=810977

[22] M. Vilkoniene, "Influence of augmented reality technology upon pupils' knowledge about human digestive system: the results of the experiment," *US-China Education Review,* vol. 6, no. 1, pp. 36-43, 2009.

[23] D. van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *Int. J. Virtual Reality,* vol. 9, no. 2, pp. 1-20, 2010.

References (Continued)

[24] S. Bloom, "Game-based learning using video game design for safety training," *Professional Safety*, vol. 54, no. 7, pp. 18-21, 2009.

[25] K. Waller, "Developing a benchmark suite for the evaluation of orientation sensors," M.S. thesis, Dept. Comp. Eng., Clemson Univ., SC, 2006. Available: http://www.parl.clemson.edu/~ahoover/theses/waller-thesis.pdf

[26] Y. Dahl, "Seeking a theoretical foundation for design of in sitro usability assessments," in *Proc. 6th Nordic Conf. Human-computer Interaction: Extending Boundaries,* New York, NY: ACM, 2010, pp. 623-626. doi: 10.1145/1868914.1868989.

[27] R. Oliva and M. Bean, "Developing operational understanding of service quality through a simulation environment," *Int. J. Serv. Ind. Man.*, vol. 19, no. 2, 2008, pp. 160-175.

[28] P. Milgram and A. F. Kishino, "Taxonomy of mixed reality visual displays," *IEICE Trans. Information and Systems, E77-D,* vol. 12, pp. 1321-1329, 1994.

[29] R. Cavallaro, M. Hybinette, M. White, and T. Balch, "Augmenting live broadcast sports with 3D tracking information," *IEEE Multimedia,* vol. 18, pp. 38-47, April 2011. doi: 10.1109/MMUL.2011.61. Available: http://www.cc.gatech.edu/~tucker/Papers/2011-multimedia.pdf

[30] H. Lee, D. Kim, and W. Woo, "Graphical menus using a mobile phone for wearable AR systems," *Proc. 2011 Int. Symp. Ubiquitous Virtual Reality,* Jeju-si, Korea (South), 2011, pp. 55-58. Available: http://www.informatik.uni-trier.de/~ley/db/conf/isuvr/isuvr2011.html

[31] M. Browne, K. Moffitt, and M. Winterbottom, "Improving the utility of a binocular HMD in a faceted flight simulator," in *I/ITSEC 2009 Proc. Interservice/Industry Training, Simulation and Education Conf.*, Orlando, FL, 2009, pp. 2-11.

[32] M. Eissele, M. Kreiser, and T. Ertl, "Context-controlled flow visualization in augmented reality," *Proc. Graphics Interface Conf.,* Windsor, ON: CHCCS/SCDHM, 2008, pp. 89-96.

[33] C. Woodward *et al.*, "Mixed reality for mobile construction site," *10th Int. Conf. Construction Applications Virtual Reality,* Sendai, Japan, 2010, pp. 35-44. Available: http://www.researchgate.net/publication/48330620_Mixed_reality_for_mobile_construction_site_visualization_and_communication

[34] S. Botden, I. Hingh, and J. Jakimowicz, "Suturing training in augmented reality: Gaining proficiency in suturing skills faster," *Surgical Endoscopy*, vol. 23, no. 9, pp. 2131-2137, 2008.

[35] S. Jaszlics, "Augmented Reality C-130 Loadmaster Trainer (ARCLT)," Contract Number FA3089-08-P-0078, Feb. 6, 2008.

[36] S. Jaszlics, "Augmented Reality C-130 Loadmaster Trainer (ARCLT)," Contract Mod. Number NBCH090076/0001, Sept. 29, 2010.

[37] HQ AMC/A37V. (2010). *Flying operations: C-130 operations procedures* (Air Force Instruction 11-2C-130, Vol. 3) [Online]. Available: http//:www.e-publishing.af.mil.

[38] Wikipedia. (2011). *Link Trainer* [Online]. Available: http://en.wikipedia.org/wiki/Link_Trainer

Authors' Biographies

Dr. Charles R. Mayberry

Dr. Charles R. Mayberry has been a C-130 training program manager for Air Education and Training Command, Graduate Training Division, at Randolph AFB for the past thirteen years. During his active duty tour, as a flying crew member, he worked with the C-130 schoolhouse overseeing simulation and curriculum. He finished his Ph.D. program in the summer of 2013, Computing Technology in Education, with Nova Southeastern University specializing in on-line learning and augmented reality.

Ms. Sheila Jaszlics

Ms. Sheila Jaszlics is the president of Pathfinder Systems, Inc., an engineering firm that specializes in the development and manufacture of devices and technologies to support military training and command and control requirements. Ms. Jaszlics has been the program manager on a variety of research and development programs associated with the utilization of augmented reality (AR) to support a variety of training, maintenance and command and control tasks. She has been an advocate of AR since the early 1990s when her firm patented the first AR processes that addressed the real time occlusion of virtual entities injected into real world scenes (US patent 6,166,744).

Mr. Gary Stottlemyer

Mr. Gary Stottlemyer is a software engineer/consultant for Pathfinder Systems, Inc. He has worked extensively in the simulation technology field for the past 27 years, specializing in visual image generation, synthetic terrain, physics, aerodynamics, ballistics, artificial intelligence, complex visual interfaces, and simulation networking.

Mr. Garrett Fritz

Mr. Garrett Fritz is an aerospace engineer for Pathfinder Systems, Inc. (PSI) with a B.S. in aerospace engineering from the Massachusetts Institute of Technology. He has developed flight models, human-machine interfaces, and software packages. He is responsible for experimentation and data analysis for independent research and development at PSI.

Cloud Simulation Infrastructure – Delivering Simulation from the Cloud

AUTHORS

Dr. Michael R. Macedonia
Assistant Vice President for Research, University of Central Florida
Orlando, FL
Michael.macedonia@ucf.edu

Dr. Christina L. Bouwens
Chief Technologist
MSCI
Orlando, FL
Cbouwens@msci-us.com

Mr. James E. Shiflett
Program Strategist
Leidos
Orlando, FL
James.e.shiflett@leidos.com

ABSTRACT

THIS PAPER DISCUSSES THE PRACTICAL ASPECTS OF ARCHITECTING A SEMI-AUTOMATED FORCES (SAF) SYSTEM FOR A CLOUD COMPUTING ENVIRONMENT AND DESCRIBES SOME OF OUR RECENT EXPERIMENTS WITH SAF TECHNOLOGY IN THE CONTEXT OF A CLOUD-ENABLED ENVIRONMENT. THE RAPID TRANSITIONING OF TRADITIONAL COMPUTER APPLICATIONS SUCH AS EMAIL TO CLOUD COMPUTING IS BEGINNING TO EXTEND TO MILITARY SIMULATION. THE UBIQUITY OF THE GLOBAL INTERNET AND ADVANCES IN MOBILE EXERCISES HAS BEEN COMMON FOR OVER A DECADE; HOWEVER, THE MODEL HAS BEEN BASED ON SCHEDULED, DEDICATED, AND OFTEN TEMPORARY INFRASTRUCTURE. CLOUD SOLUTIONS OFFER THE POTENTIAL OF "ANYTIME, ANYWHERE," ON-DEMAND SIMULATION AND TRAINING CAPABILITIES. THE PRIMARY CHALLENGE HAS BEEN IN ARCHITECTING SIMULATIONS FOR VIRTUALIZATION AND PROVIDING THE REQUISITE SECURITY FOR MILITARY OPERATIONS. SOLUTIONS TO THESE PROBLEMS ARE BEING VIGOROUSLY ADDRESSED. THIS PAPER EXPLORES SOME POTENTIAL IMPLEMENTATIONS OF A CLOUD SIMULATION INFRASTRUCTURE (CSI) CONCEPT—HOW A SIMULATIONS SYSTEM COULD BE HOSTED AND ACCESSED VIA THE CLOUD. ALTHOUGH NOT THE SAME AS CLOUD COMPUTING, HIGH PERFORMANCE COMPUTING (HPC) HAS SOME USEFUL SIMILARITIES TO CLOUD COMPUTING AND MAY OFFER AN ALTERNATIVE DELIVERY INFRASTRUCTURE FOR SIMULATION SERVICES. WE OFFER RESULTS FROM OUR WORK IN HPC AND SAF SYSTEMS AS A PARTIAL CONTRIBUTION TO UNDERSTANDING AND DEFINING THE CSI CONCEPT. IN ADDITION, WE PRESENT RESULTS FROM OUR WORK WITH A WEB-BASED INTERFACE FOR MANAGING AND DEPLOYING SAF RESOURCES. COMBINING THE RESULTS OF THESE TWO BODIES OF WORK, THE HPC AND THE WEB-BASED INTERFACE, WE HAVE DEVELOPED PROTOTYPICAL MODEL OF SAF COMPUTING IN THE CLOUD. FROM THIS VANTAGE POINT, WE ALSO EXAMINE THE BENEFITS OF THE CSI CONCEPT, SUCH AS UBIQUITOUS ACCESS, COMMON (ACROSS SERVICES) CONTENT, TECHNICAL AND OPERATIONAL STANDARDS FOR TRAINING, AND POTENTIAL FOR TACTICAL MISSION PLANNING.

IMPERATIVES FOR CLOUD COMPUTING

Cloud computing has received a significant amount of attention leading to large information technology (IT) initiatives at the corporate and government levels. The National Institute for Standards (NIST) defines cloud computing as:

> *A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of*

configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1].

It is now part of national and DoD IT Strategy by virtue of the "Cloud First" guidance developed by the chief information officer (CIO) for the United States. The guidance states:

To harness the benefits of cloud computing, we have instituted a Cloud First policy. This policy is intended to accelerate the pace at which the government will realize the value of cloud computing by requiring agencies to evaluate safe, secure cloud computing options before making any new investments [2].

These policies are coming about because the technical imperatives for cloud computing (both public and private) are very clear. Computing environments are becoming more asymmetric with the development of large, high-performance data centers with virtualized hardware connected via broadband networks such as 4G LTE to resource-limited, albeit ubiquitous, mobile devices such as the iPad® (Apple Inc.). These environments are creating new and exciting capabilities for both civilian and military purposes that we address in this paper.

These imperatives, and the emergence of military cloud environments discussed later in this paper, provide major new opportunities for modeling and simulation. In particular, we have demonstrated this potential with the OneSAF® (U.S. Department of the Army) entity-level resolution simulation [3].

Current Practices for Distributed Simulation

For over 25 years, the modeling and simulation community has been exploring the concept of distributed simulation and the interoperability of defense simulations beginning in January 1989 with the first workshop on interoperability standards. At that time, the Internet was only used by a small community (mostly academic) and there was no such thing as the World Wide Web.

Over the years, the distributed simulation community has expanded modeling and simulation (M&S) capabilities to include the following:

- Interoperability standards have been defined and are in use serving a broad range of users, from high-fidelity virtual simulations (distributed interactive simulations, (DIS); high-level architecture, (HLA) to faster than real time analysis applications (HLA) to support for live test and evaluation activities test and training enabling architecture, (TENA).
- Networks such as the Defense Research and Engineering Network (DREN) and the Joint Training and Experimentation Network (JTEN) have made coordinated training and testing events possible—linking locations across the country and the world.
- Gaming technology development has led to advances in graphical rendering of simulation environments, highly interactive immersive worlds, and an introduction to new applications for interactive distance learning and highly engaging training environments.

Much of the distributed, remote accessibility that the cloud community offers has already been enjoyed by the M&S community. However, in contrast to the direction of cloud computing, the M&S community has been *distributing* its resources, contrary to the cloud model that seeks to consolidate them.

Legacy Challenges for Distributed Simulation

Distributed simulation comes at a technical and operational price that limits it utility in everyday training and experimentation:

- "Fair fight" is difficult to guarantee in training and experimentation in a long-distance environment. Different latencies and computing resources are a direct result of the distributed model. Fair fight also become problematic even when similar simulations are run on a local area network.
- Current DIS and HLA models of simulation do not support persistence because they do not have a central store or control. Therefore, it is currently very difficult to have simulations of long-duration in a distributed environment.
- Distributed simulations generally do not support the use of handheld mobile devices. Handhelds such as Android® (Google Inc.) tablets have limited computing, memory, and battery resources that are quickly overwhelmed by simulation requirements. Therefore, tactical commanders generally cannot run intense simulations on their future command and control devices.

- Each participating M&S site has to maintain its own facilities and equipment in order to participate in exercises.
  - —This requires facility space, cooling, power, and computational hardware.
  - —Operating systems and software need to be installed and maintained.
  - —Resources are idle when not in use for an exercise—this can be a substantial amount of time.
  - —People are required to maintain the environment including regular maintenance, hardware, and software upgrades—and tracking how upgrading or not upgrading affects interoperability with various participating systems.
  - —Significant time and expense is expended retooling/reconfiguring existing hardware for different exercise events.
- Set up for a particular distributed simulation exercise can take months for coordination and weeks on the ground at various sites for installing and integrating participating simulation systems.
  - —Costs are incurred for engineers to travel to site for exercise support.
  - —Integration and testing cannot take place until all are available on site.
  - —A team of operators is required to support the execution of an exercise. They must be available ahead of and during the exercise and sit idle if the exercise goes down.

RATIONALE FOR MODELING AND SIMULATION IN THE CLOUD

The federal "Cloud First" strategy has led our research team to recognize the potential value of cloud computing concepts, and we are actively exploring the use of cloud computing. We are looking at overcoming the current challenges of distributed simulations by centralizing simulation resources and effectively delivering training and simulation services to a broad set of distributed users at both the enterprise and operational levels.

We also believe that a cloud simulation infrastructure would be more defensible in the context of a cyber challenge. Cyber Command has recommended DoD move to cloud-based architectures for its intelligence systems. General Alexander, Commander, U.S. Cyber Command, recently stated, "How do we create the next set of architecture that is more defensible and can ensure the integrity of our data? I think it's in the cloud [1]."

We believe this will lead DoD to utilize data centers at existing Army facilities to deliver secure, high-performance cloud-based simulation and gaming over Army networks. This approach simplifies a number of different issues related to utilizing live, virtual, constructive and gaming (LVCG) for training.

One of the largest of these issues is ensuring that soldiers have hardware capable of running the training at an acceptable frame rate. For example, gaming technology has historically driven the development of video cards with recent years seeing a doubling of relative graphics performance each year. Centralizing the processing of these video games in a data center greatly simplifies testing and deploying new hardware that enables the top-flight features of the latest games.

The Army will only have to upgrade servers at a relatively small set of data centers, and the benefit will seamlessly extend to all of the computers connected to the network. Another common issue is ensuring that soldiers have the most up-to-date training available. Training applications that are installed on a dispersed set of computers are much more difficult to upgrade than training applications that are installed at a relatively small set of data centers. Updating the training at a data center makes the latest version immediately available to everyone on the network without having to touch each individual computer.

Implications for Modeling and Simulation

Unlike typical IT needs, M&S applications tend to use the underlying virtualized hardware more extensively for prolonged periods of time. The applications have higher memory requirements, intensive central processing unit (CPU) usage, minimum CPU counts per node, multiple distributed nodes, and a low latency/high bandwidth network. During execution, the demand on the virtualized hardware will be at a sustained high load for significant portions of the simulation exercise.

Not all M&S applications will reside in the cloud. The integration of live, virtual, and constructive simulations along with command and control (C2) systems and other operational equipment requires M&S cloud implementations to allow for a mix of cloud and non-cloud resident applications.

With a fairly mature distributed simulation infrastructure, remote access with participation from individuals has been well established. M&S has solved the harder problem of how things actually connect, where cloud promises that things will work without ever defining how that happens.

THE BENEFITS OF CLOUD/ VIRTUALIZATION SUPPORT

Cloud seeks to consolidate resources, providing the ability to co-locate many of the "back-end" components and all within a common hardware infrastructure. The virtualization component of cloud reduces the need for hardware (and space and power). For example, 50 standard PC configurations for running a very large OneSAF exercise would require a substantial lab space for all the machines and monitors (chairs, etc.). The same exercise using virtualization takes a fraction of the space with 50 virtual machines on 50 (or less) cores and running on a server. Zero or thin clients are needed only for the participants or pucksters of the exercise. This reduces the system footprint to the space necessary to support the operators and eliminates traditional PC workstations from the exercise. The overall hardware, power, and space footprint is greatly reduced. Resources can be reallocated for other applications and exercise configurations without the need to wipe and reinstall the operating system and applications every time.

Cloud continues to support remote access, utilizing the same network infrastructure already in use with more traditional distributed simulation configurations. Zero, thin, or thick client access options are available depending on display performance needs and location.

Pre-exercise scenario development, analysis, and dry runs can take place on the cloud resident resources. Users can upload, run, and share scenarios without having to download and run on locally managed systems. Software is centrally updated so users have access to the same, most recent software version as well as the old version.

The cloud capability for on-demand self service and multi-tenancy provides M&S users with availability of the simulation as needed and accessible even while other users might be using the same application.

Offering M&S as a Service provides many benefits relative to what we do today while allowing innovation in the way we utilize M&S.

- Centralized hosting of simulation resources decreases the cost of ownership by reducing licensing requirements, hardware and software maintenance/upgrades, and facility resources (such as power and space).
- With flexible, scalable environments, ramping up new users and exercise environments are performed more quickly and at relatively modest cost, resulting in faster implementation and "time to value." In addition, the environment can scale according to need, increasing in size as expanded capability is needed, decreasing when needs are reduced—with overall ability to adapt in environments of sporadic use.
- This provides an environment that is device and location independent, expanding the accessibility of the resources.
- Together this provides for increased collaboration amongst the users of the environment—with common access to the same resources. Updates to the resources provide all the users with access to the same capabilities.

With these tools in place, we have the ability to change the entire process model for how we compose and use M&S capabilities, thus providing an opportunity for innovation. With freedom from lengthy implementation timelines, one can quickly and inexpensively "try out" new ideas. Users are also not held back from utilizing new tools and capabilities because of the need to support legacy systems.

This is a fundamental shift in how simulation will be delivered to the user community.

Our M&S-as-a-Service Exploration Approach

Our approach to addressing M&S as a Service was to begin by consulting with our local IT cloud experts. A number of hardware options were offered to us, but as we defined the virtual machines we needed for running our applications, it became clear that our resource needs were very different from what the standard IT offering could provide. An important lesson here was not to count on the IT department to provide the cloud capabilities needed for simulation.

Based on the NIST definition for cloud, we decided to focus on capabilities that were uniquely cloud (and not simply virtualization). Though virtualization was part of our solution, we wanted to see how the REAL cloud would support our needs.

SIMULATION IMPLEMENTATION WITH ONESAF

Our initial prototype used a simulation with a flexible, mature software architecture (OneSAF) and a virtualization infrastructure that is relatively mature and supports the functionality needed to support simulation's unique use of the environment (VMware®, VMware, Inc.).

OneSAF is the U.S. Army's next-generation entity-level simulation that provides a composable, distributed, and scalable simulation of real-world battlefield situations using validated physical models and doctrinally correct behavior models. It can support analysis, acquisition, planning, testing, training, and experimentation. OneSAF allows users to compose a wide range of complete simulation systems from a set of component-based tools, develop new or extended existing tools, as well as compose new single or multi-resolution entities, units, and associated behaviors from existing physical and behavioral software components.

VMware has a mature product line for virtualization services. Their cloud computing products provide a flexible, tailorable environment for automation and control of infrastructure resources. Access to their software tools has allowed the Cloud Simulation Infrastructure (CSI) project to quickly provision resources for simulation use.

OneSAF was a particularly good candidate for testing targeted cloud capabilities because of its composable architecture and flexible interface. OneSAF uses a gateway to communicate with the provisioning/service broker. This allows the broker to feed OneSAF Management and Control Tools (MCT). The gateway also allows the simulation to direct the provisioning and configuration of virtual machines (VMs) for use in a user designated configuration and exercise.

VMware's VCenter functionality allowed our gateway to issue commands to provision and unprovision resources, to install the OS, software, and exercises needed by the simulation. This ensures that the details of the configuration of the infrastructure for any exercise are "hidden" from the user—who focuses on the simulation and does not require any knowledge of how the simulation is hosted within the environment.

We developed an implementation that explored how simulations would execute in such an environment and how the virtualization and cloud tools provided by our selected hypervisor could be leveraged to create new M&S capabilities.

Based on our experience, we developed a framework for defining the environment that could help the community define standard approaches for cloud delivered M&S services.

We created a cloud-based service for OneSAF, called CSI that enables the U.S. Army to deploy simulation solutions directly to warfighter locations or to centralized simulation centers via enterprise networks. The result is a solution for providing training with lower operator overhead requirements, reduced exercise lead times, and lower overall hardware capital costs associated with legacy simulation approaches.

Modeling and Simulation as a Service Framework

The M&S as a Service Framework consists of three main components (figure 1): User Application Interfaces, Simulation Services, and Physical Hardware.

Framework Overview

The User Application Interfaces include zero, thin or thick client interfaces that allow the end user to interact with the simulation services. The Physical Hardware provides the CPU, memory, and network hardware infrastructure for hosting the simulation services component. The Physical Hardware implemented is a high-end network server utilizing virtualization for hosting simulation service components. Future work will explore how high performance computing (HPC) hardware might be addressed in this framework or something similar.

The Simulation Services component contains the actual simulations used for creating the simulation services being delivered. These simulations can interact locally with other locally hosted simulations or may connect to

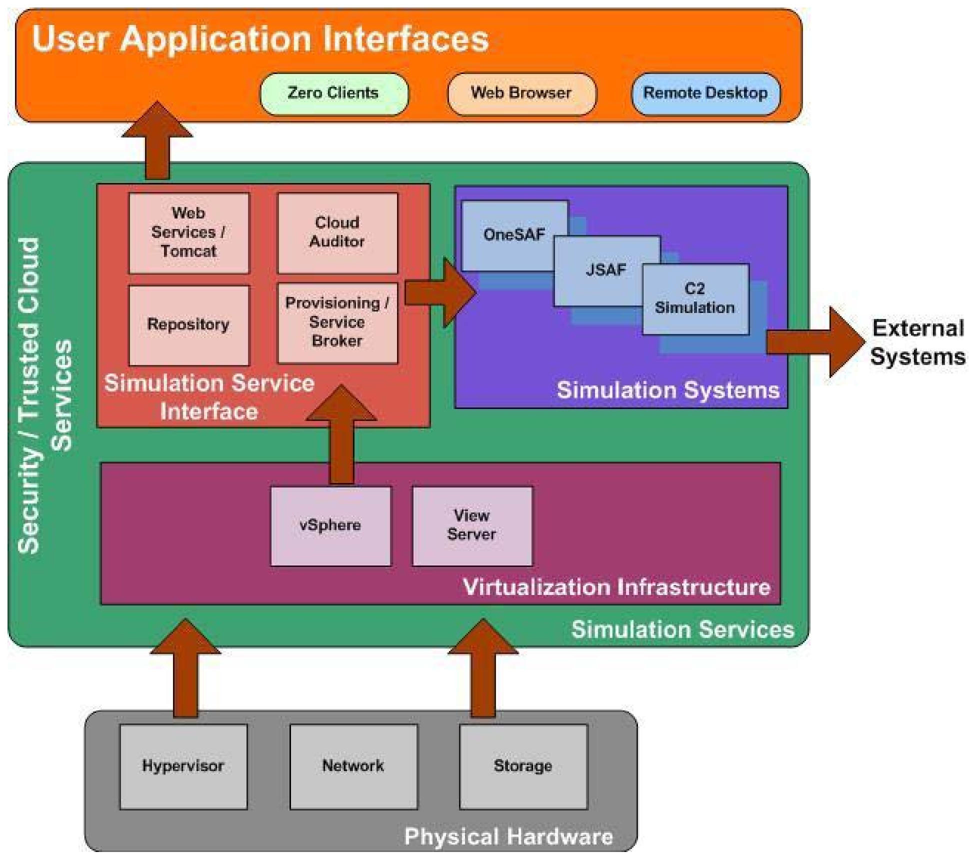


Figure 1: M&S-as-a-Service Framework

external systems. In addition, the simulation services component contains the virtualization infrastructure. In the case of an HPC not utilizing virtualization, this subcomponent might instead provide a set of job scripts that will assign simulation component processes to specific nodes in the underlying HPC infrastructure.

Prototype: Physical Hardware

Our initial configuration was built using three-year-old, high-end workstations (six total), each with Intel® Core™ i7 (quad core) processors (Intel Corporation), 12Gb RAM, 256Gb storage, and an NVIDIA® graphics card (NVIDIA Corporation). With this six-machine configuration we were able to stand up a virtualized environment and generate upwards of 25 VMs running two to three separate OneSAF exercises running on separate virtual networks.

Our current configuration (figure 2) consists of two host systems, each with:

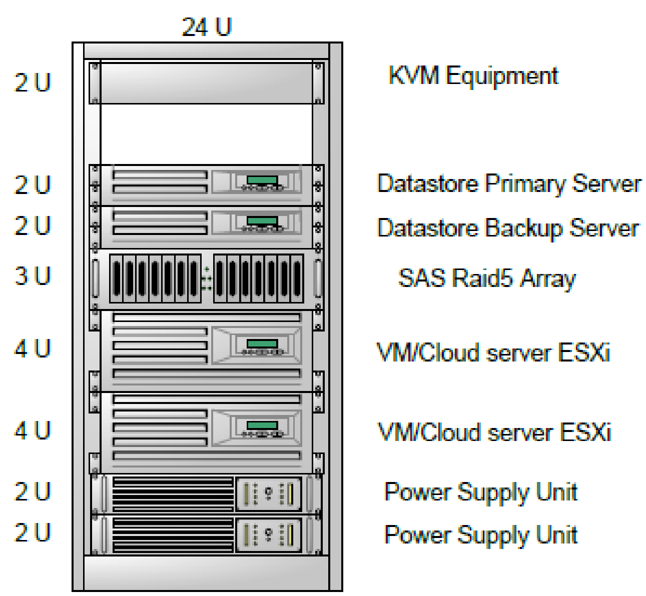


Figure 2: Cloud Server Configuration

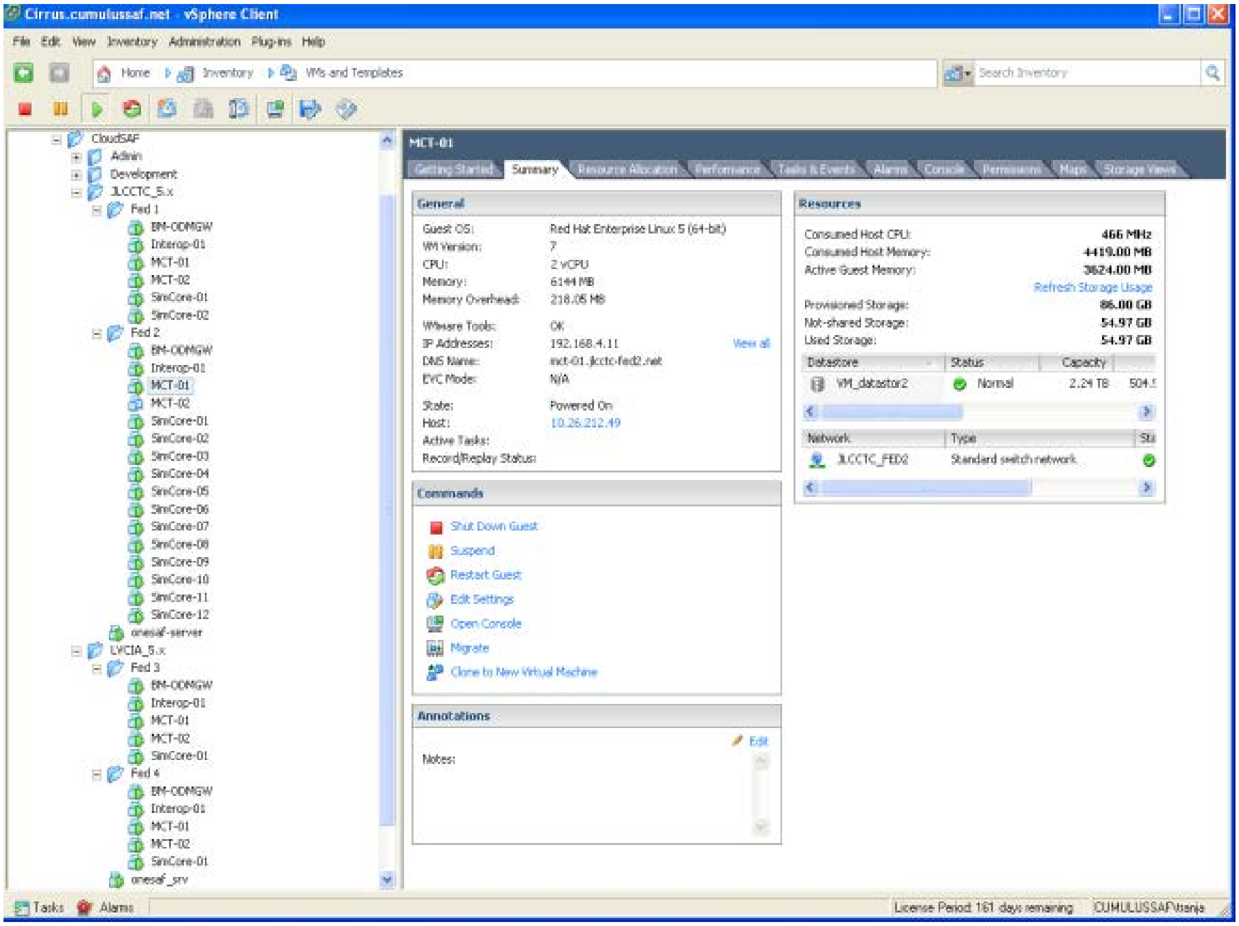


Figure 3: vSphere Access

- 40 physical CPU cores (80 hyperthreaded)
- 256Gb physical memory
- 8Gb fiber backbone between hosts and data store with a 1Gb network connection outside

In addition to the host systems, we have a 2TB data store. This configuration provides support for approximately 200 simulation-configured VMs.

Prototype: Virtualization Infrastructure

SAIC worked with VMware to assemble the right tools to support the capabilities required by our simulations (see figure 3).

In the end, we implemented:

- VMware vSphere®5 (VMware, Inc.)
- VMware View™5 (VMware, Inc.)

These products (which were themselves virtual appliances) provided a number of important capabilities:

- Linked cloning: Allowed our configuration to quickly clone VMs per the exercise configuration specified by the user
- Cloud/VM monitoring: Provided view into the performance of the individual VMs so we could monitor the state of the virtualized exercise as it responded to our user interface inputs
- Data redundancy and failover. We were able to isolate faults in individual VMs without interrupting the overall exercise.

In addition to the VMware products, we also used two products from Teradici Corporation: Zero Client and PCoIP®.

INITIAL CLOUD SERVICES IMPLEMENTED

Based on the NIST definition of the five essential characteristics for cloud computing we have implemented the following:

On-Demand Self Service*:* The configuration interface allows the user to select a OneSAF configuration and scenario—then causes the infrastructure to provision the needed resources using existing templates and linked clones. Metering is not yet provided.

Broad Network Access*:* Access to OneSAF is available over the network and has been demonstrated using thin client (web app, mobile app) as well as zero client and VM viewers. We have displayed our implementation (controls and OneSAF displays) using laptops, tablets and mobile phones.

Resource Pooling*:* OneSAF resources are dynamically assigned at the time of exercise configuration and then released when the resources are unprovisioned by the user in their application. The end user has no knowledge of where or what host the simulation services are running on. Currently, there is no control over the location of the resources since current delivery from a single location.

Rapid Elasticity*:* This is a key capability that has been implemented where resources are rapidly provisioned based on user configuration inputs. Initial configuration is performed automatically. We have also prototyped on the fly configuration of resources when the provisioned set proves to be inadequate to the simulation task at hand.

Measured Service*:* This has not yet been implemented.

CSI driven OneSAF is more than just Software as a Service (SaaS) where users are provided application level support. Our implementation controls the underlying platform and infrastructure services—and they are modifiable based on *simulation management and simulation exercise events.* The implementation crosses cloud layers so that the delivered simulation services are supported by the underlying infrastructure as needed—yet this capability is hidden from the user. From a user perspective, they are configuring a OneSAF exercise—the infrastructure provisions the required virtual resources (VMs, virtual nets) on the fly (they don't need to be pre-configured and sitting in VMs ready for use). Our goal in this implementation was to demonstrate the cloud philosophy that hides all the details of what was going on underneath the hood in order to offer a pure simulation service. The user accesses a service—CSI seamlessly delivers the service. It is this capability that takes the implementation beyond virtualization and more toward the cloud.

RESULTS

Our solution was unique in that we have developed the ability for OneSAF to dynamically provision new processors and VMs on demand in order scale to very large scenarios (more than 20,000 entities).

We developed middleware for OneSAF and VMware to exploit mechanisms that let CSI know when the simulation needs more compute resources as entities and scenario complexity impact performance. This middleware can be used with other simulation systems. CSI automatically cloned VMs to distribute the processing of OneSAF. The internal network of 1Gb/s provides more than sufficient data transfer capacity for several hundred VMs.

CONVERGENCE OF CLOUD SIMULATION WITH OPERATIONAL ARCHITECTURES

The U.S. Army and the U.S. Air Force (USAF) have recognized this force for change at both the enterprise and tactical levels of the military. For example, the CIO for the USAF recently explained their rationale for pursing cloud computing in the enterprise:

> *So I think the cloud may have a benefit there and may be a way for us to get at the mobile apps and mobile computing in a manner that now allows a greater mobility and at the same time doesn't increase the security or decreases our security posture [4].*

The U.S. Army is quickly moving cloud computing and virtualization into the tactical realm with its Distributed Common Ground System (DSCGS-A).

> *"In some cases, cloud computing has shortened the time needed to analyze information from days and hours to literally seconds," said Clark Daugherty, Distributed Common Ground System-Army program manager for Lockheed Martin Global Training and Logistics. "This program directly saves soldiers' lives*

> *when it comes to dealing with improvised explosive devices, intelligence about planned ambushes and attacks," Daughtery said [5].*

DSCGS-A is central to the U.S. Army's Operations-Intelligence (Ops-Intel) Convergence (OIC) concept identified by Program Executive Office for Command, Control, and Communications-Tactical (PEO C3T) for the Common Operating Environment represents an enormous step forward for the U.S. Army. The existing Army command and Control environment at tactical level has a large level of complexity because of the existing pattern of deploying individual computing hardware with each software element of the U.S. Army's mission command systems. Tactical cloud environments have the potential to save both personnel and dollar resources while making the U.S. Army more tactically agile and powerful by virtualizing these applications on high-performance, multiprocessor computers.

The Future

Imagine the scenario where a tactical commander at the edge of the battlefield will soon be on their handheld device, using the power of the cloud to ask:

- Computer—Plot three routes from my current location (point A) to my new assembly area/linkup (point B) where I will link up with local friendly forces. I want four routes, fastest, shortest, best concealment, and best coverage given current threat intel.
- Computer—In the point B assembly area, plot recommended location of my comms devices for the best coverage within the assembly area.
- Computer—I have three sensors with me, two EO/IR and one thermal. Plot recommended locations for these sensors to cover expected avenues of approach into the assembly area.
- Computer—I have two machine guns and six rifles, plot recommended locations and range cards to provide the best 360 protection for these weapons.
- Computer—I see a group of people on cell phones moving to my location. Who are these people and what do you know about the cell phones?
- As a group moves to link up and have provided the password, each person is asked to look into a handheld device and state his/her name. The computer takes a picture and checks his/her voice to verify that these are the expected people.

These are not science fiction scenarios. They are capabilities that are, for the most part, possible now with technologies that combine speech understanding, cloud, and mobile computing. Add to this mix entity-level simulation with high-resolution geospatial databases and we have a revolution in training, mission planning, and command systems.

We are now continuing this research as part of a continuing working relationship with the U.S. Army Research Labs under a Cooperative Research and Development Agreement (CRADA) for Cloud-based Simulation. We have continued to evaluate the capability with an implementation developed for experimentation. Communications-Electronics Research Development and Engineering Center (CERDEC) in using the Cloud-Simulation Infrastructure to support long-distance experiments. Finally, we are examining the potential of employing CSI on tactical systems to be used for mission planning.

The convergence of OneSAF with CSI promises new capabilities undreamed of a decade ago and is being driven by advances in technology and the U.S. Army's experience in moving high performance computing forward to tactical units in combat areas. We see the possibilities of CSI in that environment and are now exploring ways to make this future real.

Acknowledgments

We would like to acknowledge the support of the Army Research Labs under CRADA #12-05. We also thank the VMware team to include Scott Drucker and Jay Bokulic and appreciate the technical support of the Science Applications International Corporation (SAIC) research team, Kurt Hawkes, and Jay Duncan.

This paper was originally submitted for the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) by its authors under SAIC which is now the Leidos Corporation, Orlando, FL; has been adapted from I/ITSEC 2012, and is reprinted with permission.

References

[1] P. Mell and T. Grace, "The NIST Definition of Cloud Computing," NIST, Gaithersburg, MD, Special Publication 800-145, 2011. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

[2] V. Kundra, "Federal Cloud Computing Strategy," The White House, Washington, DC, Feb. 2011. Available: http://www.whitehouse.gov/sites/default/files/omb/assets/egov_docs/federal-cloud-computing-strategy.pdf

[3] http://www.onesaf.net/community [restricted access]

[4] B. Rosenberg, "Air Force CIO pushes for operational mindset," Defense Systems, Aug. 2011. Available: http://defensesystems.com/articles/2011/08/08/Interview-Lt-Gen-William-Lord.aspx?admgarea=DS&Page=1

[5] R. Barrett, "State companies helping Army with cloud computing," *Journal Sentinel*, May 2012. Available: http://www.jsonline.com/business/state-companies-helping-army-with-cloud-computing-7258roe-150105585.html

Authors' Biographies

Dr. Michael R. Macedonia

Dr. Michael R. Macedonia, is the assistant vice president for research at the University of Central Florida. He formerly was vice president and chief scientist for SAIC. Prior to joining SAIC, Macedonia was the general manager for Forterra Federal Systems in Orlando, FL, which was acquired by SAIC. Macedonia also was a senior national intelligence executive and the director of the Disruptive Technology Office (DTO, now IARPA) for the Office of the Director of National Intelligence. Prior to DTO, he was the chief technology officer for the U.S. Army Program Executive Office for Simulation, Training and Instrumentations (PEO STRI) in Orlando, FL. Macedonia was responsible for developing the technology strategy for the U.S. Army's lead training technology acquisition organization. Macedonia has a Ph.D. in computer science from Naval Postgraduate School. He also has an M.S.T. from the University of Pittsburgh and a B.S. from the United States Military Academy, West Point.

Dr. Christina L. Bouwens

Dr. Christina L. Bouwens is the chief technologist at MSCI where she is performing System of Systems architecture development and analysis for the Assistant Secretary of the Army (Acquisition, Logistics and Technology ASA(ALT)) System of Systems Engineering and Integration Directorate (SoSE&I). Before that she was a senior research scientist for Science Applications International Corporation (SAIC) in Orlando, FL. She has over 21 years' experience performing research and development with Forces Modeling and Simulation (FMS) and high performance computing technologies, in particular with distributed networked simulation (distributed interactive simulation, high-level architecture). She holds a B.S. in mathematics from Geneva College, an M.S. in mathematical science, and a Ph.D in industrial engineering (systems operations and modeling) from the University of Central Florida.

Mr. James E. Shiflett

Mr. James E. Shiflett is a program strategist for Leidos. He has over 30 years of experience in the management, research and development of DoD software systems. He has focused on strategic policy, project advocacy, operational requirements, test and evaluation, fiscal management and successful program execution. He formerly served as the director for FCS Training Systems and as the program manager for the largest simulation effort in the United States Army, the Combined Arms Tactical Training (CATT) program.

Soil Modeling for Mine Blast Simulation

AUTHORS

Mr. Frank Marrs, Dr. Mike Heiges
Aerospace, Transportation, and Advanced Systems Lab
Georgia Tech Research Institute
7220 Richardson Road
Smyrna, GA 30080
frank.marrs@gtri.gatech.edu
mike.heiges@gtri.gatech.edu

ABSTRACT

THIS PAPER PRESENTS THE RESULTS OF AN EFFORT TO CORRELATE AN LS-DYNA SIMULATION OF A BURIED MINE BLAST WITH PUBLISHED EXPERIMENTAL TEST DATA. THE FOCUS OF THE STUDY WAS ON SIMULATING THE EFFECTS OF SOIL MOISTURE CONTENT ON BLAST CHARACTERISTICS. A MATHEMATICAL MODEL FOR SAND IS PRESENTED THAT IS BASED ON SEVERAL PREVIOUSLY PROPOSED MODELS. THE SIMULATION CORRELATED WELL WITH THE RESULTS OF A MINE BLAST EXPERIMENT, THUS VALIDATING THE MATERIAL MODEL FOR SAND AT VARYING LEVELS OF SATURATION. THE MODEL PROVIDES AN EXCELLENT BASELINE FOR BLAST SIMULATIONS OF BURIED MINES AND A SOIL MATERIAL MODEL THAT CAN BE EXPANDED TO INCLUDE HIGHER FIDELITY MODELING, DIFFERENT SOIL TYPES, AND REAL-WORLD APPLICATIONS.

INTRODUCTION

Background

Over the past decade, significant efforts have been expended on developing personnel carriers that are substantially more resistant to landmines and improvised explosive devices (IEDs). While live fire testing on these vehicles is crucial to validating their effectiveness, the tests are expensive, variables can be difficult to control, and the results can show significant variability depending on which parameters are measured. The purpose of this investigation was to demonstrate the ability of modeling and simulation (M&S) – specifically LS-DYNA (Livermore Software Technology Corporation) – to reproduce test conditions using a realistic soil model.

A literature review found a number of experimental and simulation-based studies regarding factors that can affect blast response. The experimental studies were typically conducted at the sub-scale level with small test charges (e.g., 50 g to 200 g of C4 or TNT). The tests involved either a series of pressure probes mounted above the charge or a movable plate to capture the impulse imparted by the explosion. Investigators have examined the effects of the depth of burial, soil composition, soil moisture content, the location of the detonation point, charge shape, and type of explosive. Of these, soil moisture content can be a difficult test parameter to control, particularly in large-scale testing. Thus, the ability to use M&S to account for soil moisture variation could significantly improve the analysis of test results.

This study focused on simulating a blast event using the high strain rate finite element analysis software package LS-DYNA, and correlating the results with published test data. The soil model used explicitly reproduces the effects of the soil moisture level. This paper presents a brief description of the published experimental results used for the simulation correlation study followed by a description of the LS-DYNA simulation set up. A description of the soil modeling methods is also presented. The simulation results are then compared to the experimental data along with a discussion on the quality of the correlation.

Experiment Description

Anderson et al. [1] conducted a series of blast experiments consisting of a buried high explosive (HE) and a momentum plate suspended above the charge. The explosive used in that experiment was comprised of 625 g of Composition B with a 10 g PET-N detonator located bottom-center of the charge. The explosive was buried with the top of the disk 5 cm beneath the surface of the sand. The disk measured 3.7 cm in height by 11.3 cm in diameter, giving it a height to diameter (H/D) ratio of approximately H/D=1/3. Figure 1 depicts the experimental setup.

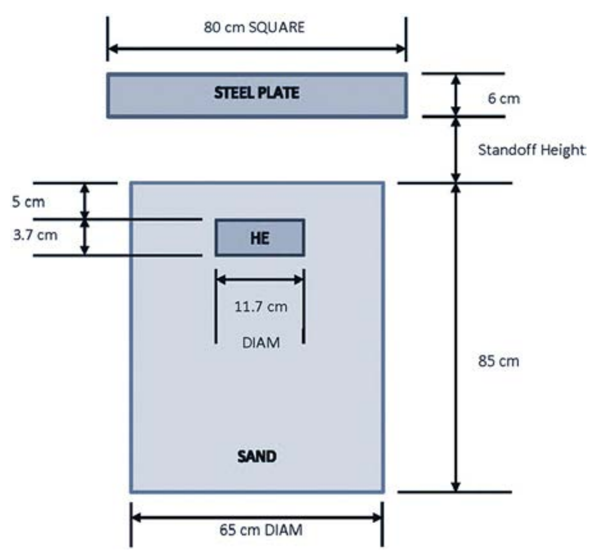


Figure 1: Dimensioned diagram of experimental setup

The momentum plate was a flat, square plate of steel measuring 80 cm x 80 cm x 6 cm and weighing 300 kg. The plate was supported by posts 20 cm above the surface of the sand for most of the experiments. In one case, the plate was positioned 30 cm above the sand. In addition to the flat steel plate, two V-shaped steel plates were also tested. One plate had a 90° internal angle while another plate had a 120° internal angle. The V-shaped plates were suspended such that the center of mass was 25 cm above the surface of the sand.

The sand used in the Anderson et al. experiment was described as "common silica sand" [1]. Grain size was less than 1 mm in diameter with 99% of the sand having a grain size less than 0.5 mm. As delivered, the sand had a mass density of 1.37 g per cubic centimeter and a moisture content of 7%. Moisture percentages of 14% and 22% were also tested. The mass densities at these moisture levels were 1.49 g/cm^3 and 1.67 g/cm^3, respectively. The type or method of moisture content measurement is not described; however, the tolerance on the measurement is given as ±0.03 g/cm^3.

The vertical displacement of the plate from its initial position was measured by using cable-pull potentiometers. The maximum height was the resultant variable, which was verified using high-speed video. Additionally, the accuracy of the cable-pull potentiometers was verified in one experiment via a plate-mounted accelerometer. The "jump velocity" – the theoretical maximum velocity of the plate – was calculated by using

$$V_j = \sqrt{2gH}$$

where V_j is the jump velocity, H is the maximum height that the plate reaches, and g is the acceleration of gravity at sea level. Note that this equation neglects air resistance. For each experimental setup, three experiments were performed. Table 1 lists the resulting jump velocities. Generally, the Anderson et al. [1] results demonstrated good repeatability.

MATHEMATICAL MODEL

Air, Explosive, and Plate Models

LS-DYNA, a general-purpose finite element program [2], was used to simulate the mine blast in three dimensions. The air, soil, and explosive were modeled as a single mesh domain of multi-material Arbitrary Langrangian Eulerian (ALE) elements measuring 60 cm square by 135 cm in height. All ALE mesh elements were cubic or very nearly so. The vertical direction of plate motion was selected as the Y-direction. Symmetry of the model was enforced at X = 0 and Z = 0 in order to reduce the number of elements by 75%; this accounted for two boundary surfaces. Movement at the bottom of the soil (Y = -85 cm) was constrained in the Y direction. Non-reflecting boundary conditions were applied to the remaining surfaces such that the air and soil were assumed to be infinite in the X and Z directions. The air was assumed to extend infinitely in Y. The default material for all

Experiment Setup	Sand Density (g/cc)	Moisture Content	Steel Plate Type	Standoff (cm)	Jump Velocity (m/s)
1	1.37	7%	Flat	20	6.54
1	1.37	7%	Flat	20	6.75
1	1.37	7%	Flat	20	6.50
2	1.37	7%	Flat	30	5.76
2	1.37	7%	Flat	30	5.42
2	1.37	7%	Flat	30	5.18
3	1.37	7%	V-Plate 90°	25	2.69
3	1.37	7%	V-Plate 90°	25	2.75
3	1.37	7%	V-Plate 90°	25	2.46
4	1.37	7%	V-Plate 120°	25	4.15
4	1.37	7%	V-Plate 120°	25	3.65
4	1.37	7%	V-Plate 120°	25	3.65
5	1.49	14%	Flat	20	7.23
5	1.49	14%	Flat	20	7.30
5	1.49	14%	Flat	20	7.01
6	1.67	22%	Flat	20	8.47
6	1.67	21%	Flat	20	7.58
6	1.67	22%	Flat	20	9.06

Table 1: Experimental Matrix and Jump Velocity Results

elements was specified as air; soil and explosive were carved out of the mesh using the *INITIAL_VOLUME_FRACTION_GEOMETRY card.

The air above the sand spanned 50 cm in height (Y = 0 to Y = 50 cm), and was modeled as *MAT_NULL (material type 9). Standard parameters were chosen for the density of air at sea level, viscosity, and the equation of state. Tables 2 and 3 list the parameters. In table 3, C0 – C6 are the polynomial equation coefficients, E0 is the initial internal energy per unit reference specific volume, and V0 is the initial relative volume. For all LS-DYNA inputs, the LS-DYNA theory manual [2] provides additional detail regarding the equations and theory behind modeling parameters.

Density	Pressure Cutoff	Viscosity
1.3	-1.00E-10	2.00E-05

Table 2: *MAT_NULL inputs for air (kg, m, s)

C0	C1	C2	C3	C4	C5	C6	E0	V0
0	0	0	0	0.4	0.4	0	2.5e+05	1.0

Table 3: *EOS_POLYNOMIAL inputs for air (kg, m, s)

The explosive was modeled within the ALE domain previously described. The material model, *MAT_HIGH_EXPLOSIVE_BURN, treats the products of the explosion as purely gaseous, making the choice of ALE elements appropriate. The parameters for Composition B were experimentally determined by Urtiew et al. [3] and we used those values to populate the material card (Material Type 8, *MAT_HIGH_EXPLOSIVE_BURN) and the equation of state card (*EOS_JWL) for the Jones-Wilkins-Lee equation of state (JWL-EOS) [4] for an explosive:

$$p = A\left(1 - \frac{\omega}{R_1 V}\right)e^{-R_1 V} + B\left(1 - \frac{\omega}{R_2 V}\right)e^{-R_2 V} + \frac{\omega E}{V}$$

Tables 4 and 5 present the values used to populate these cards.

Density	Deto-nation Velocity	Chapman-Jouget Pressure	Beta (burn flag)
1700	8000	3.00E+10	2

Table 4: *MAT_HIGH_EXPLOSIVE_BURN inputs for Comp B (kg, m, s)

A	B	R1	R2	Omega	E0	V0
5.24E+11	7.68E+09	4.2	1.1	0.50	8.50E+09	1.0

Table 5: *EOS_JWL inputs for Comp B (kg, m, s)

The modeling of the sand was significantly more complex than that of air, and is discussed in the soil material modeling section.

The steel plates were modeled using Lagrangian elements of type 2 in LS-DYNA; the average mesh size of the plate was the same as that in the ALE elements. Although the elements in the flat plates were cubic, the elements in the V-plates contained a limited amount of elements that were somewhat skewed. Standard properties for steel (table 6) were selected for the flat plate model, although the density was altered so that the mass of the plate was exactly 300 kg. Since the *MAT_ELASTIC card was used, elastic deflections of the steel plate were considered. Yielding of the plates, however, was not considered. The jump velocity was assumed to be the maximum rigid body velocity in the Y-direction; this is the average of the nodal velocities of the plate.

Density	Young's Modulus	Poisson's Ratio
7813	4.20E+11	3.30E-01

Table 6: *MAT_ELASTIC inputs for flat steel plate (kg, m, s)

The V-plates were modeled using the same material properties as the flat plate; however, the material densities were modified to match the plate masses given in Anderson et al. [1]. The stiffening plates were not modeled; therefore, the stiffness of the plates was scaled to account for this omission. The V-plates with stiffening plates were modeled in Solidworks® (Dassault Systèmes, Waltham, MA) and deflected with a normal pressure of 2 MPa. Then, the stiffness of V-plate models without stiffening plates was scaled up to match the deflection of the stiffened plates. The parameters used for the 90° and 120° plates are given in table 7. Solidworks was also used to find the height of the center of gravity (CG) for each plate. The heights of the CG was 170 mm and 130 mm above the bottom edge of the V-plate for the 90° and 120° plates, respectively.

Plate	Density (g/cc)	Poisson's Ratio	Stiffness (Pa)
90°	8.682	0.33	5.0E+11
120°	8.422	0.33	4.2E+11

Table 7: V-Plate Parameters

The interaction between the solid plate and the three fluid models – air, explosive products, and soil – bears mentioning here. The fluid-structure interaction (FSI) is the key to determining the reaction of structures to explosions. LS-DYNA employs a penalty-based coupling approach. This means that each time a specified number of time steps has elapsed, the code checks for penetration of the fluids into the structure. When penetration is detected, a weighted force proportional to the penetration distance is applied. This approach is clearly non-physical. However, this paper only validates the non-physical approach as only the simplest parameters were chosen, yet good correlation was achieved. A single FSI card (*CONSTRAINED_LAGRANGE_IN_SOLID) was used to couple each fluid to the solid plate, for a total of three FSI cards (table 8). Since the densities and stiffnesses were very different between the materials being coupled and the speed of impact was high, the ILEAK flag was turned on (set ILEAK=2) for all FSI cards. Otherwise, all the options used in the code were the default ones. The FRCMIN flag was set to 0.4-0.6 for each coupling so that the couplings did not "turn on" at the same time. The number of quadrature points, NQUAD, was set to five. All other flags defined the materials used in the coupling.

SLAVE	MASTER	SSTYP	MSTYP	NQUAD	CTYPE	DIREC	MCOUP
4	1	1	1	5	4	2	-93
START	**END**	**PFAC**	**FRIC**	**FRCMIN**	**NORM**	**NORMTYP**	**DAMP**
0.00E+00	1.00E+10	0.1	0	0.40	0	1.0	0
CQ	**HMIN**	**HMAX**	**ILEAK**	**PLEAK**	**LCIDPOR**	**NVENT**	**BLOCKAGE**
0	0	0	2	0.1	0	0	0

Table 8: Example *CONSTRAINED_LAGRANCE_IN_SOLID card

Soil Material Model

LS-DYNA offers several material models that can be used to represent soil. The soil material model selected for this analysis was *MAT_SOIL_CONCRETE (material type 78). The main components of the model are the normal stress – volumetric strain relationship and the plastic yield function. Fiserova's thesis [5] was the basis for the stress-strain relationship presented here for partially saturated soil, and the yield function is a modified version of that described in Laine and Sandvik [6].

Fiserova's method [5] for developing the stress-strain relationship for partially saturated soil was based on a relative volume approach. Soil consists of solid granular particles and inter-particle voids filled with either air or water. The density of soil, ρ, is determined by:

$$\rho = \frac{m_s + m_a + m_w}{V_s + V_a + V_w}$$

where m represents mass, V is the partial volume with the subscript s denoting the solid portion of the soil, a denoting air, and w denoting water. For dry soil $V_w = 0$ and $m_w = 0$. Assuming the mass of the air is negligible, the dry density, ρ_d, can be expressed as:

$$\rho_d = \frac{m_s}{V_s + V_a}$$

The initial bulk density, ρ, can be expressed in terms of the dry density and water content, ω:

$$\rho = \rho_d(1 + \omega)$$

The void ratio, e, is defined by:

$$e = \frac{V_a + V_w}{V_s} = \frac{\rho_s}{\rho_d} - 1$$

where ρ_s is the average density of the solid particles. Porosity, n, is defined as:

$$n = \frac{V_a + V_w}{V_s + V_a + V_w} = \frac{e}{1 + e}$$

The degree of saturation, S_r, is defined as:

$$S_r = \frac{V_w}{V_a + V_w} = \frac{\omega\rho_s}{e\rho_w}$$

where ρ_w is the density of water, 1000 kg/m^3.

The initial relative volumes of the air, water, and solid particles are defined as:

$$\alpha_{a_0} = \frac{V_a}{V_s + V_a + V_w} = n(1 - S_r)$$

$$\alpha_{w_0} = \frac{V_w}{V_s + V_a + V_w} = nS_r$$

$$\alpha_{s_0} = \frac{V_s}{V_s + V_a + V_w} = 1 - n$$

The relative volumes for air, water, and solid particles under pressure were calculated using their respective equations of state. The equations of state for air and water are well established:

$$\alpha_{a_p} = \alpha_{a_0}\left(\frac{p}{p_0}\right)^{-1/k_a}$$

$$\alpha_{w_p} = \alpha_{w_0}\left(\frac{p - p_0}{\rho_{w_0} c_{w_0}^2} k_w + 1\right)^{-1/k_w}$$

where p is the pressure, k_a=1.4, k_w=3, and c_w is the speed of sound in water (1,414 m/s). The equation of state for solid particles is not known, but the relative volume for the solid particles under pressure is assumed to have the form:

$$\alpha_{s_p}/\alpha_{s_0} = a_s(p - p_0)^{k_s}$$

where a_s and k_s are unknown but can be solved for given pressure-strain data for the soil at a specified saturation level. The density of the soil under pressure can be expressed as:

$$\rho_p = \frac{\rho_0}{\alpha_{s_p} + \alpha_{a_p} + \alpha_{w_p}}$$

Using the pressure-density data derived by Laine and Sandvik [6], and calculating the relative volumes of air and water based on their equations of state, the relative volume of the solid particles, α_{s_p}, can be found as a function of pressure using the previous equation.

$$\alpha_{s_p} = \frac{\rho_0}{\rho_p} - \alpha_{a_p} - \alpha_{w_p}$$

Laine and Sandvik [6] was used as the baseline test data to define the parameters a_s and k_s. The sand used for these tests was relatively similar to that used in Anderson et al. [1], with a moisture content of 6.57% and a "dry density" of 1.574 g/cm^3. It was assumed that the "dry density" did not include the 6.57% moisture. Finally, the Laine and Sandvik [6] sand was described as "medium to coarse" rather than the mostly medium and fine sand of Anderson et al. [1]. The values calculated for Laine and Sandvik sand data at 6.57% moisture content are listed in table 9. Table 10 presents the relative volumes as a function of pressure.

Parameter	Symbol	Value
Moisture content	ω	0.0657
Dry density	ρ_d	1.574E+03
Solid particle density	ρ_s	2.641E+03
Initial bulk density	ρ_0	1677
Void ratio	e	0.68
Porosity	n	0.40
Density of water	ρ_w	1000
Saturation	S_r	0.256
Initial relative air vol.	α_{a0}	0.30
Initial relative water vol.	α_{w0}	0.10
Initial soil relative vol.	α_{s0}	0.60

Table 9: Laine and Sandvik [5] Sand Data at 6.57% Moisture Content

The relative volume of the solid particles [6] were plotted against pressure and fitted with an exponential curve to determine a_s and k_s (figure 2). The endpoint values are removed as possible outliers (i.e., positions of high and low compression). The values of a_s and k_s are found to be 5.9267 and -0.0926, respectively.

Once a_s and k_s are known, the relative volume of the solid particles can be determined based on the equation of state for soil at any level of water content. Table 11 presents the

p (Pa)	p-p_0 (Pa)	ρ_p (kg/m^3)	α_{ap}	α_{wp}	α_{sp}	α_{sp}/α_{s0}
1.01E+05	0.00E+00	1.68E+03	0.30060	0.1034	0.5959	
4.58E+06	4.48E+06	1.74E+03	0.01972	0.1032	0.8414	1.4118
1.50E+07	1.49E+07	1.87E+03	0.00846	0.1027	0.7841	1.3156
2.92E+07	2.91E+07	2.00E+03	0.00526	0.1020	0.7328	1.2295
5.92E+07	5.91E+07	2.14E+03	0.00317	0.1005	0.6787	1.1389
9.81E+07	9.80E+07	2.25E+03	0.00221	0.0988	0.6445	1.0814
1.79E+08	1.79E+08	2.38E+03	0.00143	0.0955	0.6078	1.0199
2.89E+08	2.89E+08	2.49E+03	0.00102	0.0917	0.5823	0.9770
4.50E+08	4.50E+08	2.59E+03	0.00074	0.0871	0.5611	0.9414
6.51E+08	6.51E+08	2.67E+03	0.00057	0.0824	0.5449	0.9144

Table 10: Relative Volumes as a Function of Pressure

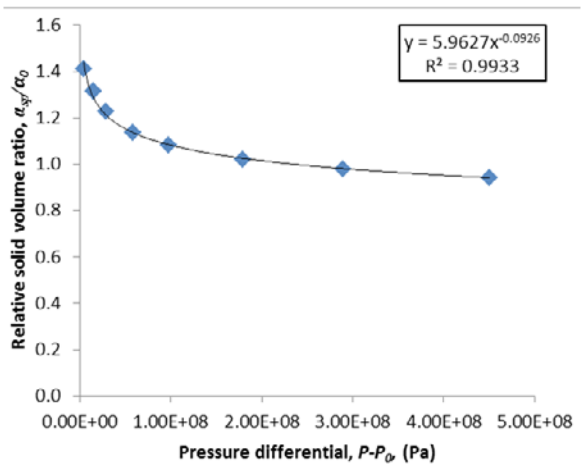


Figure 2: EOS for Soil Solid Particles Using Laine and Sandvik Data

initial density and relative volumes for the three levels of saturation used in Anderson et al. [1]. The particle density was assumed to be the same as in Laine and Sandvik [6], and the dry density was adjusted to meet the water content and wet densities given in Anderson et al. [1]. The dry density varied slightly with water content; it was assumed that wet density was a more important parameter to match.

Parameter	Symbol	7%	14%	22%
Moisture content	ω	0.07	0.14	0.22
Initial Dry Density (kg/m^3)	ρ_d	1280	1310	1390
Initial bulk density (kg/m^3)	ρ_0	1370	1490	1690
Initial relative air vol.	α_{a0}	0.43	0.32	0.17
Initial relative water vol.	α_{w0}	0.09	0.18	0.30
Initial soil relative vol.	α_{s0}	0.48	0.48	0.52

Table 11: Sand Data at Various Moisture Contents

Several departures were taken from the Fiserova [5] derivation of strength model and equation of state of the soil: the theoretical maximum density was included, cohesion was considered, Poisson's ratio was calculated, and the volumetric strain was defined as positive in compression. First, the theoretical maximum density of the soil was enforced. This state occurs when all of the air has been expelled from the soil by increasing pressure. The density at zero pressure of the theoretical maximum density can be calculated from a volume-weighted average of the densities of water and the solid soil:

$$\rho_{TMD} = \frac{\rho_{w0}\alpha_{w0} + \rho_{s0}\alpha_{s0}}{\alpha_{w0} + \alpha_{s0}}$$

Under pressure, the theoretical maximum density material was assumed to compress linearly along a line of constant bulk modulus. Thus, the bulk modulus, K_{TMD}, was defined based on the definition of the bulk modulus of mixtures [7]:

$$K_{TMD} = \frac{K_w K_s}{K_w \alpha_{s0} + K_s \alpha_{w0}}$$

The bulk modulus of water is approximately 2.16E+9 Pa while the bulk modulus of the soil particles was assumed to be 5.67E+10 Pa [6]. The relationship between pressure and density can be computed from the definition of bulk modulus using:

$$K = \rho \frac{\partial P}{\partial \rho}$$

$$\frac{\partial P}{\partial \rho_{TMD}} = \frac{K_{TMD}}{\rho_{TMD}}$$

In addition, a pressure-density curve defines the soil without air – referred to here as the line of theoretical maximum compaction – across which the pressure-density relationship of the complete soil material cannot cross as:

$$P_i = \left(\frac{\partial P}{\partial \rho_{TMD}}\right)(\rho_i - \rho_{TMD})$$

This limit was imposed on the computed pressure-density relationships after computation.

Note that at very low pressure levels and high levels of saturation, the approach to developing the EOS in this study has an unrealistic artifact. The volumetric strain, ε, is initially negative; i.e., the volume initially "increases" as the pressure increases. This is caused by the assumption that all three materials (air, water, and solid particles) are always under the same pressure. When this pressure is used in the EOS for air, the calculated air density increases significantly or the volume decreases significantly. To balance out the densities of the air, water, and solid particle mixture to match the soil test data, the volume of solid particles expands initially; i.e., the volume of solid particle under pressure is higher than the volume under nominal pressure. In reality, the structure of the solid particles is able to bear pressure loading so that the air voids are under a lower pressure and do not collapse as quickly. At much higher pressures, the soil acts like a fluid and all three components are under the same pressure. Because explosions occur at very high pressures, the low-pressure expansion artifact does not appear to adversely affect the simulation results. Interpolation data points were chosen so that this mathematical anomaly was avoided.

The computation of EOS for the various densities went as follows. First, ten pressure data points were selected. Constitutive equations for volume fraction were imposed to obtain the relative volume fractions of soil, air, and water at each of these ten pressures. Next, the density was calculated at each of the ten pressure values. Each point was checked to ensure that it was to the left of the line of theoretical maximum compaction. If any density was found to be above the theoretical maximum, the pressure-density relationship was adjusted so that the curve was parallel to the line of theoretical maximum compaction. Volumetric strains were calculated at each of the ten pressures (defined as positive for LS-DYNA) as follows:

$$\varepsilon_i = -ln\left(\frac{\rho_0}{\rho_i}\right)$$

The ten pressure values were then reselected so that no negative volumetric strains occurred and exactly one segment was as close as possible to the line of theoretical maximum compaction. The pressures were also chosen such that they were geometrically spaced in volumetric strain. These ten pairings of (gage) pressure and volumetric strain defined the EOS for the sand at three separate levels of water content. Figure 3 depicts the plots of the resulting EOS, along with Laine and Sandvik [6] data.

The strength model was the same as that in Fiserova [5]; a Mohr-Coulomb model of failure. In this model, the yield stress is linear in pressure:

$$Y = c + P\tan(\phi)$$

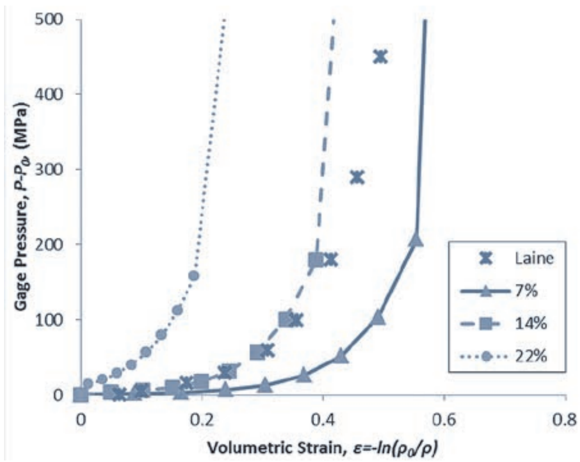


Figure 3: EOS models of sand for varying water contents

where c is the maximum tension the soil can carry (cohesion), P is the applied pressure, and φ is the friction angle. In Fiserova [5] it is assumed that the friction angle is constant and the Poisson's ratio, v, is calculated. However, in the current paper, it is assumed that the Poisson's ratio varies with water content (as a volume-weighted average). Using the partial volume approach:

$$\nu = \nu_s\alpha_{s_0} + \nu_w\alpha_{w_0} + \nu_a\alpha_{a_0}$$

where v_w=0.4999 and v_a=0.0001, Poisson's ratio for the solid particles, v_s, was found to be 0.222, again using Laine and Sandvik's data [6] as a reference.

The friction angle calculated from the Poisson's ratio is expressed as:

$$\phi = \sin^{-1}\frac{2\nu - 1}{\nu - 1}$$

The value of the friction angle was assumed to be constant throughout the range of pressures. Although the Poisson's ratio varies during compression, the changes in friction angle are very small with respect to changes in Poisson's ratio.

Cohesion, c, was calculated from Grujicic et al. [8]:

$$c = \omega^5(729\text{ kPa})$$

where ω is the water content. This parameter was expected to make little difference in results but was included for completeness, as sandy soil does not tend to adhere to itself.

The yield stress was linear up to an assumed maximum of 2.26E6 Pa, the unconfined strength of Pike's Peak Granite and an estimated maximum of the soil particles, as in Laine and Sandvik [6]. Figure 4 presents the strength curves.

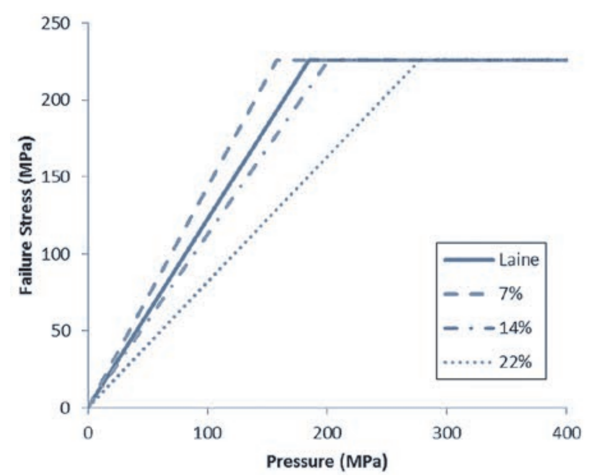


Figure 4: Strength models for varying water contents of sand

The bulk modulus, K, was calculated as a function of pressure and density from the definition of bulk modulus,

$$K_i = \rho_i \frac{P_i - P_{i-1}}{\rho_i - \rho_{i-1}}$$

and the shear modulus, G, was calculated as a combination of Poisson's ratio and bulk modulus,

$$G = \frac{3K(1-2\nu)}{2(1+\nu)}$$

The material model in LS-DYNA only allows constant values for the bulk and shear moduli; i.e., the variation with pressure is not modeled. The values of K and G were selected as the values that occur at an approximate soil density of 1.6-1.7 g/cm^3. Tables 12 through 14 displays the calculated parameters for all soil types.

RESULTS AND DISCUSSION

In the experiment by Anderson et al. [1], the "jump velocity" was used to evaluate the resulting energy of the

7% Moisture		14% Moisture		22% Moisture	
ε	P (Pa)	ε	P (Pa)	ε	P (Pa)
0.000	0.00E+00	0.000	0.00E+00	0.000	0.00E+00
0.087	1.60E+06	0.049	3.00E+06	0.011	1.41E+07
0.166	3.20E+06	0.101	5.39E+06	0.035	1.99E+07
0.238	6.42E+06	0.151	9.67E+06	0.059	2.81E+07
0.305	1.29E+07	0.198	1.74E+07	0.083	3.97E+07
0.368	2.58E+07	0.245	3.11E+07	0.108	5.60E+07
0.430	5.16E+07	0.291	5.59E+07	0.133	7.91E+07
0.491	1.03E+08	0.338	1.00E+08	0.159	1.12E+08
0.554	2.07E+08	0.388	1.80E+08	0.187	1.58E+08
0.605	1.24E+09	0.460	9.70E+08	0.261	6.69E+08

Table 12: Soil compression parameters used in LS-DYNA simulation

7% Moisture		14% Moisture		22% Moisture	
Pressure (Pa)	Yield Stress (Pa)	Pressure (Pa)	Yield Stress (Pa)	Pressure (Pa)	Yield Stress (Pa)
-1.23E+00	0.00E+00	-3.92E+01	0.00E+00	-3.76E+02	0.00E+00
1.58E+08	2.26E+08	2.01E+08	2.26E+08	2.77E+08	2.26E+08
3.00E+09	2.26E+08	3.00E+09	2.26E+08	3.00E+09	2.26E+08

Table 13: Soil strength parameters used in LS-DYNA simulation

	7% Moisture	14% Moisture	22% Moisture
G (Pa)	1.68E+07	3.90E+07	2.22E+08
K (Pa)	2.12E+07	4.67E+07	2.45E+08
ν	0.15	0.2	0.27
ϕ (deg.)	55.09	48.39	39.22

Table 14: Soil constants used in LS-DYNA simulation

explosion delivered to the steel momentum plates. The LS-DYNA experiment showed excellent correlation with the experimental results. Figures 5 and 6 show the results of using various mesh sizes for the flat plate and V-plate experiments, respectively. The solid bars in these figures represent the range of the experimental results.

All results show convergence as the mesh size decreases. It appears that the smallest mesh size – 10 mm – provides the best agreement, as expected. However, an 18 mm mesh provided results that are reasonably similar to the 10 mm mesh. The flat plate results suggest that the 50 mm mesh is too coarse to characterize the blast. Note that the explosive was only 37 mm in height and 117 mm in diameter. Thus, the 10 mm mesh has approximately six elements in the radial direction and four elements in the thickness direction of the explosive. This is on the lower end of the recommended

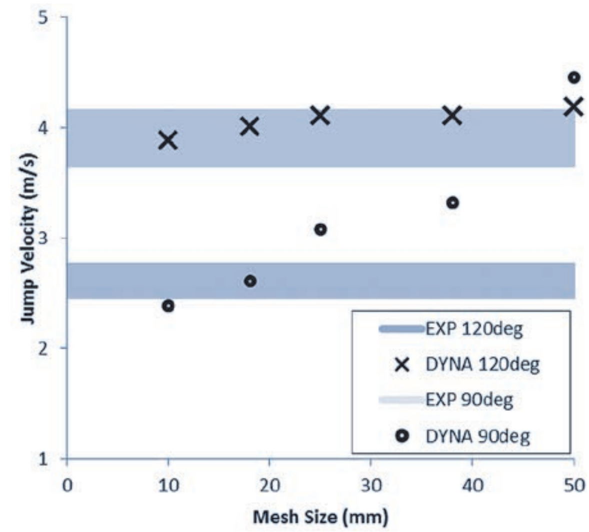


Figure 6: V-plate simulation results

number of desired elements, and explains why complete convergence is not observed. Computational time, however, was a limiting factor. A 5 mm mesh simulation would have taken several days to perform while running 30 parallel processes at 2.7 GHz processor speed each.

Simulation results are tabulated in table 15. One can see the excellent agreement between the simulation and the experiment. The error is less than 10% at all points and less than 5% in most cases. In all but one case, the simulation result falls within the variation of the experimental data.

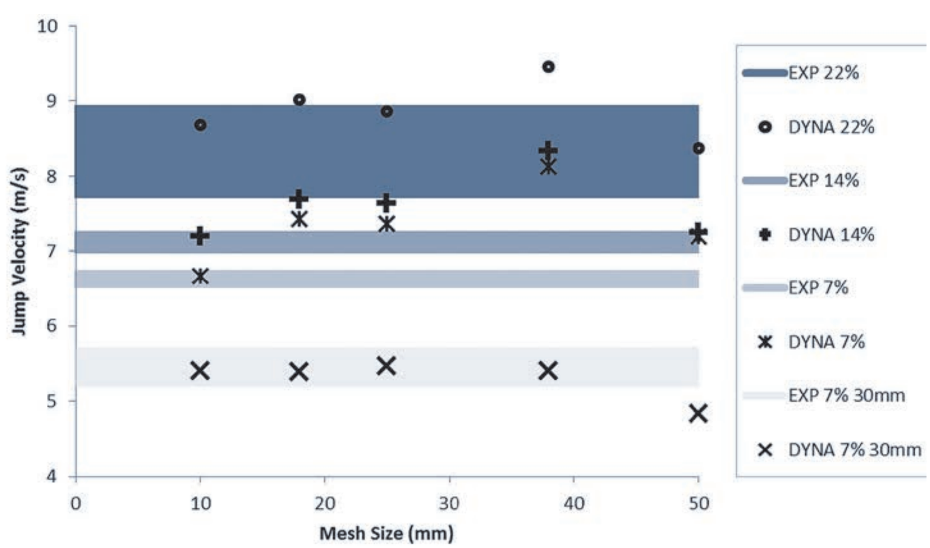


Figure 5: Flat plate simulation results

Note that the equation for the jump velocity used in Anderson et al. [1] neglects the air drag and gravity losses during the motion of the plate after impact. These phenomena are integrated into the LS-DYNA simulation, although their contributions are expected to be negligible. The good agreement of the simulation and experiment confirms this assumption.

The experimental data suggests a linear relation-

Soil Density (%)	Standoff (cm)	Plate Configuration	Average Exp. V_J (m/s)	Maximum Change V_J (m/s)	Simulation V_J (m/s)	% Error
7	20	Flat	6.60	0.15	6.66	1.0
14	20	Flat	7.18	0.17	7.20	0.3
22	20	Flat	8.37	0.79	8.68	3.7
7	30	Flat	5.45	0.31	5.40	-1.0
7	25	V-90deg	2.63	0.17	2.38	-9.6
7	25	V-120deg	3.82	0.33	3.88	1.7

Table 15: Tabulated simulation results compared to experimental results

ship between the soil moisture content and momentum (or jump velocity) of the flat plates at 20 mm standoff,

$$V_i = \frac{\rho_i}{\rho_0} V_0$$

Figure 7 shows the experimental results compared to those found in the LS-DYNA simulation. The simulation closely mimics the trend found in Anderson et al. [1], suggesting that the density of the soil is a key parameter in momentum transfer from the blast to the structure. A linear trend line is also provided. The coefficient of determination is ~0.986, suggesting that momentum transfer may be linear in soil density.

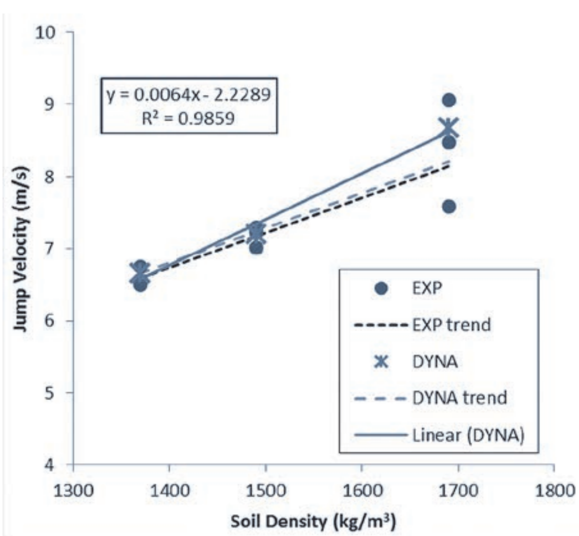


Figure 7: Variation of jump velocity with soil density

Conclusion

The results from the LS-DYNA simulation correlate very well with the data from a mine blast experiment. These results validate an explicit material model for sand that accounts for variation in soil saturation levels. The model provides an excellent baseline for blast simulations of buried mines and a soil material model that can be expanded to include higher fidelity modeling decision, different soil types, and real-world applications.

The success of the LS-DYNA simulation opens several areas of further research. Most obviously, the model can be applied to simulate the impact of real-world blasts on structures designed to survive blasts (e.g., military vehicles). The soil model can be extended to examine other types of soils such as silty or clayey soils. The model can also be used to generalize the blast momentum transferred based on charge depth of burial, charge size, and soil type.

References

[1] C. E. Anderson et al., "Mine blast loading experiments," *International Journal of Impact Engineering*, vol. 38, no. 8-9, pp. 697–706, 2011.

[2] *LS-DYNA Theory Manual*, March 2006, Livermore Software Technology Corporation, Livermore, CA Available: http://www.dynasupport.com/manuals/additional/ls-dyna-theory-manual-2005-beta/at_download/file

[3] P. A. Urtiew et al., "Shock Initiation of Composition B and C-4 Explosives; Experiments and Modeling," *International Conference on Shock Waves in Condensed Matter*, Saint Petersburg, Russia, September 18, 2006.

[4] E. Lee et al., "JWL Equation of State Coefficients for High Explosives", Lawrence Livermore Laboratory, Rept-UCID-16189, (1973).

[5] D. Fiserova, "Numerical Analyses of Buried Mine Explosions with Emphasis on Effect of Soil Properties on Loading," PhD dissertation, Defence College of Management and Technology, Cranfield University, Shrivenham, UK, 2006.

[6] L. Laine and A. Sandvik, "Derivation of Mechanical Properties for Sand," Proceedings of the 4th Asia-Pacific Conference on Shock and Impact Loads on Structures, CI-Premier PTE LTD, Singapore, Nov. 2001, pp. 361-368.

[7] G. Bar-Meir. (2010). *Basics of Fluid Mechanics* (Version 0.3.0.3) [e-book]. Available: http://f3.tiera.ru/2/P_Physics/PC_Classical%20physics/PCfm_Fluid%20mechanics/Bar-Meir%20G.%20Basics%20of%20Fluid%20Mechanics%20(free%20web%20version,%202010)(282s)_PCfm_.pdf

[8] M. Grujicic et al., "Parameterization of the Porous-Material Model for Sand with Different Levels of Water Saturation," *Soil Dynamics and Earthquake Engineering*, vol. 28, pp. 20-35, 2008.

Authors' Biographies

Mr. Frank Marrs

Mr. Frank Marrs is a research engineer at Georgia Tech Research Institute (GTRI). Mr. Marrs' background is control algorithms and modeling and simulation (M&S) for defense applications. His M&S activities include the dynamical simulations of impacts and mine blasts in LS-DYNA. Other activities include the development of a multi-body dynamics simulation for an investigation into large ground vehicle suspension algorithms. Mr. Marrs received an M.S. in mechanical engineering from the University of California-Davis in 2010 and a B.S. in aerospace engineering from the Georgia Institute of Technology in 2008. He joined GTRI in 2010.

Dr. Mike Heiges

Dr. Mike Heiges is a principal research engineer at Georgia Tech Research Institute (GTRI) with over 25 years of experience as an aerospace engineer. His background is in flight dynamics and control and in modeling and simulation (M&S) for aerospace applications. Dr. Heiges' M&S activities include the development of a multi-body dynamics simulation for an investigation into advanced technologies for projectile guidance and control. This work supported a study on using guided projectiles to intercept rockets, artillery, and mortar threats. Other activities include the development of simulation-based tools for test and evaluation of intelligent autonomous systems. Dr. Heiges received a Ph.D. in aerospace engineering from the Georgia Institute of Technology in 1989. He joined GTRI in 1992 after working for three years at Bell Helicopter Textron as a handling qualities specialist.

About the *M&S Journal*

The *M&S Journal* is a quarterly publication for modeling and simulation professionals in the Department of Defense and other government organizations, academia, and industry.

- Focus. The focus of the *M&S Journal* is on the topical and the timely: What is changing in M&S applications and practices? What trends are affecting M&S professionals? What challenges can we expect in the future?
- Organized by Theme. Each issue of the *M&S Journal* explores a single area in M&S, offering the reader a greater understanding of the challenges and opportunities from a variety of perspectives.
- Our Authors. Articles are written by accomplished M&S practitioners recognized for expertise in their areas of specialty.
- Guest Editors. Respected leaders serve as guest editors, providing viewpoints relating to the theme of the issue.
- Editorial Board. Content for the *M&S Journal* is shaped by an Editorial Board comprised of leaders from all sectors of M&S.
- Peer Reviewed. The *M&S Journal* is peer reviewed by M&S professionals. Our review process is based upon substance, accuracy, relevance, and clarity.

How to Subscribe

If you would like to subscribe to the *M&S Journal*, send an email to:

MS-Journal-Subscribe@Lists.AlionScience.com

Comments and questions for the *M&S Journal* Editorial Staff may be emailed to:

osd.pentagon.ousd-atl.list.msco-ask-msco@mail.mil_
or call 703-933-3323 or 888-566-7672

Editorial Board

Editorial Staff

Spring Edition 2014
Volume 9 Issue 1

M&S JOURNAL

A DoD MODELING AND SIMULATION COORDINATION OFFICE PUBLICATION